2009

Polyphonie Pitch Estimation and Modified Frequency Tracking System

Quan Wen

Polyphonic Pitch Estimation and Modified Frequency Tracking System

(Spine title: Polyphonic Pitch Estimation)

(Thesis format: Monograph)

by

Quan Wen

Graduate Program
in
Engineering Science
Electrical and Computer Engineering

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Engineering Science

School of Graduate and Postdoctoral Study
The University of Western Ontario
London, Ontario, Canada

THE UNIVERSITY OF WESTERN ONTARIO
SCHOOL OF GRADUATE AND POSTDOCTORAL STUDY

CERTIFICATE OF EXAMINATION

Supervisor:

Dr. Brown, Lyndon J.

Supervisory Committee:

Examiners:

Dr. Asokanthan, S.

Dr. Kermani, M.

Dr. McIsaac, K.

The thesis by

Quan Wen

entitled:

Polyphonic Pitch Estimation and Modified Frequency Tracking System

is accepted in partial fulfillment of the
requirements for the degree of
Master of Engineering Science

Date: _____

_____
Chair of the Thesis Examination Board

ABSTRACT

In this thesis, two applications are presented. The first one is the polyphonic pitch estimation algorithm based on a previous peer's work. The algorithm is an iterative multiple-step approach. The input signals are first transformed to a time-frequency representation using Instantaneous Fourier Decomposition. Then a Computational Auditory Scene Analysis based method extracts notes from the time-frequency representation. The modifications are presented and compared with previous work, results on MIDI music are presented and discussed.

The other application is the frequency tracking algorithm based on adaptive internal model theory, being able to capture the initial part of a monophonic signal with multiple harmonics. By running an adaptive internal model based closed loop system two times: the first time forwards-in-time, and the second time backwards-in-time with correct initial values for the state variables calculated using the result in the first run, perfect tracking in the whole period of the signals is achieved. Results on synthesized signals are presented and discussed.

Keywords: Polyphonic Pitch Estimation; Instantaneous Fourier Decomposition; Computational Auditory Scene Analysis;Adaptive Internal Model Theory; Frequency Tracking

ACKNOWLEDGEMENTS

I would first like to express my sincere appreciation to my supervisor, Dr. Lyndon J. Brown, for his guidance, patience and encouragement throughout the past two years of my study at Western. It has been an honor working with such an intelligent and gracious mentor.

Second, I would like to thank my peers for their support and help throughout the duration of my study at Western. Especially Jin Lu and Yan Ma, without their help in normal life and in research, I could never finish my thesis.

I also want to thank my family. Their love is always the source for me to seek power, to stay confident even when I am at the bottom of my life.

I can never express enough gratitude to my Lord, Jesus Christ, for being the foundation of my life in all situations. Just by taking a glance at the universe and the nature, his *elegance* in creating this world outweighs any engineer's work that we can only knees under him and praises.

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS

CWT	Continuous Wavelet Transform
DFT	Discrete Fourier Transform
EDS	Exponentially Damped Sinusoidal
EMD	Empirical Mode Decomposition
HHT	Hilbert-Huang Transform
IM	Internal Model
IMP	Internal Model Principle
MATLAB	MATrix LABoratory
MGM	Moving Global Median
MIDI	Musical Instrument Digital Interface
STFT	Short-Time Fourier Transform
TITO	Two-Input Two-Output
WT	Wavelet Transform

LIST OF SYMBOLS

contLTH	continuous fraction length threshold
e	error signals
K_i	gains of each internal model output
M_i	instantaneous magnitude of internal model i
matchProbTH	matching probability threshold
n	number of internal models
r	reference signals
u	summation of output of internal models
X_i	vector of two state variables of the internal model i
x_{1i}, x_{2i}	two state variable of internal model i

Chapter 1

Introduction

1.1 Music Background

In western music system, sounds with different frequencies is named by notes. For example, tone with fundamental frequency of $440Hz$ is named A, and with fundamental frqquency of $261.63Hz$ is named C. In total, there are 12 notes names: A, $A\#$, B, C, $C\#$, D, $D\#$, E, F, $F\#$, G and $G\#$, where normally $A\# = Bb$, $C\# = Db$, $D\# = Eb$, $F\# = Gb$ and $G\# = Ab$. The suffix "#" and "b" denote sharp and flat respectively. One interesting phenomenon of human brains in processing musical signal is that, if the frequency of one note is doubled, human brains can tell the pitch is higher but perceive the sound similar to the original one, while perceive all sounds with frequencies in between differently[1]. Due to this phenomenon, notes are assigned the same name if frequency distance is integer multiple of a certain frequency, but following with a different number. For example, $A4$ is $440Hz$, $A3$ is $220Hz$ and $A5$ is $880Hz$. The distance between the frequency of two sounds, called interval, is measured by dividing the frequency of one over the other. The notes in ideal western music system are logarithmic evenly distributed with the interval between any two adjacent

notes is $2^{1/12} \simeq 1.05946$. Such an interval is called a *semitone*. An octave is any interval between two frequencies when one is twice the other. An octave equals to 12 semitones.

1.2 Objectives

The purpose of this thesis is to develop new methods and algorithm for the task of *polyphonic pitch estimation* based on one previous student's work[2], and to propose a new method to track the beginnings of signal in a system to track predictable signals with uncertain frequency[4, 15]. The term *polyphonic pitch estimation* refers to the estimation of possible pitches in the polyphonic music signal that several music notes may occur simultaneously.

1.3 Motivations

Compared with speech signals, music signals are more complex, not only because it has a wider frequency range(almost the same with the frequency range of sounds that human ear can perceive, from $20Hz$ to $20kHz$) than the speech signals(the frequency range is narrowed from $50Hz$ to $4kHz$[3]), but also because some characteristics of music signals, such as *timbre*, make it difficult to be analyzed. In music signal processing, polyphonic pitch extraction plays an essential role so that it can be employed for the detection of melody(sequences of notes over time, or music score) and har-

mony(the relationship between concurrent notes). Polyphonic pitch estimation can also be utilized for various music application, including content-based music retrieval, interactive music system, low bit rate compression coding for music signal, and so on.

1.4 Main Contributions of the Thesis

The main contributions of this thesis can be summarized as follows:

1. Proposed advanced note extraction module based on previous work.

   Multiple changes compared with previous work have been made to identify music with multiple fundamental frequency(up to 4 concurrent notes), including the matching probability functions, the peak picking criteria, the algorithm flow, and the post-processing criteria.

2. Proposed evaluation criteria to evaluate the performance of the system.

3. Current system is able to identify music with multiple pitches played by various instruments, including violin, viola, cello, clarinet, oboe, and flute.

4. Proposed a brand-new unknown frequency tracking algorithm able to track the beginning of notes with minimal information.

1.5 Organization of the Thesis

This thesis is organized as follows. Chapter 2 is literature review summarizing the theory we are using in the thesis, existing time-frequency analysis methods and polyphonic pitch estimation methods in music transcription, and analyzing their advantages and problems. Chapter 3 reviews the recursive CASA based note extraction module for violin music in previous work, and presents a modified recursive multiple fundamental frequency estimation system based on CASA. Chapter 4 presents a method able to recover the information at the beginning of each segment of periodic signals with uncertain frequency. Chapter 5 concludes the thesis and discusses the future work.

Chapter 2

Literature Review

2.1 Internal Model Principles

The Internal Model Principle is first proposed by B.A.Francis and W.M.Wonham in 1976[6]. It was originally to eliminate periodic disturbances when designing control systems. They stated that perfect cancelation will be achieved if a suitably reduplicated model of the disturbance or reference signals is incorporated in the feedback loop. For an input sinusoidal disturbance or reference signals, this means that the controller should have a pair of poles on the $j\omega$-axis in the s-plane at a location corresponding to the frequency of the input signals[4]. The characteristic of internal model is that it supplies closed loop zeros which cancel the unstable poles of the disturbance or reference signals.

The basic block diagram of a control system with an IM is shown in Fig.2.1, in which $L(s)$ represents the process to be controlled, possibly combined with a standard controller. Here, the algorithm has been modified to be a signal processing algorithm by replacing the process to be controlled with a tuning function. L(s) can be selected to optimize the behavior of the algorithm. The input disturbance signal

$d = Acos(\omega_d t + \varphi)$, with the frequency to be identified is ω_d, the initial phase is φ, the output feedback error serving as the input to IM is e, the estimated frequency of d is ω, the two states of IM are x_1 and x_2, and the output of the IM identical to the input d is x_2. The transfer function from e to x_2 is $\frac{s}{s^2+\omega^2}$. The state space form of IM is given by

$$\begin{aligned} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} &= \begin{bmatrix} 0 & \omega \\ -\omega & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} e \\ y &= \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \end{aligned} \tag{2.1}$$

In steady state, we have

$$\begin{aligned} x_1(t) &= Asin(\omega_d t + \phi) \\ x_2(t) &= Acos(\omega_d t + \phi) \\ e(t) &= A_e sin(\omega_d t + \phi) \end{aligned} \tag{2.2}$$

The difference $\Delta\omega$ between estimate frequency ω and the actual frequency ω_d can be expressed as a non-linear function:

$$\Delta\omega = \omega - \omega_d \approx \frac{ex_1}{x_1^2 + x_2^2} \tag{2.3}$$

A simple integral controller could be utilized to force $\Delta\omega$ converge to zero as follows:

$$\frac{d\omega}{dt} = K_e \Delta\omega \approx K_e \frac{ex_1}{x_1^2 + x_2^2 + \varepsilon} \tag{2.4}$$

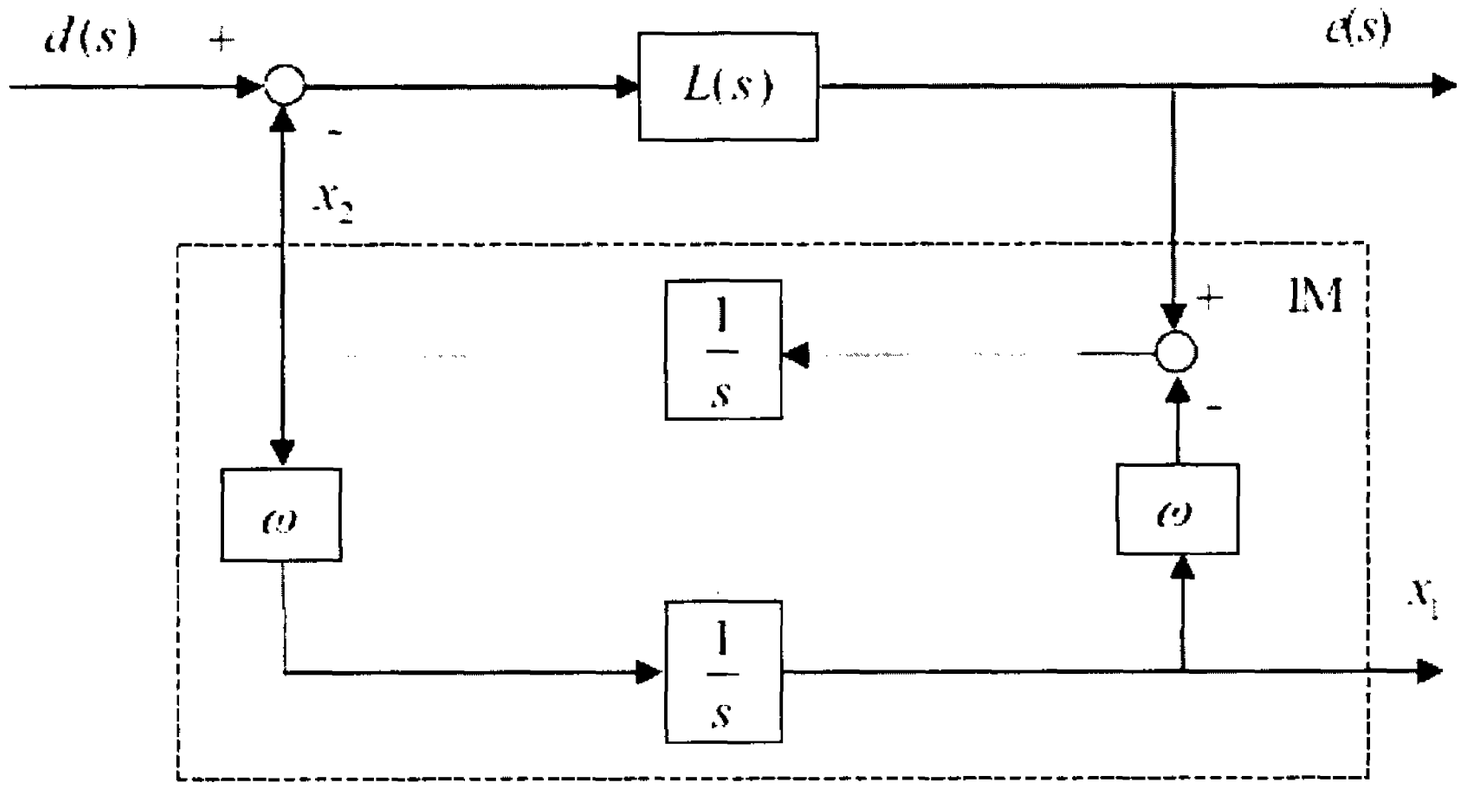

Figure 2.1: The basic block diagram of internal model for periodic disturbance cancelation

where K_e is the adaptation gain, and ε is a very small constant to guarantee no zero division problem. The stability and convergence of the algorithm are proven in [15].

2.2 Time-Frequency Analysis

Tine-Frequency Analysis is a generalization and refinement of Fourier analysis. The classic Fourier Transform is only applicable to stationary signals that maintain the same frequency with infinite duration. However, musical signals are non-stationary and time-varying. Its frequency components change with time. Also, according to uncertainty principle, it is impossible for a time-frequency analysis to have both best time resolution and frequency resolution at the same time. A piano has 88 keys with fundamental frequencies ranging from $A0(27.5Hz)$ to $C8(4186Hz)$. The frequency resolution should be relatively high in order to tell apart the lowest two piano notes

$A0(27.5Hz)$ and $A\sharp 0(29.14Hz)$, which means a relatively poor time resolution. But for some high frequency notes and fast paced music signals, the time frequency analysis requires a relative high time resolution. Therefore, there must be a tradeoff between frequency resolution and time resolution.

2.2.1 Short-Time Fourier Transform

Since classic Fourier Transform is not suitable to process non-stationary musical signals, an extension of the Fourier Transform, the Short-Time Fourier Transform is applied for musical signals. In order to obtain a joint time-frequency analysis, STFT cuts the signals into different frames or snapshot with possible overlap between adjacent frames. The STFT and its power spectrum named spectrogram, can be defined as below:

$$STFT(t,\omega) = \int_{-\infty}^{\infty} f(\tau)w(\tau - t)e^{-j\omega\tau}d\tau \tag{2.5}$$

$$SPECTROGRAM(t,w) = |STFT(t,w)|^2 \tag{2.6}$$

The STFT at time t is the Fourier Transform of a segmented local signal, obtained by multiplication of the signal $f(t)$ and a short window function $w(\tau - t)$ centered at time t. The shape of window function is one key factor of the STFT characteristics. The default window is the rectangle window, causing frequency leakage problem due to the signal's discontinuity at the edge of the window. Many other windows have been utilized to reduce the spectral leakage when processing music signals, such as

the Hanning[17] or the Hamming[18] window. By moving the window along the time axis, we can calculate the STFT at different time instants, and obtain a joint time-frequency analysis. Another important factor is the length of the window, or frame. The longer the frame, the better the frequency resolution, but with a poorer temporal resolution according to the uncertainty principle. As defined in Equation 2.5 , the window function of STFT is independent of the frequency ω, therefore, the temporal and frequency resolutions of STFT are the same in the time-frequency domain. This is one main drawback of STFT in music signal processing, since in music signal processing, it is usually required to have better time resolution at high frequency and better frequency resolution at low frequency. Multiple resolution Fourier Transform was investigated to solve this problem[20] [19].

2.2.2 Wavelet Transform

In compare with STFT, the Wavelet Transform provides a varying time-frequency resolution in the time-frequency domain. The Wavelet Transform(WT), or commonly named Continuous Wavelet Transform(CWT), can be defined as below:

$$WT(s,t) = \int_{-\infty}^{\infty} f(\tau)\frac{1}{\sqrt{s}}\phi^*(\frac{\tau - t}{s})d\tau \tag{2.7}$$

In order to make the reverse transform, the *admission condition* must be met:

$$C_\phi = \int_{-\infty}^{\infty} \frac{|\Phi(\omega)|^2}{|\omega|}d\omega < +\infty \tag{2.8}$$

where $\Phi(\omega)$ is the Fourier Transform of the wavelet function $\phi(\tau)$:

$$\Phi(\omega) = \int_{-\infty}^{\infty} \phi(\tau) e^{-j\omega\tau} d\tau \tag{2.9}$$

When the condition 2.8 is met, the signal $f(\tau)$ can be reconstructed by the inverse transform of CWT[21]:

$$f(\tau) = \int_{0}^{+\infty} \frac{ds}{s^2} \int_{-\infty}^{+\infty} WT(t,s)\Phi_{t,s}(\tau) dt \tag{2.10}$$

The CWT decomposes the signal into a time-frequency domain according to a continuously varying scale and translation and represents the signal with high redundancy, enabling us to performs an adaptive time-frequency analysis according to the music signal content. We can flexibly select a time-frequency resolution for different frequency bands, thus fulfilling the requirement of high temporal resolution at low frequency and high frequency resolution at high frequency in music signal processing.

2.2.3 Constant-Q Transform

Constant-Q Transform is related to the Fourier Transform. The Discrete Short-Time Fourier Transform is defined as follows:

$$X[k] = \sum_{n=0}^{N-1} W[n]x[n]e^{\frac{-j2\pi kn}{N}} \tag{2.11}$$

Given a data series, sampled at $f_s = \frac{1}{T}$, with T being the sampling period of the data. For each frequency bin, we can define following:

- Filter width, δf_k
- Q, the "Quality Factor", defined as follows:

$$Q = \frac{f_k}{\delta f_k} \tag{2.12}$$

with $\delta < 1$. The quality factor can be seen as the integer number of cycles processed at a center frequency f_k.

- Window length for the k-th bin is a function of the bin number

$$N[k] = \frac{f_s}{\delta f_k} = Q\frac{f_s}{f_k} \tag{2.13}$$

where f_s is the sampling frequency, and f_k is the center frequency of k-th bin.

Then, the Constant-Q Transform can be defined as follows:

$$X_{CQT}[k] = \frac{1}{N_k}\sum_{n=0}^{N_k-1}(win_{N_k}[n]x[n]e^{-j2\pi kn/N_k}) \tag{2.14}$$

Where win_{N_k} is a window function of length N_k. Since the Constant-Q Transform can geometrically space the center frequency, and the time resolution increase with frequency increasing, it has been explored on analyzing musical signal[30, 31]. How-

ever, following its definition in Equation 2.14, constant Q transform is relatively time consuming in calculation. An efficient calculation algorithm has been proposed by Brown and Puckette to solve this problem[32]. The algorithm is based on Fast Fourier Transform, and through calculating a sparse spectral kernel matrix, the number of multiplication greatly decreases, but its computational complexity is still higher than FFT.

2.2.4 Instantaneous Fourier Decomposition

The Instantaneous Fourier Decomposition approach was first introduced by Malhotra in [7], and developed by Sun [8], and Yan [2]. Like the Hilbert-Huang Transform(HHT)[13, 14], IFD decomposes the input signal into a sum of narrowband signals and applies the Hilbert Transform. As represented in Equation 2.2, at steady state, the two state variables of the internal model, x_1 and x_2 are sinusoidal and orthogonal. The instantaneous frequencies could be calculated. A complex analytic signal, $z(t)$ can be represented as:

$$z(t) = x_2(t) + ix_1(t) = a(t)e^{i\theta(t)} \tag{2.15}$$

Where

$$a(t) = \sqrt{x_2^2(t) + x_1^2(t)} \tag{2.16}$$

is the instantaneous amplitude of $z(t)$, and

$$\theta(t) = \arctan(\frac{x_1(t)}{x_2(t)}) \tag{2.17}$$

is the instantaneous phase of $z(t)$. Then the instantaneous frequency $\omega(t)$ can be calculated as:

$$\omega(t) = \frac{d\theta(t)}{dt} \tag{2.18}$$

As shown in Fig. 2.1 in section 2.1, the system with one basic internal model can only deal with a pure sinusoid signal. Unfortunately, in signal processing areas, most signals are not pure sinusoids. Take music signal processing for example, the input music signal may have multiple tones and multiple harmonics. Since any arbitrary periodic signal can be approximately represented by a Fourier series with finite terms, Sun[8] proposed employing multiple IMs in parallel in the feedback loop in order to track all the frequency component and decompose the input signal into a sum of sine and cosine pairs. The block diagram of IFD is shown in Fig.2.2 , where L is a tuning plant, and S is an arbitrary periodic signal which could be represented in the form of

$$S(t) = \sum_{i=1}^{n} S_i sin(\omega_i t + \varphi_i) \tag{2.19}$$

Where ω_i and φ_i are the frequency and initial phase of the $i^{th}(i = 1, 2, \ldots, n)$ frequency component in the signal $S(t)$.

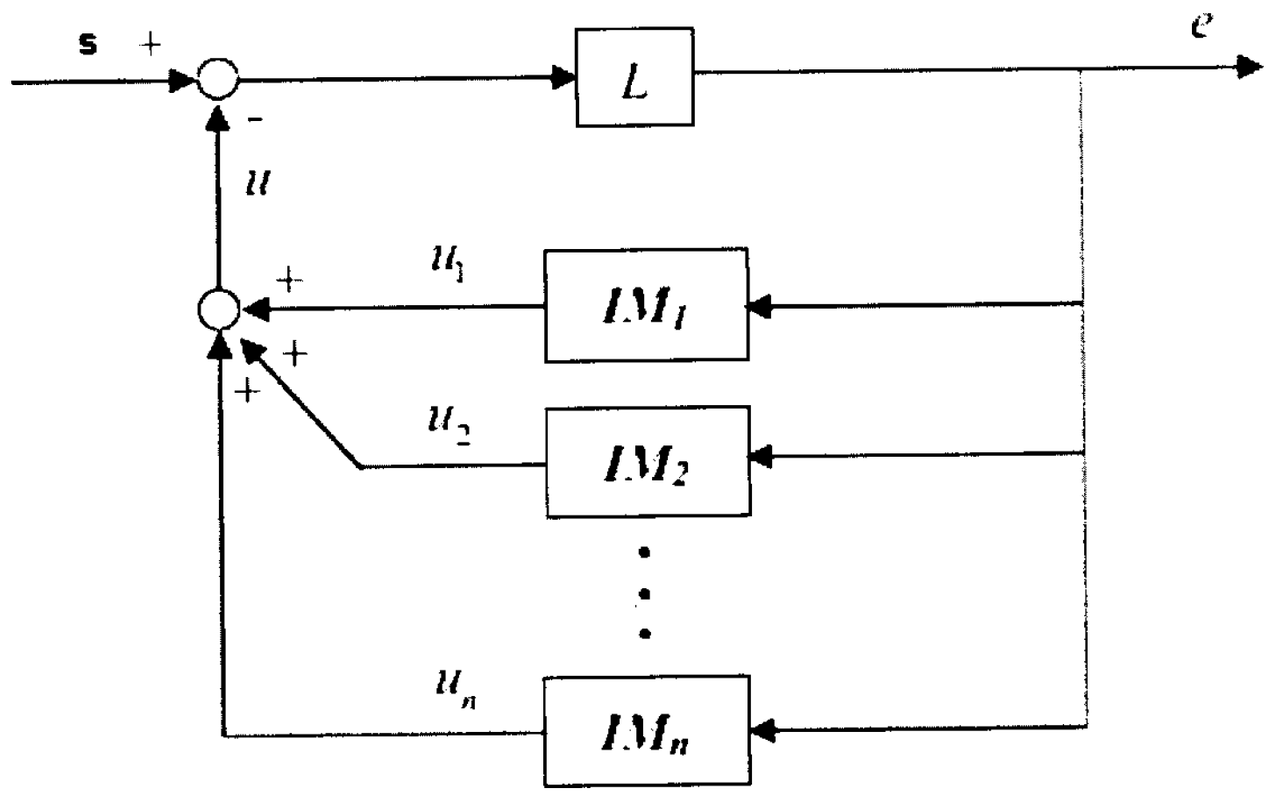

Figure 2.2: Block Diagram of Instantaneous Fourier Decomposition for an arbitrary periodic signal

Each IM can be represented in state space form as

$$\begin{bmatrix} x_{1i} \\ x_{2i} \end{bmatrix} = \begin{bmatrix} 0 & \omega_i \\ -\omega_i & 0 \end{bmatrix} X_i + \begin{bmatrix} 0 \\ 1 \end{bmatrix} e \tag{2.20}$$

$$u_i = \begin{bmatrix} K_{1i} & K_{2i} \end{bmatrix} X_i$$

Then in steady state, similar to Equation.2.2, we have

$$x_{1i}(t) = |x_i(t)|sin(\omega_i(t) + \phi_i) \tag{2.21}$$

$$x_{2i}(t) = |x_i(t)|cos(\omega_i(t) + \phi_i)$$

where

$$|x_i(t)| = \sqrt{x_{1i}^2(t) + x_{2i}^2(t)}, \phi_i(t) = arctan\left(\frac{x_{1i}(t)}{x_{2i}(t)}\right) - \omega_i t \tag{2.22}$$

With the parameters ω_i and $[K_1, K_2, \ldots, K_n]$ properly chosen such that the closed

loop system is stable, we can decompose the input complex signal using a finite number of IMs in parallel in the feedback loop. The feedback signal $u(t)$ is a summation of the output of each IM, with a form given by

$$u(t) = \sum_{i=1}^{n} u_i(t) = \sum_{i=1}^{n} (K_{1i}x_{1i}(t) + K_{2i}x_{2i}(t)) = \sum_{i=1}^{n} |K_i||x_i|cos(\phi_i) \tag{2.23}$$

where $|K_i||x_i|$ is the instantaneous amplitude of the input signal, and

$$\phi_i = arctan(\frac{x_1 i(t)}{x_{2i}(t)}) - arctan(\frac{K_{1i}}{K_{2i}}) \tag{2.24}$$

is the instantaneous phase of the input signal. Then instantaneous frequency could be calculated according to Equation2.18, and instantaneous Fourier decomposition is achieved. In order to get rid of the DC component and frequency component at higher frequency, Sun[8] proposed to incorporate a bandpass filter into the system. Thus, the desired system behaves as a bandpass filter with multiple notches, dealing with a small number of narrow-band signals, and being able to reject noise and isolate useful signals. Sun[8] applied the IFD algorithm to analyze real time experimental weld voltage data collected from a welding machine. In the application, at most 4 internal models are incorporated in parallel in the feedback loop to track the 4 frequency components(60Hz,180Hz,300Hz, and 420Hz) of the test signal. The algorithm is able to identify the input signals, eliminate the induced noise, and realize the Fourier decomposition of the input multi-tone signal. Sun[8] also states that, with more IMs

incorporated in the feedback loop, the more accurate the result will be. But in her test data, the power spectral energy is usually significantly decreased at higher order harmonics, thus only finite necessary IMs are incorporated. Also, the number of internal models is limited by the increasing difficulty of designing L.

2.3 Polyphonic Pitch Estimation

2.3.1 Introduction

Pitch is the perceptual and subject attribute of sounds, allowing them to be ordered from low to high on a frequency-related scale. Pitch plays a very important role in human understanding of music, since the auditory system tries to assign a pitch frequency to almost all kinds of acoustic signals.

Polyphonic pitch estimation, which can also be called multipitch estimation [9], or multiple fundamental frequencies estimation, is defined as "the task of estimating the fundamental frequencies of several concurrent sounds" [10]. This topic attracted a lot of research attention, and many methods have been proposed varying from traditional signal processing to machine learning, and integration of both. The following paragraph will first introduce the main problem in polyphonic pitch estimation, and then will discuss several most common existing methods.

2.3.2 Problems in Polyphonic Pitch Estimation

Polyphonic pitch estimation is very difficult and challenging because the spectrum of music signal spans in a wide range, and the spectral structure of different instrument varies a lot. The spectral structure is defined as the energy distribution on the fundamental frequency and all the harmonics of a music note. Different instruments may have totally different spectral structure. For example, music played by violin has the predominant energy on its fundamental frequency, while music played by piano, the highest energy may range from the fundamental frequency to the 4^{th} harmonic. Another problem in polyphonic pitch estimation is called inharmonicity. For ideal harmonic sound, the ratio of harmonic component to fundamental frequency should be integer. But for some instrument, such as piano, the ratio is not strictly integer. For music signal played by piano, the harmonic component could be expressed as follows:

$$f_n = nF\sqrt{1+\beta(n^2-1)}$$

Harmonic sharing is another problem in polyphonic pitch estimation. Consider for the most severe case, where two music notes with fundamental frequencies $f_1 = nf_2$, and n is an integer. The harmonic components of music note f_1 is completely overlapped by those of music note f_2, such as, the k^{th} harmonic of music note f_1 is the same with the $(nk)^{th}$ harmonic of music note f_2. For a general case, if two music note with

their fundamental frequencies having the following relation:

$$f_2 = f_1 \frac{n}{m}$$

where n and m are both integers. Then the $p^{th}(p = nk)$ harmonic of music note f_1 is the same with the $q^{th}(q = mk)$ harmonic of music note f_2. Taken the above case for example, consider there is a harmonic component in the mixture with the frequency $f = pf_1 = qf_2$, harmonic sharing problem makes it complicated to decide the energy distribution of this harmonic component on the two notes f_1 and f_2. Klapuri[24] proposed a method based on spectral smooth principle to solve the problem of harmonic sharing.

2.3.3 Auditory Model of Pitch Perception

Modern psychoacoustic research tries to build a human pitch perception model based on some known physiological and psychoacoustic knowledge. The two main theories include: the temporal theory and the spectral theory. The temporal theory uses temporal processing to detect the periodicity in different channels. In this theory, the acoustical signal is first processed by a frequency analyzer, and then the periodicity is detected by analyzing the time-domain envelop of the output signal in each channel. Unlike the temporal theory, the spectral theory transforms the pitch perception into pattern recognition processing of acoustic signals. In human auditory system, cochlear is an important portion of inner ear, acting as a frequency analyzer.

Sinusoid frequency components are picked by the spectral peaks in the spectrum. The positions of the relative spectral peaks form a position pattern which could be recognized by pattern recognition processing. Goldstein [11] proposed an optimum processor theory. In this theory, a central frequency analyzer is utilized to recognize the spectral pattern. A maximum likelihood statistical estimator determines which pitch best matches the spectral pattern of the acoustical signal.

2.3.4 Machine Learning Methods

Machine learning methods are introduced into polyphonic pitch estimation because recognizing a note from note mixture is a typical pattern recognition problem, and machine learning methods work well solving pattern recognition problems. Marolt introduced neural networks to build a polyphonic transcription system for piano system. In his system, the acoustic signal is first processed by a gammatone filter bank with 200 logarithmically-spaced frequency channels. In order to detect the periodicity, the output of the filter bank is further processed by adaptive oscillators to track the partial in each frequency channel. Thus, the network of the adaptive oscillators is used to track a group of harmonically relative partials. Finally, a combination of oscillator's network output and amplitude envelope of each channel is input into the neural network to recognize note. The system is tested with synthesis signals and real piano music. It is reported that the system performs better on synthesis signal than on real piano music.

2.3.5 Estimation Using Instrument Model

Yin proposed a music transcription system based on instrument model. The instrument model is defined as the harmonic structure of different instruments. In this system, FFT is first used in the front-end time-frequency analysis to generate amplitude spectrum. Then the amplitude spectrum is separated into 88 semitone bands with the spectrum from A0 to C8 (from 27.5Hz to 4.196kHz), covering the whole frequency spectrum of modern piano. The amplitude spectrum bins in each frequency band are then combined to generate the band energy spectrum $Z[i]$, $i = 1, 2, ..., 88$, where the index i corresponds to the MIDI(Musical Instrument Digital Interface, an industrial standard protocol for electronic music) note number of a western music note. For each music note i, $Z[i]$ denotes the energy of the fundamental, and the energy of the lowest 16 harmonics partials lie in the $Z[i]$, $Z[i+12]$, $Z[i+19]$, ..., $Z[i+48]$. A 49 number vector of $Z[i, ..., i+48]$ is considered as the instrument model of a music note at pitch i. It is assumed that the harmonic structure of an instrument is the same, regardless of the pitch and the transient. Based on this assumption, only one 49 number vector could completely signify the harmonic structure of music note for a certain instrument. For a music note with volume a and MIDI number p, the note spectrum could be generated as:

$$F(I, a, p) = \begin{cases} a \cdot I(i-p) & i \in [p, ..., p+48] \\ 0 & otherwise \end{cases}$$

Then, for an acoustic signal with band energy spectrum Z_M, the polyphonic pitch estimation is converted to resolve the following minimization problem:

$$Minimizing \left| Z_M - \sum_{i=1}^{n} F(I, a_i, p_i) y_i \right|^2$$

where n is the total number of estimated notes. The algorithm is evaluated by limited MIDI files and compared with the system proposed by Klapuri. It is believed that the performance could be improved if the acoustic signals have stable harmonic structure.

2.3.6 Iterative Methods

Klapuri[10] proposed a polyphonic pitch estimation algorithm based on the iterative method. In this method, the predominant pitch of concurrent musical signal is estimated, then all harmonics corresponding to it is removed from the musical signal. The process is repeated iteratively on the residual signal until all the harmonic sounds have been detected. The input acoustic signal is first preprocessed to suppress the noise by a magnitude warping method. The suppressed spectrum is then input to the predominant pitch estimation module. To find the predominant pitch, the spectrum is first separated into 18 different frequency bands logarithmically spaced between $50Hz - 6kHz$. In each frequency band, a corresponding weight vector is calculated to represent the likelihood of a certain pitch. Then, the bandwise likelihood weight vectors are combined to globally estimate the predominant pitch across the all frequency bands. As the predominant pitch is estimated, all its harmonics

will be subtracted from the input acoustic signal according to spectrum smoothness principle. The pitch estimation and subtraction procedure will be repeated on the residue signal until all the harmonics have been detected. This algorithm has been tested on polyphonic acoustic signals with multiple timbres, and performs well.

2.3.7 Computational Auditory Scene Analysis

Bregman first discussed Auditory Scene Analysis in his book [23]. He described the mechanisms that how human auditory system perceiving and recognizing sound sources from a complex sound mixture. He also states that it is important for sound separation, since after each sound source is separated, one could use conventional fundamental frequency estimation method to identify all the pitches for each group. The sound separation procedure include two stages, first is to decompose the acoustic signal to time-frequency spectrogram, then to group all the time-frequency components from each sound source based on the grouping principles, including proximity in frequency and time, periodicity/harmonic frequency relationship, continuous or smooth transition, onset and offset, amplitude and frequency modulation, rhythm and common spatial location/same direction of arrival[22]. Computational Auditory Scene Analysis(CASA) has many applications in music signal processing, previous works includes those by Mellinger[25], Kashino etc[26, 27], Godsmark and Brown[28], Sterian[29].

2.3.8 Recursive CASA based Note Extraction

Yan[2] propsed a recursive approach based on CASA for multiple fundamental frequency estimation for music played by solo violin. In her algorithm, the original music signal is first transformed to a time-frequency magnitude spectrogram, using a method called Instantaneous Fourier Transcription proposed by Malhotra in 2005[7] and developed by Sun in 2006[8]. The input music signal is separated as monaural music frames at a sampling rate of $44.1kHz$. Each frame has length of $10ms$. The magnitude spectrogram is also averaged for every $10ms$ frame, and then used as the input for note extraction. Note extraction is done recursively. First magnitude spectrogram is transformed to peaks for peak picking. Only the local maxima above given threshold are selected as peaks among the magnitude spectrogram. Those selected peaks are all normalized having a magnitude of 1, and the rest having a magnitude of 0. Then, a group of instrument dependent probability functions called Matching Probability Functions are utilized to extract notes from this frame level peak representation. The Matching Probability Functions quantize the probability of any notes presented in the music signal. The value ranges from 0 to 1. The higher the value, the higher the probability note presented at this music frame. During the first recursive loop, silent segment, single note fractions with a probability higher than 99% and Glissandi note with a probability higher than 95% are identified. The residual signal frames are repeatedly processed, with the identification thresholds, including continuous fraction length threshold *contLTH* (initially $70ms$), and matching proba-

bility threshold *matchProbTH* (initially 70%), adjusted to lower standard. *contLTH* will decrease to 50*ms*, and *matchProbTH* will decrease to 30%. The whole process will not stop until all the music frames are identified, or the residual signal is too complex to analyze. At the end, a postprocessing method is applied on the result to remove apparent errors and connect glissandi notes. This algorithm is tested on synthesized signals, midi music, and real recording of violin music. The experiment result is demonstrated as note number versus time for the estimated result. However, she does not proposed a proper evaluation method to evaluate the accuracy, error rate. Also, this algorithm deal with solo violin, meaning it can not handle music signals with more than 2 concurrent notes.

2.3.9 Modified Multiple Fundamental Frequency Estimation based on Recursive CASA

This work is using the major framework of Yan[2]'s system. In order to process music signal with more than 2 concurrent notes and improve the system performance, the algorithm is modified from the original algorithm. Also, we proposed a proper evaluation criteria to evaluate the performance of the system. The details of the Modified Multiple Fundamental Frequency Estimation based on Recursive CASA is organized as follows. In Section 3.2.1, we will discuss how we use Instantaneous Fourier Decomposition to transform original music signal into magnitude spectrogram, a form of time-frequency representation of desired resolution. In Section 3.2.2, the main

changes on Note Extraction will be illustrated, and the performance before and after the change will be demonstrated. In Section 3.2.3, a proper evaluation will be proposed to evaluate the performance of the result. In the final Section 3.2.4, the algorithm is tested on a group of MIDI music signals played by various instruments, and evaluated using the criteria proposed in Section 3.2.3.

Chapter 3

Polyphonic Pitch Estimation

3.1 Introduction

Polyphonic pitch estimation, which can also be called multipitch estimation [9], or multiple fundamental frequencies estimation, is defined as "the task of estimating the fundamental frequencies of several concurrent sounds" [10]. In some cases, polyphonic pitch estimation is simplified to note recognition [12], the recognition of musical notes present in the music sound. The existing proposed methods have been discussed in Section2.3. Since this work is based on Y. Ma's work[2], with several modification to improve the system performance, the following will mainly discuss the difference from this work to Y. Ma's work, and how much the system performance is improved.

3.1.1 Recursive CASA based Note Extraction

Y. Ma[2] proposed a recursive approach based on CASA for multiple fundamental frequency estimation for music played by solo violin. In her algorithm, the original music signal is first transformed to a time-frequency magnitude spectrogram, using a

method called Instantaneous Fourier Transcription proposed by Malhotra in 2005[7] and developed by Sun in 2006[8]. The input music signal is separated into monaural music frames. Each frame is represented by a single point in time by averaging over all time points in the frame, and then used as the input for note extraction. Note extraction is done recursively. First magnitude spectrogram is transformed to peaks for peak picking. Those selected peaks are all normalized having a magnitude of 1, and the rest having a magnitude of 0. Then, a group of instrument dependent probability functions called Matching Probability Functions are utilized to extract notes from this frame level peak representation. The Matching Probability Functions quantize the probability of any notes presented in the music signal. The value ranges from 0 to 1. The higher the value, the higher the probability of the note being present at this music frame. During the first recursive loop, silent segments, single note fractions with a probability higher than 99% and Slur note with a probability higher than 95% are identified. The residual signal frames are repeatedly processed, with the identification thresholds, including continuous fraction length threshold $contLTH$ (initially $70ms$), and matching probability threshold $matchProbTH$ (initially 70%), adjusted to lower standard. $contLTH$ will decrease to $50ms$, and $matchProbTH$ will decrease to 30%. The whole process will not stop until all the music frames are identified, or the residual signal is too complex to analyze. At the end, a postprocessing method is applied on the result to remove apparent errors and connect Slur notes.

This algorithm is tested on synthesized signals, midi music, and real recordings. The

experiment result is demonstrated as note number against time for the estimated result. However, she does not proposed a proper evaluation method to evaluate the accuracy, error rate of her algorithm. Also, this algorithm dealt only with solo violin, which allowed it to assume no more than two notes could be played simultaneously.

3.2 Modified Multiple Fundamental Frequency Estimation based on Recursive CASA

This work is using the major framework of Y. Ma[2]'s system. In order to process music signal with more than 2 concurrent notes and improve the system performance, the algorithm is modified significantly from the original algorithm. Also, this paper proposed a proper evaluation criteria to evaluate the performance of the system. The details of the Modified Multiple Fundamental Frequency Estimation based on Recursive CASA is organized as follows. In 3.2.1, we will discuss how we use Instantaneous Fourier Decomposition to transform original music signal into magnitude spectrogram, a form of time-frequency representation with desired resolution. In 3.2.2, the main modification on Note Extraction will be illustrated, and the performance before and after the change will be demonstrated; also a proper evaluation will be proposed to evaluate the performance of the result; finally the algorithm is tested on a group of MIDI music signals, and evaluated using the proposed criteria.

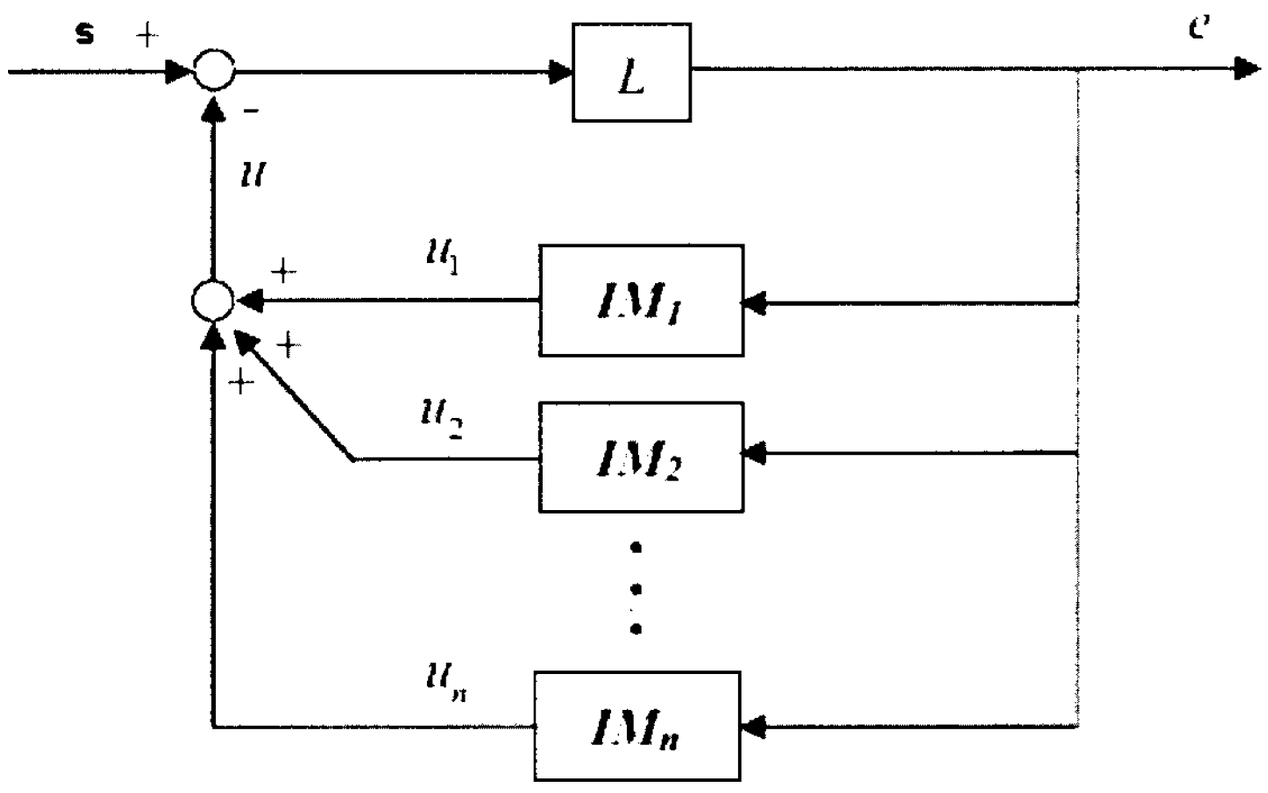

Figure 3.1: Block Diagram of Instantaneous Fourier Decomposition

3.2.1 Time-Frequency Analysis

The Time-Frequency Analysis is performed using Instantaneous Fourier Transform to transform the input signal into a magnitude spectrogram. As illustrated in Figure 3.1, the discrete-time(sampling time is normalized to 1 in practise) state-space form of IFD is as follow:

$$X_i(T+1) = \begin{bmatrix} cos(\omega_i) & sin(\omega_i) \\ -sin(\omega_i) & cos(\omega_i) \end{bmatrix} X_i(T) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} e(T) \tag{3.1}$$

$$u_i(T) = \begin{bmatrix} K_{1i} & K_{2i} \end{bmatrix} X(T) \tag{3.2}$$

where

$$X_i(T) = \begin{bmatrix} x_{1i}(T) & x_{2i}(T) \end{bmatrix}^T \tag{3.3}$$

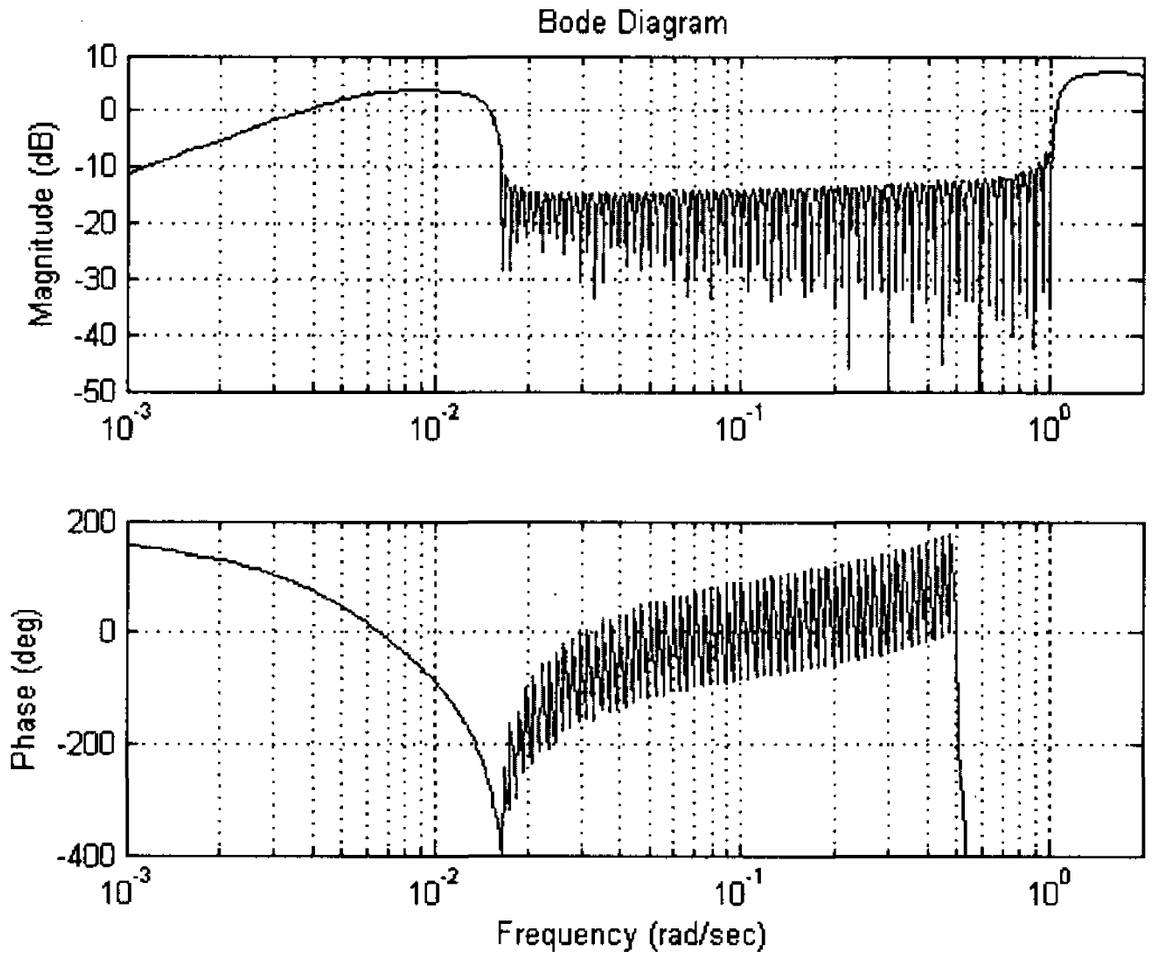

Figure 3.2: Bode diagram of the theoretical IFD system, with normalized digital frequency $[10^{-3}, \pi]$

Corresponding to Equation 2.23, its instantaneous magnitude is as follows

$$M_i(T) = |K_i||X_i| = \sqrt{K_{1i}^2 + K_{2i}^2} \cdot \sqrt{x_{1i}^2 + x_{2i}^2} \tag{3.4}$$

As mentioned in 2.2.4, the desired IFD system behaves like a band-pass filter with multiple notches, with signal s as the input, and error signal e as the output. The bode diagram of the theoretical system is illustrated in Figure 3.2. This bode diagram is plotted in MATrix LABoratory(MATLAB) version 7.0.4, since the sampling frequency is 44.1kHz, the normalized digital frequency π is corresponding to the frequency 22.05kHz. We set the number of the notches to be 72, with the central frequencies of the notches spanning from $116.54Hz(A\#2)$ to $7.04kHz$, and the cut off frequency set as 2/3 of the frequency of the first notch($116.54 * 2/3 = 77.69Hz$)

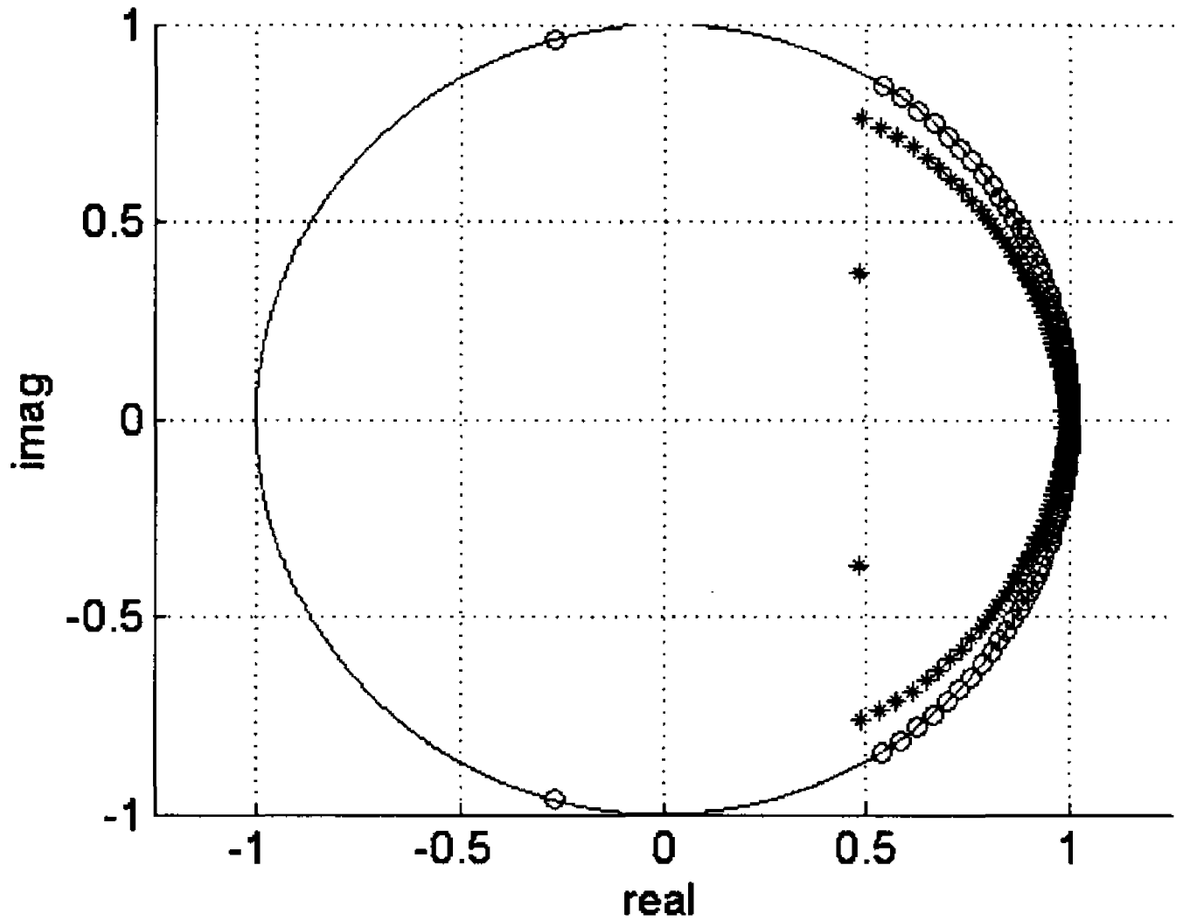

Figure 3.3: The zero-pole location of the system in z-domain

and 3/2 of the frequency of the last notch($7.04 * 3/2 = 10.56kHz$). Due to poor numeric properties of difference equations and precision limitation of MATLAB in drawing the diagram, the notches are shallow with largest only $-20dB$, while they are in reality $-\infty$.

The poles and zeros position of the desired system is shown in Figure 3.3. All the poles are within the unit circle, thus the system is stable. However, although the system is stable, the poles are still close to unit circle. The radius of the furthest pole is 0.9977, rendering the system responses very slow. This will be discussed in next section.

By properly setting the parameters of the plant L and the gains K_{1i} and K_{2i} for each IM_i, the band-pass filter with multiple notches is realized by a closed-loop system with the plant L as the control panel, and paralleled IMs as the feedback. The

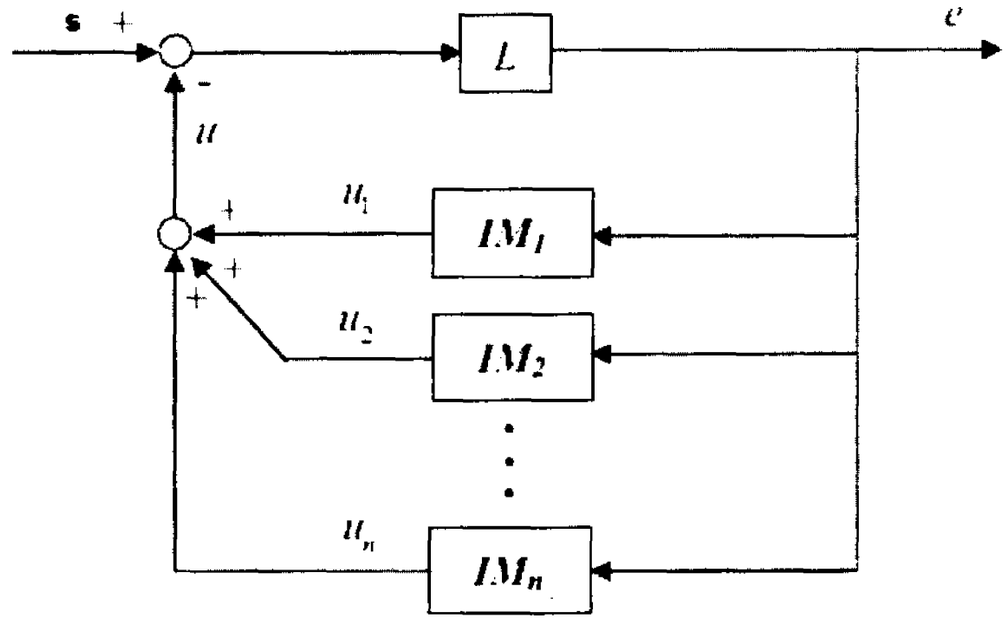

Figure 3.4: System block diagram

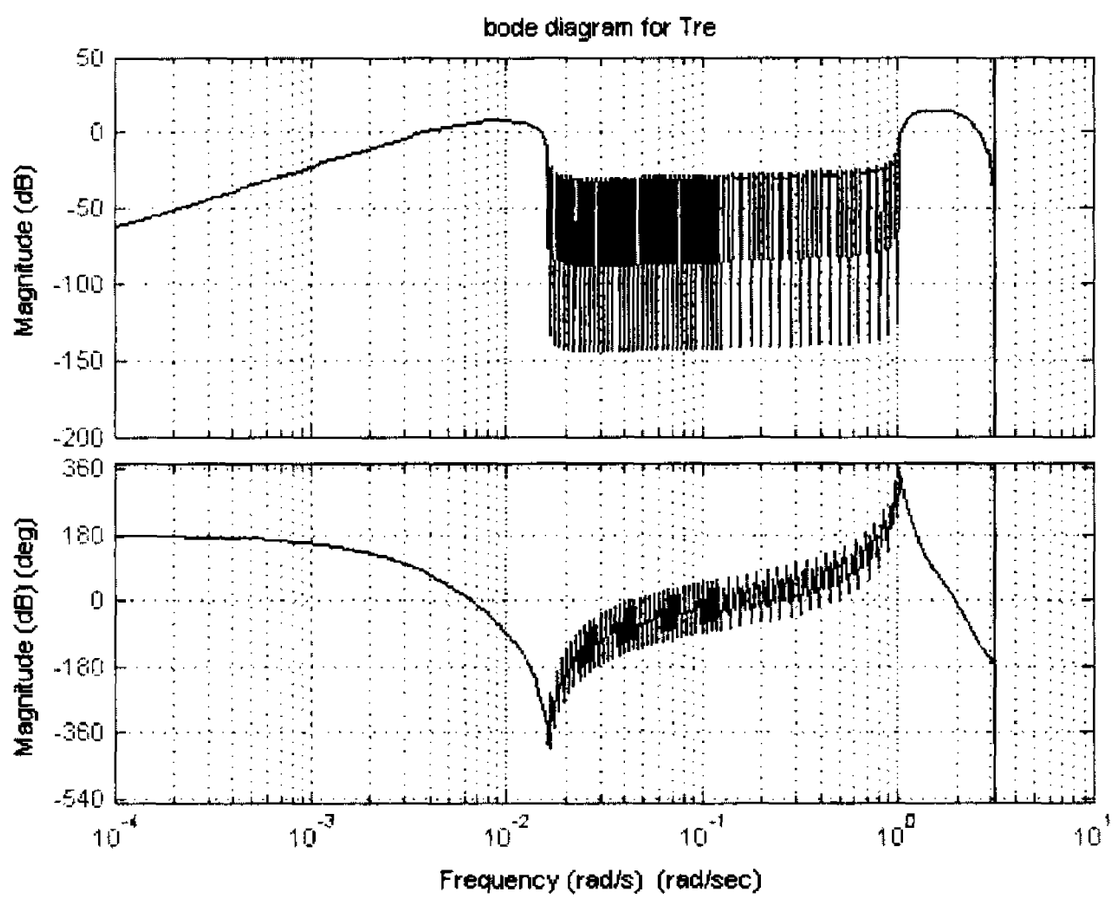

Figure 3.5: Bode diagram of IFD system from r to e

detailed designing method could be found in Appendix A of Y. Ma's paper[2].

The system block diagram illustrated in Figure 3.4, and the bode diagram for the actual system is shown in Figure 3.5. The output of each IM has the form of $u_i = K_{1i}x_{1i} + K_{2i}x_{2i}$, representing a replicated model of the frequency component of the input signal. As long as the frequency range of the group of IMs covers the spectrum of the input signal, the output of the input signal subtracting the output

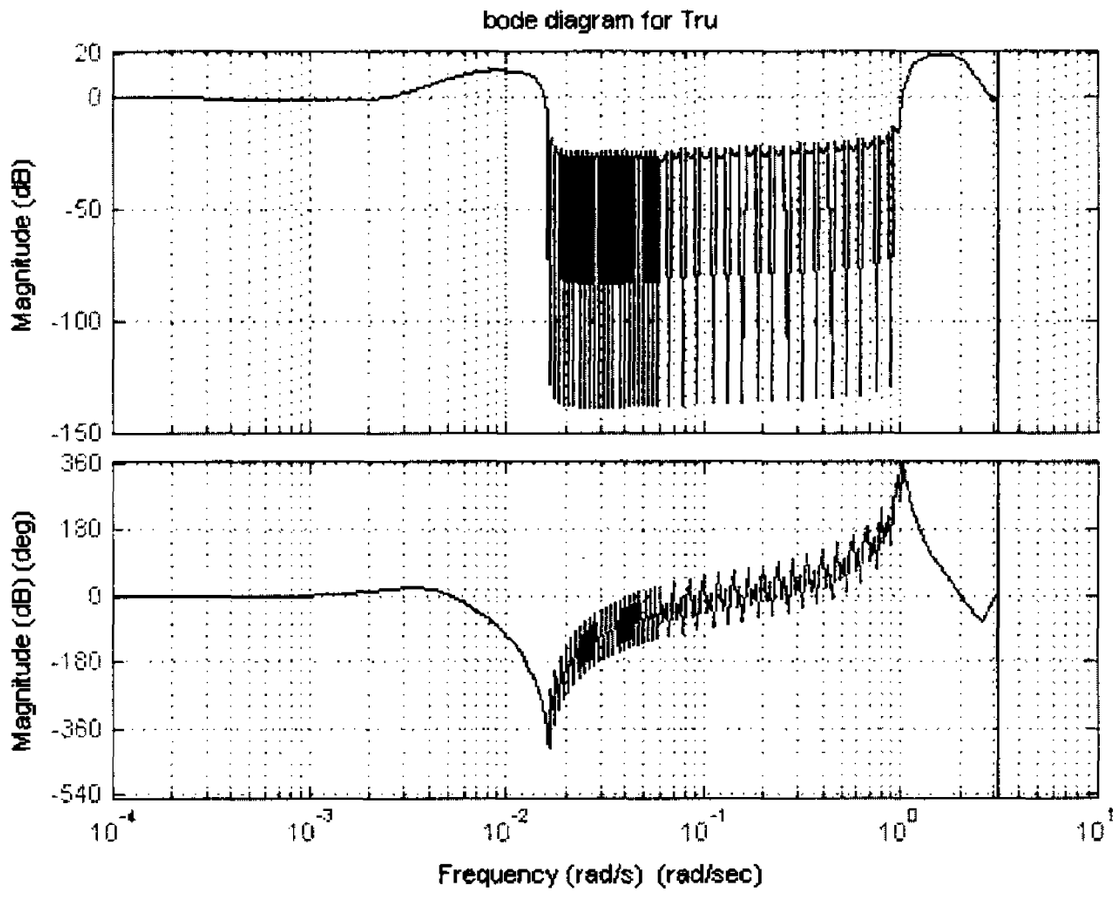

Figure 3.6: Previous bode diagram of IFD of $r - u$

of the group of IMs should converge to zero.

This system is designed using the method in Y. Ma's paper, and it has one obvious drawback: it may lead to a considerable positive gain at frequencies where noise is present. As mentioned above, the error of the system should be the output of the input signal subtracting the sum of the feedback output. The bode diagram from the input signal to the error is illustrated in Figure 3.6. From Figure 3.6, the gain in the pass band has a gain at most $-20dB$, which means the pass band will not introduce noise since it always has a negative gain; however, at the shoulders of the passing band, the gain has a considerable positive gain(at most $18dB$). If the signal has a frequency component beyond the spectrum of the system, it will be present in the error signal with about 8 times amplified. This will introduce a lot of noise in the error signal, making it hard to analyze.

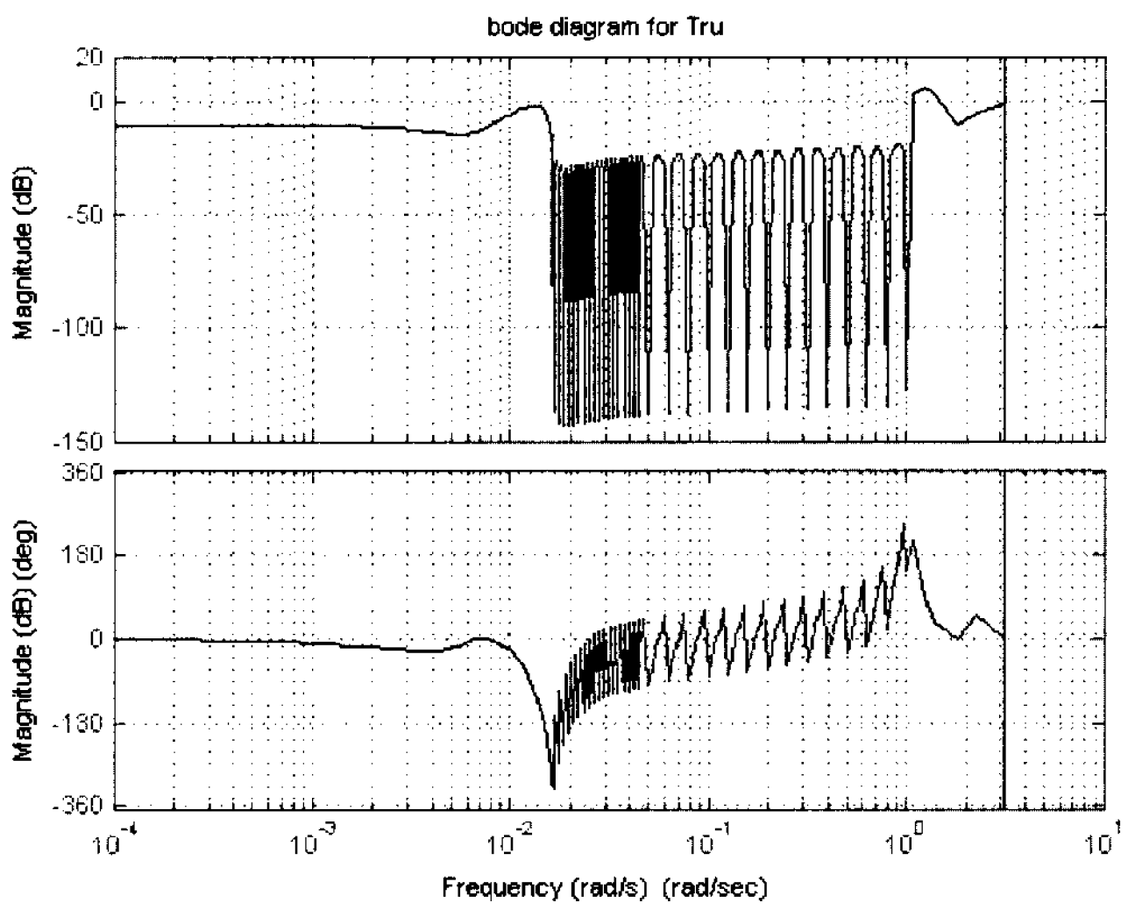

Figure 3.7: Modified system bode diagram of $r - u$ with Chebyshev II filter

To overcome this drawback, we modified the band-pass filter design. Instead of using the Chebyshev type I bandpass filter, a Chebyshev type II bandpass filter was employed. The main difference of these two types is that, Chebyshev II filter does not roll off as fast as Chebyshev type I filter, and has no ripples in the pass band but has ripples in the stop band. By changing Chebyshev type I filter into Chebyshev type II filter, we want to depress the gain at the shoulders of the passing band since Chebyshev type II filter has more slow varying shoulders with ripples. The modified system bode diagram is illustrated in Figure 3.7. As shown in Figure 3.7, the gain at the shoulders of the passing band is quite small(at most $5dB$ at high frequency) compared with the previous one(around $18dB$) in Figure 3.6. Therefore, by changing the original Chebeyshev type I bandpass filter into Chebeyshev type II bandpass filter, the modified system has a better performance with the ability of rejecting more

WOODWIND	
Flute	$250Hz - 2.5kHz$
Oboe	$250Hz - 1.5kHz$
Clarinet	$125Hz - 2kHz$
BRASS	
Trumpet	$170Hz - 1kHz$
STRINGS	
Violin	$200Hz - 3.5kHz$
Viola	$125Hz - 1kHz$

Table 3.1: Spectrum of instruments processed in the algorithm

noises.

3.2.1.1 Experimental Result on MIDI Music

Instantaneous Fourier Decomposition is applied on a piece of MIDI music to demonstrate the effect. MIDI stands for Musical Instrument Digital Interface, which is a industrial-standard protocol for electronic music defined in 1982 [33]. The spectrum of most woodwind, brass, and string instrument is illustrated in Table 3.2.1.1.

The system parameters are set in accordance with the characteristics of these instruments. The whole spectrum is in range from $116.54Hz(A\#2)$ to $7.04kHz$, a total of 72 notes (6 octaves). Theoretically, the more the number of the notches or notes, the wider spectrum the system can cover. For example, to cover the whole spectrum of piano, the spectrum should span from $27.5Hz$ to at least $4.186kHz$(the fundamental frequency of the highest note), which covers 88 semitones, more than 7 octaves. Due to the relatively poor numerical properties of difference equations, continuous time implementations can be used with greater IM's at the expense of

substantially higher computational burden.

Compared with Y. Ma's work[2], which only deals with violin, and its spectrum has a total of 60 notes(5 octaves), spanning from $196Hz$ to $5.92kHz$, this system has a wider spectrum range, enabling it to analyze more instruments which have a wider spectrum, and more harmonics for some notes, especially for high frequency notes. For example, the highest fundamental frequency of violin is $3.5kHz$, Y. Ma's work can only include the fundamental frequency, while this system can analyze its $2nd$ harmonic.

The central frequencies of each IM is set as the theoretical frequency of musical note, thus each note has an individual IM to track the energy change on its channel. The ratio of notch width to its central frequency is 0.1; compared with the distance between adjacent notes $2^(1/12) - 2 = 0.0595$, there will be some overlap between adjacent notches.

We test the time frequency representation on a piece of midi music, with a length about $8sec$ and played by flute. The time frequency representation is shown with grey scale magnitude in Figure 3.8.

3.2.2 Modified Note Extraction

3.2.2.1 Simplification and Condition

In music signal processing, based on common sense and spectral smoothness principle, it is not necessary to estimate F0s at each sampling moment. The signal is

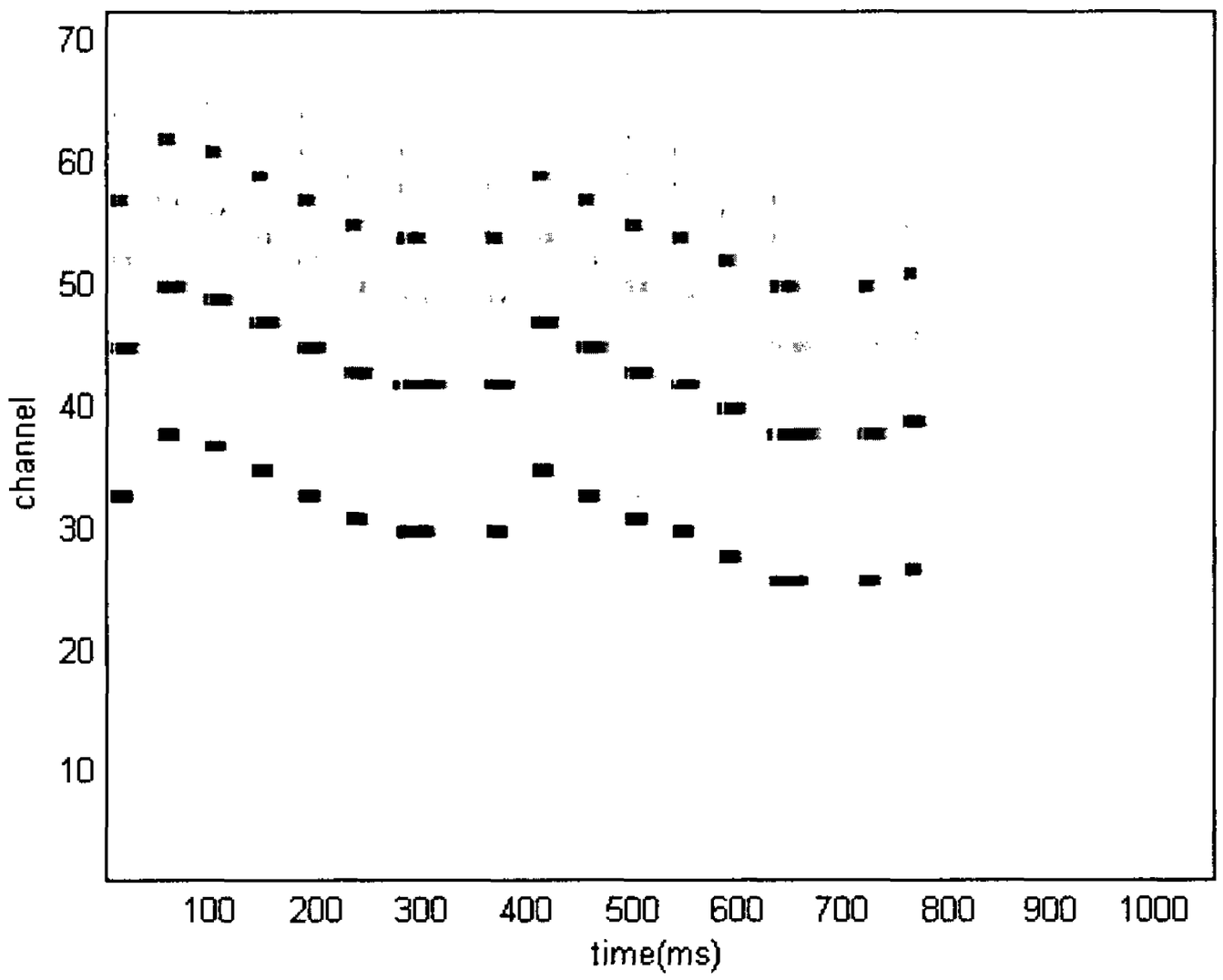

Figure 3.8: Time frequency representation with grey scale magnitude

normally segmented to small fractions, or frames. The signal is assumed to be stationary and treated as one entity for each fraction. The length of each fraction is set $10ms$ because it is believed human ears can not separate any two transients with less distance[34]. Thus we transform the time frequency magnitude spectrogram into a fraction-semitone band representation of the music signal.

We also assume that if we can decide which semitone band contains the fundamental frequency component, the corresponding note is present at that fraction. Combined with Matching Probability Function which will be discussed below, this simplification enables us to ignore note pitch error and overcome inharmonicity in most cases.

The goal of this method is to find F0/F0s at each fraction of the signal. Silent fraction in music is regarded as no F0 present. To simplify the estimation, we assume

the first 5 fractions and the last 5 fraction of the music signal are silent segments. To guarantee all music signals we are dealing with fulfill this simplification, we add a $50ms$ zero segment to the start and the end of the original music signal. In the evaluating process, these two segments are remove to make sure the actual notes and the estimated notes have the same length.

3.2.2.2 Algorithm Description

The system diagram of our note extraction algorithm is shown in Figure 3.9. A group of instrument dependent probability functions, called Matching Probability Functions, are designed to quantize the probability of any note/notes present in any fraction. The functions are designed to guarantee that, for any note/notes and any fraction, its value ranges from 0 to 1. The higher the value, the higher probability the note/notes present at this fraction.

This algorithm is applied on the result of time-frequency analysis – the magnitude semitone band spectrogram. The algorithm starts with Peak Picking (Section 3.2.2.3 to extract useful data from the fraction-level magnitude-semitone band representation. The initial step before the loop is to identify the silent fractions, note/notes fractions with 100% matching probability. These identified fractions are marked as identified by setting a variable /textsldone in the program from -1(the default value, in the initial, all fractions are marked as unidentified) to 1 for note/notes fractions, or from -1 to 0 for silent fraction, while other fractions remain un-identified($done = -1$). An

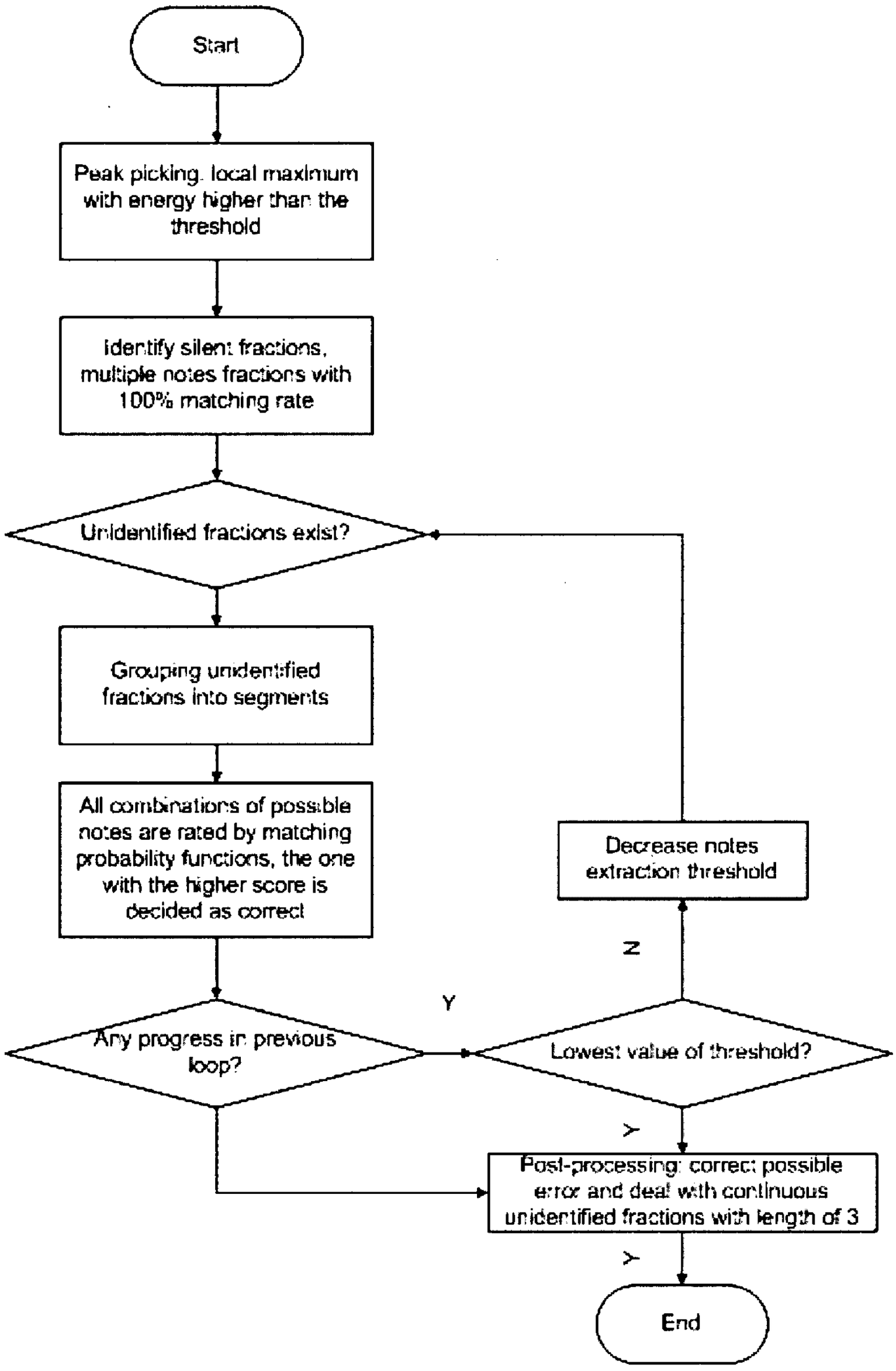

Figure 3.9: Algorithm diagram of note extraction system

iterative loop then begins to process those un-identified fractions. These un-identified fractions are grouped into continuous 5-fraction length segments. If we have less than 5 continuous un-identified fractions, they will not be grouped into a segment, and will be processed in the post-processing step at the end of the algorithm. If we have continuous un-identified fractions with a length which is not the integer multiple of 5, we will increase the length of the segment for the last un-grouped fractions. For example, if we have 18 continuous un-identified fractions, 2 segments will be formed by picking the first 10 un-identified fractions, and the last segment will have a length of 8 fractions to include the last 3 fractions. The initial matching probability threshold *matchProbTH* is set as 70%. If no process has been made after a loop, the matching probability threshold will be decreased gradually until they reach the lower limit of 30%. The identification process continues until all fractions are identified or the un-identified segments are too complex to analyze. At the end, a post-processing method is applied on the result to remove apparent errors, and to deal with those continuous un-identified fractions with length less than 5.

3.2.2.3 Peak Picking

Peak picking helps remove possible noise from the fraction-semitone band representation and leaves only those segments with high enough energy to be noticed. Peaks are those local maximum segments with energy above given thresholds. The rest are deemed as zero. Two thresholds are used to determine the presence of a peak. A

peak is deemed to present in a music fraction only if the energy at its semitone band is higher than both thresholds.

The first threshold is call semitone band median. Generally, for any fraction, all major harmonics(fundamental to $10th$ harmonics, as explained in Section3.2.2.4) of cocurrent notes will cover only a small part of the semitone bands. For example, in single note music segment, the major harmonics cover no more than 10 semitone bands; in two notes music segment, the major harmonics cover no more than 20 semitone bands; while the total number of semitone bands is 72. Thus, the majority of the 72 semitone bands contain only noise, and their energy is normally lower than those contained in major harmonics of the present notes. If we use the median of all semitone bands magnitude at this time fraction as a threshold, it is safe to assume the component at a semitone band with energy lower than the semitone band median is noise.

In Y. Ma's paper, another threshold she proposed is global median, which is the median of the magnitude in the fraction-semitone band representation at all time. In implementation, twice of the global median is used as the second threshold. However, since most music signal is not energy constant signal, each note having a transient period including a rising part when energy rise from 0 to stable state, and a falling part when energy drops back to 0, choosing the global median is not a good choice because it will sometimes fail to pick up those peaks with energy near 0 at transient period.

To overcome this drawback, this paper proposed a method to use the Moving Global Median(MGM) as the second threshold. A moving window is sliding along the time axis to select the time fractions, which will be used to calculated the MGM for a certain time fraction. The window is set as $5 - fraction$ length, since we assume that each note should last at least $50ms$. At each time fraction, the moving global median is calculated as the median of the magnitudes of the previous 2 time fractions, the current time fraction, and the next 2 time fractions. Since we add 5 zero segment at the beginning and the end of the signal separately, the moving global median for the first 5 fractions and the last 5 fractions are all 0.

The result of peak picking for a single note flute midi music is shown in Figure 3.10. It is seen from this figure that, there are misidentified peaks around the time of $3.5s$, $7.0s$, $7.17s$, $7.59s$ and $8.0s$. These identified peaks do not belong to the real signal. It is also noticed that some peaks belong to higher harmonics are lost in this process. These kind of mistakes will be corrected in following steps.

In order to compare the performance of global median and moving global median, two more experiments are carried out as shown in Figure 3.11 and 3.12. With all the parameters set the same, only the second threshold is different, one set as the global median, and the other set as the moving global median with the moving window length set as 5. As shown in the amplified figure 3.12, the actual note starts from the time of $2.65s$ to $3.44s$. The result using global median as the 2nd threshold estimates the note starting from the time of $2.68s$ to $3.35s$, missing 12 notes, while the result

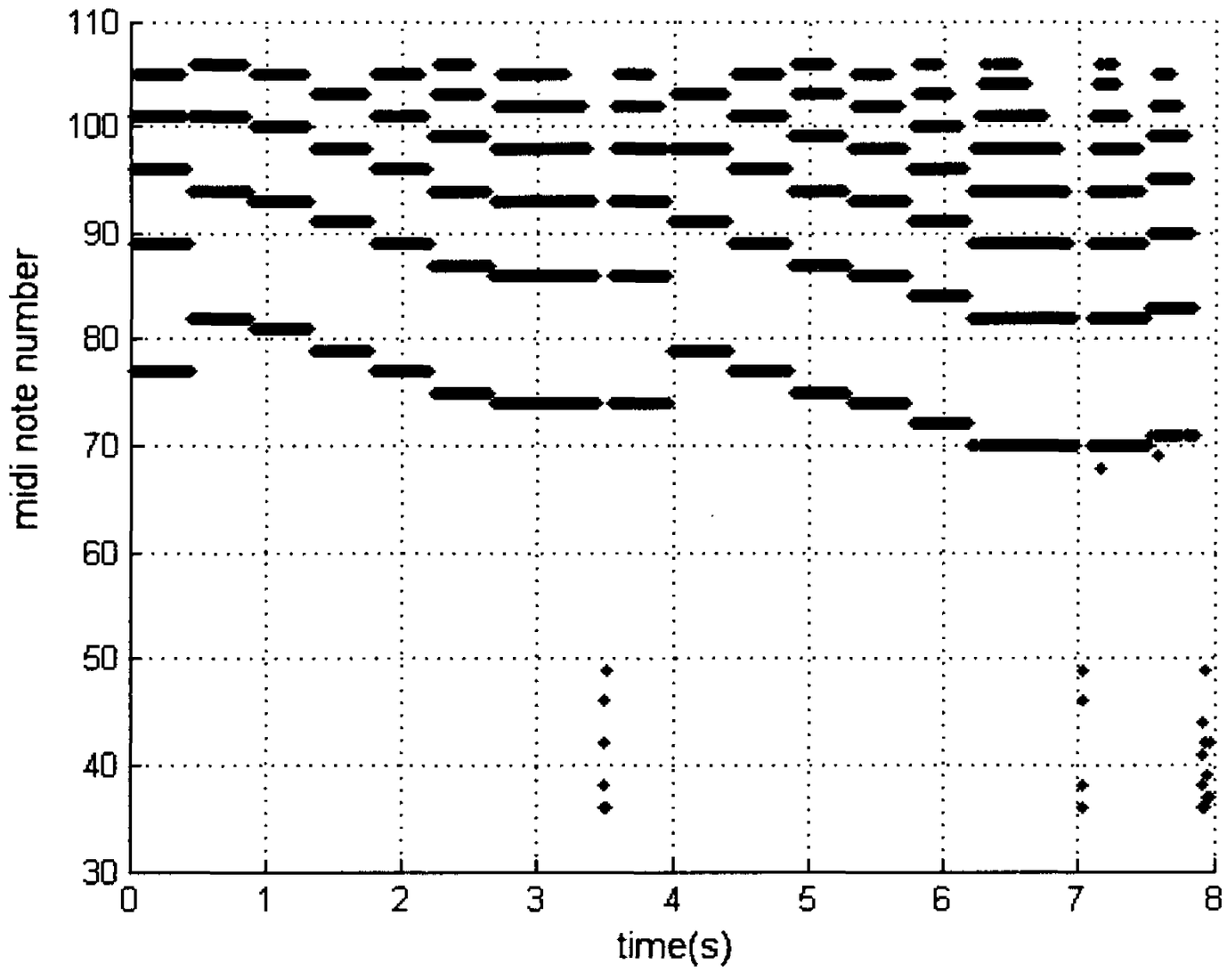

Figure 3.10: Peak pick result of flute midi music

using moving global median as the 2nd threshold estimates the note starting from the time of $2.67s$ to $3.44s$, missing only $1note$. Generally, for fast energy changing music signal, using moving global median as the 2nd threshold could identify about 5 more time fractions($2-3$ at the beginning, and $2-3$ at the end) than the one using global median as the 2nd threshold in one single note duration. This improvement is very important because in fast-pacing music, notes are usually short and fast-changing. Missing about $50ms$ for each note will dramatically deteriorate the system performance by decreasing the Recall Rate(this is the evaluation method and will be discussed in Section 3.2.3) with so many True Negative notes.

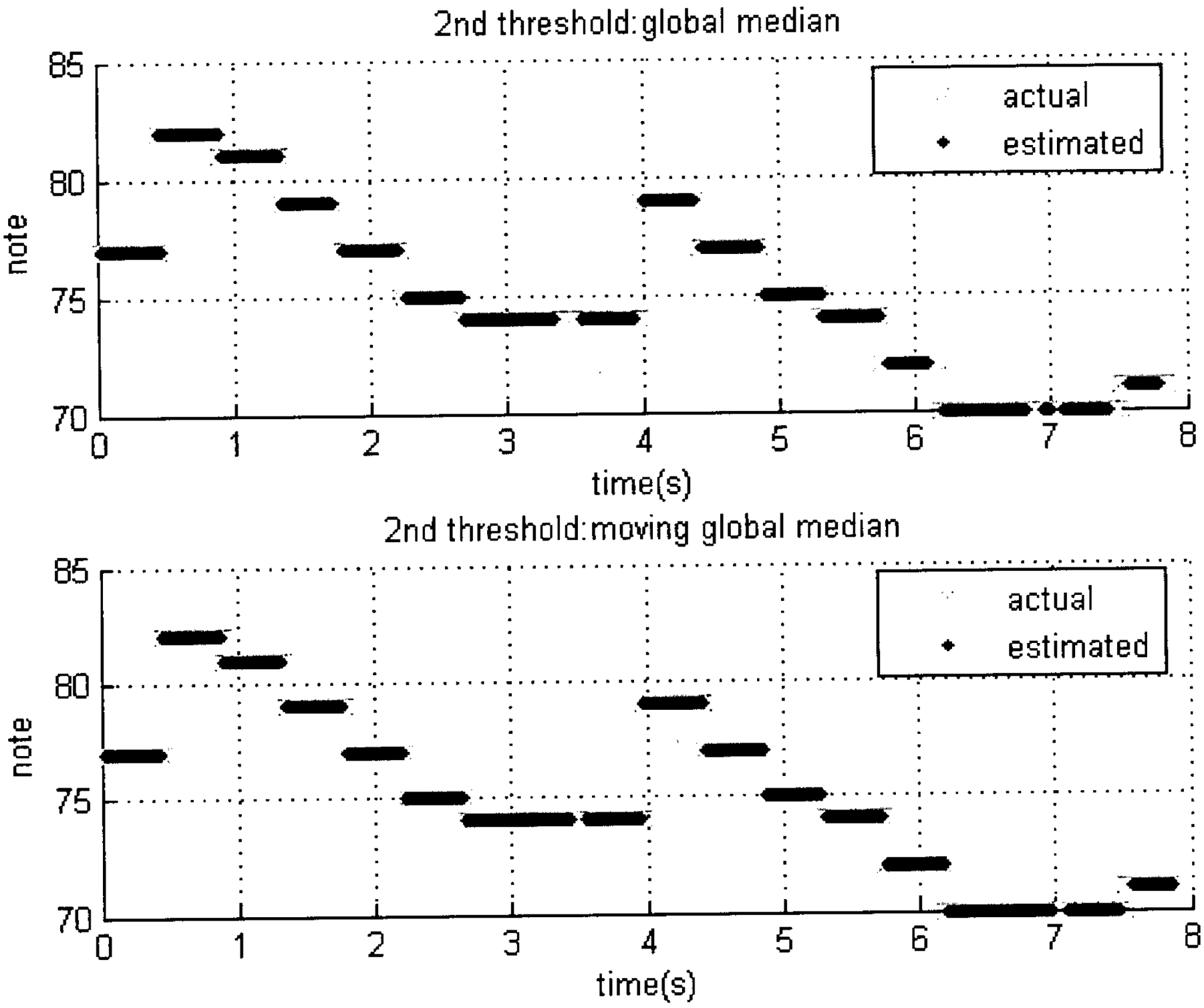

Figure 3.11: Note extraction result with different 2nd threshold in peak picking

harmonic	semitone number	ratio
0	n	$2^{0/12} = 1$
2	n+12	$2^{12/12} = 2$
3	n+19	$2^{19/12} \simeq 3.00$
4	n+24	$2^{24/12} = 4$
5	n+28	$2^{28/12} \simeq 5.04$
6	n+31	$2^{31/12} \simeq 5.99$
7	n+34	$2^{34/12} \simeq 7.13$
8	n+36	$2^{36/12} = 8$
9	n+38	$2^{38/12} \simeq 8.98$
10	n+40	$2^{40/12} \simeq 10.08$

Table 3.2: The relation of harmonics and notes position

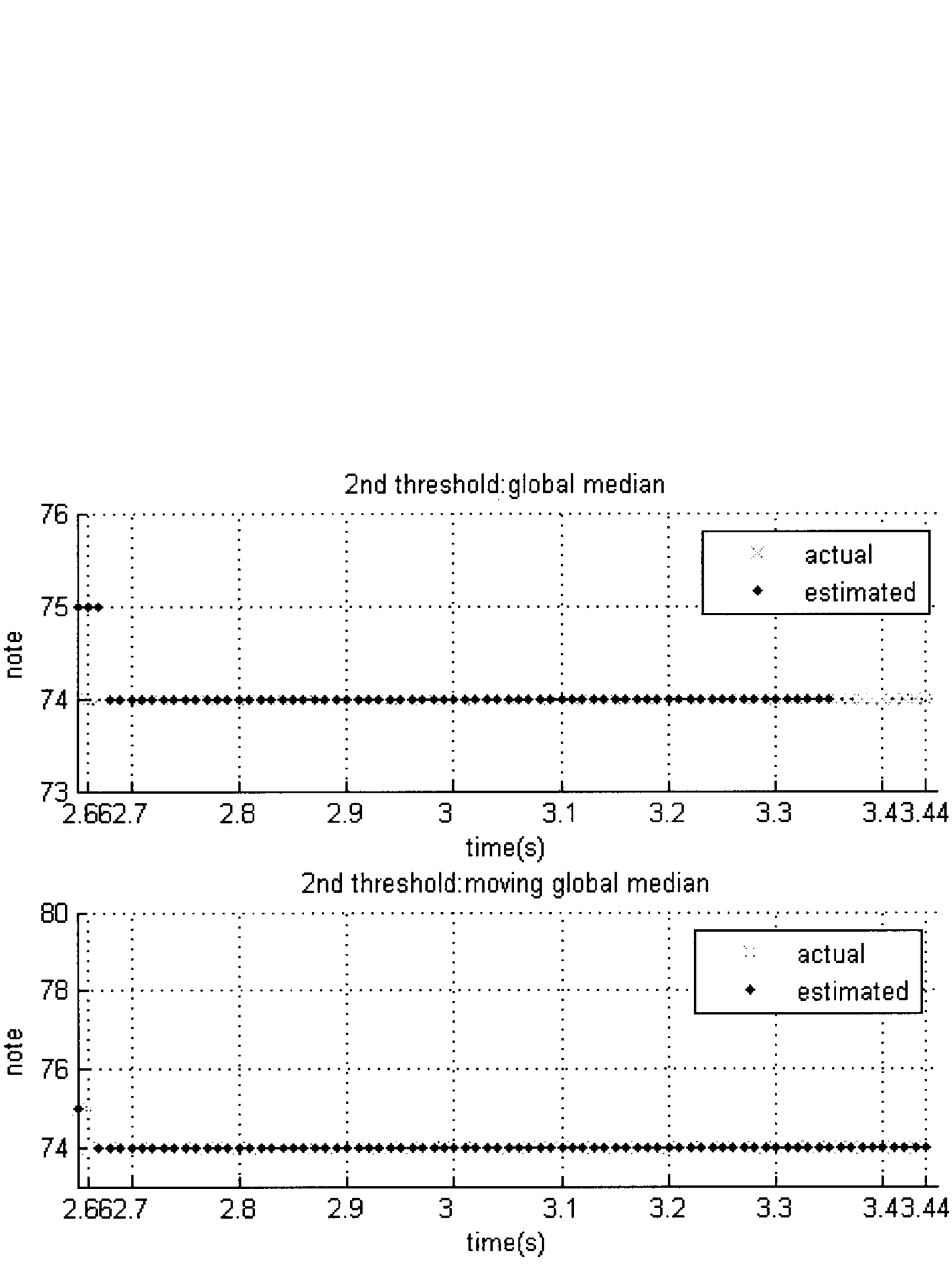

Figure 3.12: Note extraction result with different 2nd threshold in peak picking: amplified version

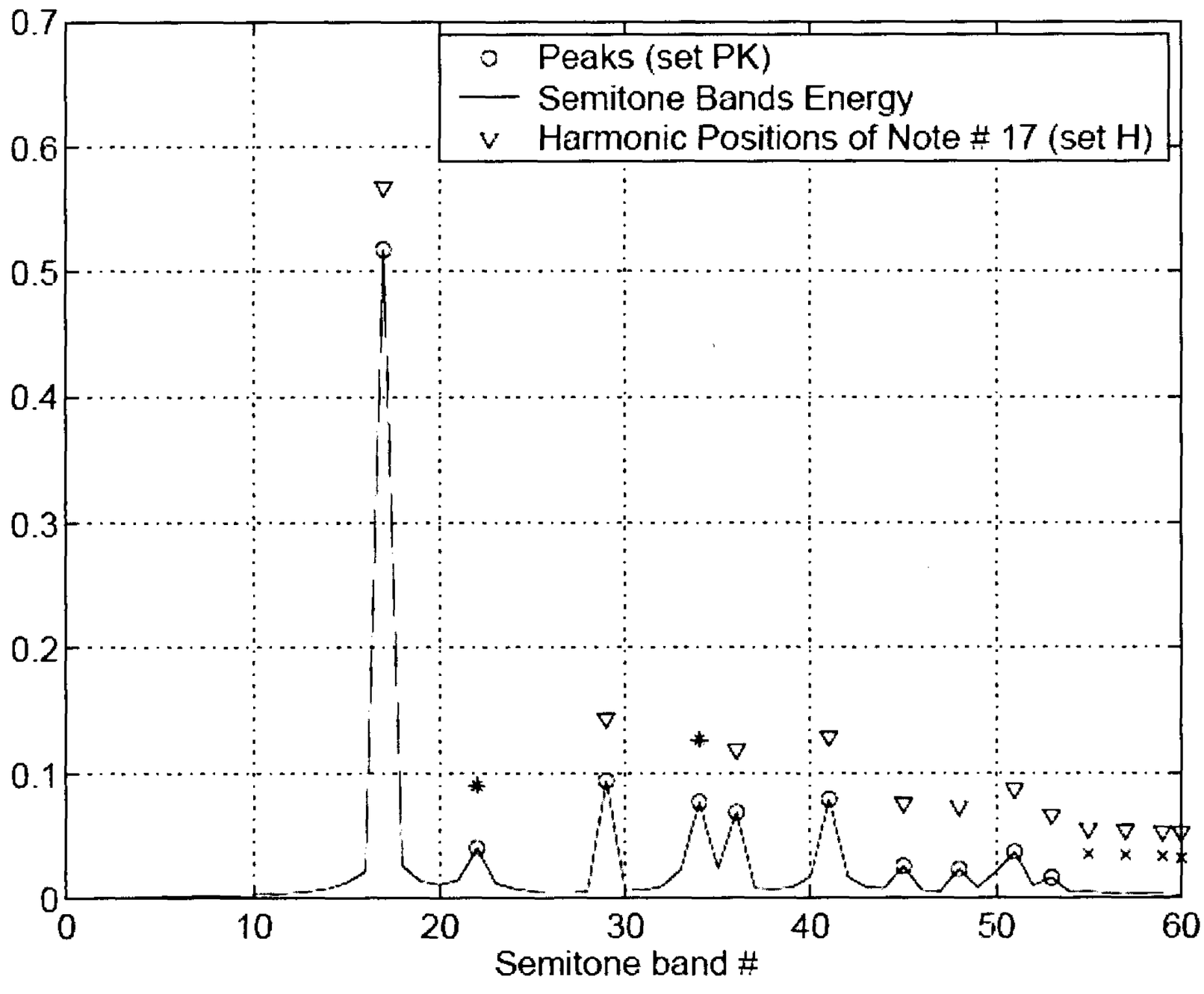

Figure 3.13: An example of Matching Probability Function

3.2.2.4 Matching Probability Function

As explained in Section 1.1 and in Table 3.2, the first 10 major harmonics of any ideal note either match another musical note, or are so close to a real one that they can be considered one. And the magnitude semitone-band spectrogram gives us the energy change on each narrow band of the 72 music notes. We will assume that, for a certain note, the energy of the narrow band covering its harmonics is the energy of the harmonics itself.

In Y. Ma's paper[2], she proposed a method call "Matching Probability Function" to decide if certain note/notes is present in the music at a certain time, from the energy distribution in the magnitude semitone-band spectrogram. The function is the multiple of two independent probability functions. The first one is named "Matching Peak Function". Consider the example illustrated in Figure 3.13, the circles 'o' mark all the peaks, which are the semitone bands with energy high enough to be put into consideration. They are defined as set PK. The triangulares '$\bigtriangledown$' mark all the semitone bands corresponding to the fundamental and harmonics(only the first 10 harmonics are considered) of a note(number 17, the frequency is $493.88Hz$). They are defined as set H. The Matching Peak Function is the weighted energy sum of semitone bands belong to both PK and H over the weighted energy sum of semitone bands belong to PK. Any semitone bands cannot be explained by given note/notes will decrease the value

$$P_{peak} = \frac{\sum_{s \in PK \cap H} W_p(s) \cdot M(s)}{\sum_{s \in PK} W_p(s) \cdot M(s)} \tag{3.5}$$

where s is a semitone band, $M(s)$ is the magnitude on this semitone band, which could be regarded as the energy over this band, and $W_p(s)$ is the weight of each peak in proportion to the reciprocal of its position in PK

$$W_p(s_n) = \frac{1}{n} \tag{3.6}$$

with s_n is the semitone number of the n^{th} peak.

The second independent probability function is named Matching Harmonic Function, which is the weighted sum of semitone bands belong to both PK and H over the weighted sum of semitone bands belonging to H

$$P_{harm} = \frac{\sum_{s \in PK \cap H} W_h(s)}{\sum_{s \in H}} \tag{3.7}$$

where s is a semitone band, and $W_h(s)$ is the weight of each harmonic in proportion to the reciprocal of its position in the harmonic structure

$$W_h(s_m) = \frac{1}{m} \tag{3.8}$$

with s_m is the semitone number of the m^{th} harmonic.

The selection of the two relative weight functions $W_p(s)$ and $W_h(s)$ is based on the knowledge that the energy of violin signals are mainly on the first several harmonics since Y. Ma is only dealing with violin music signal. She states that in order to work

on other musical instruments, these weight function need to be modified by searching the library of music produced by the instruments. Also, the harmonics set H only include the first 10 harmonics in both probability functions. It is chosen to be 10, partly because the closeness of a certain musical note harmonics to other integer note is not satisfied after the $11th$ harmonic($2^{41/12} \simeq 10.6787$, $2^{42/12} \simeq 11.3137$), partly because the energy on higher harmonics are normally much lower than the first 10 harmonics for violin music that it could be omitted as zero

The matching probability function is defined as

$$Prob = P_{peak} \cdot P_{harm} \tag{3.9}$$

Since both the matching peak function and the matching harmonic function are guaranteed to range from 0 to 1(when $H \in PK$, the value is 1, otherwise the value is less than 1), their multiple is also guaranteed to the range of $[0, 1]$.

The two matching probability functions are modified in order to identify musical signals with multiple concurrent notes. The matching peak function is modified as:

$$P_{peak}(t, C(H(n_m))) = \frac{\sum_{s \in PK(t) \cap (H(n_1) \cup \cdots \cup H(n_m))} W_p(t,s) \cdot M(t,s)}{\sum_{s \in PK(t)} W_p(t,s) \cdot M(t,s)} \tag{3.10}$$

where t is the fraction number, $PK(t)$ is the set of peaks for fraction t, $W_p(t,s)$ is the weight function following Equation3.6, $H(n_i)$ is the harmonics structure of note n_i – the set of all semitone bands corresponding to the major harmonics of note n_i,

$C(H(n_m))$ is the combination of possible notes harmonic structures at fraction t with their number m restricted to no more than 4 in practical experiment.

And the matching harmonic function is modified as:

$$P_{harm}(t, C(H(n_m))) = \sqrt[m]{\prod_{k=1}^{m} \frac{\sum_{s \in PK(t) \cap H(n_k)} W_h(n_k, s)}{\sum_{s \in H(n_k)} W_h(n_k, s)}} \tag{3.11}$$

where $W_h(n_k, s)$ is the weight function following Equation3.8 on note n_k at fraction t.

The matching probability function is then defined as the product of the modified matching peak function and the modified matching harmonic function

$$Prob(t, C(H(n_m))) = P_{peak}(t, C(H(n_m))) \cdot P_{harm}(t, C(H(n_m))) \tag{3.12}$$

These modified matching probability functions defined by Equation 3.10,3.11, and 3.12 are used at the stages of identifying continuous un-identified fractions. For each fraction in the continuous un-identified fraction segments, all possible present notes combination will be rated by these functions. The one with the highest score will be considered as the correct one. The system will estimate that, the notes combination with the highest score is present at that time. Our current experiments restrict the number of notes m in the combination to be no more than 4. Musical signal with more than 4 simultaneous notes is a possible topic for future work.

Figure 3.14 illustrates an example of two concurrent notes at certain fraction. Figure

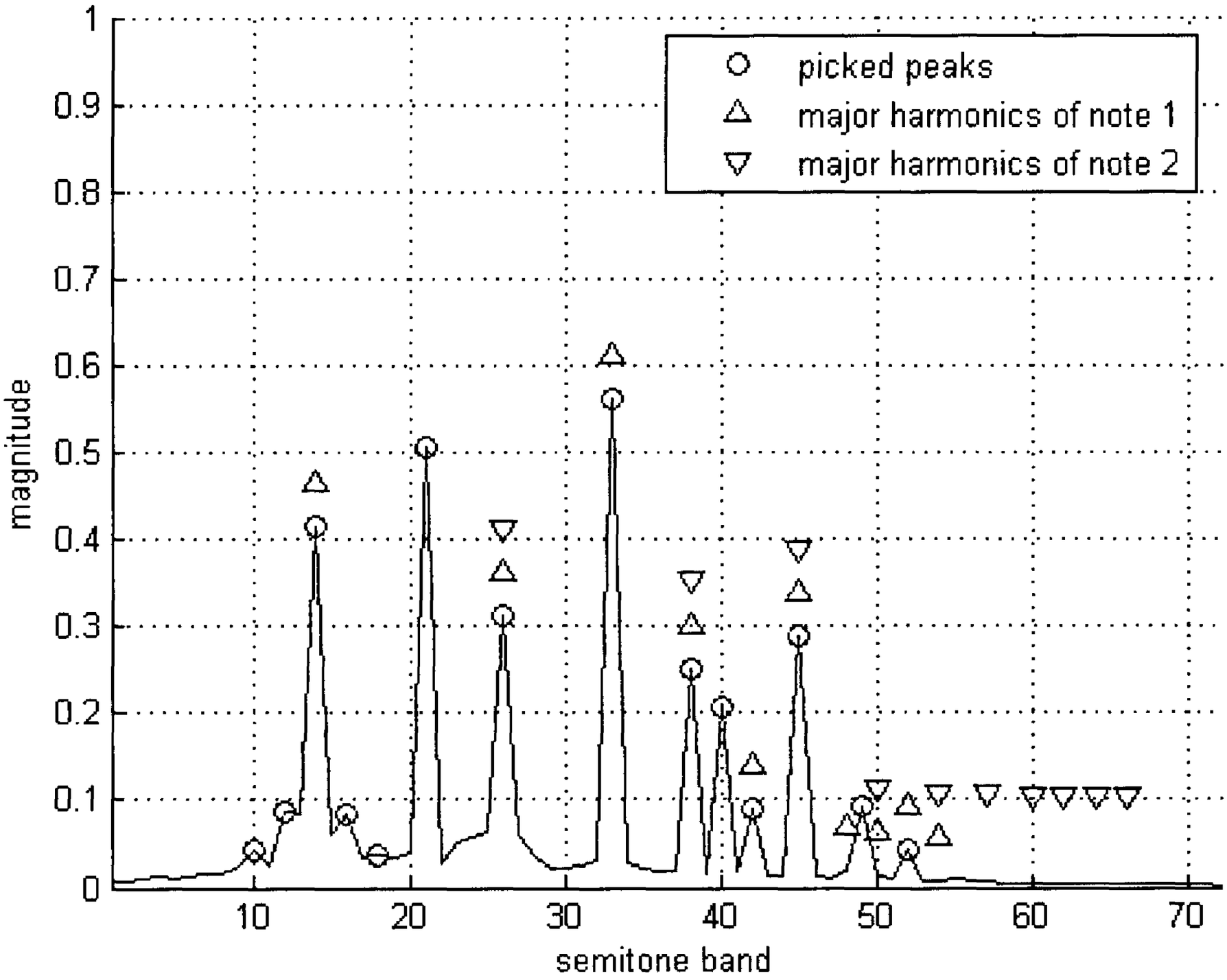

Figure 3.14: An example of two concurrent notes

3.15 illustrates an example of three concurrent notes at certain fraction.

3.2.2.5 Post-processing

A post-processing method is applied on the result to remove possible octave errors in the transient stage of some notes. Another task of post-processing is to deal with continuous un-identified fractions with a length less than 5. Similar to the post-processing method in [35], we assume silent segment should have a duration no less than $50ms$ and note/notes segment should have a duration longer than $50ms$, that

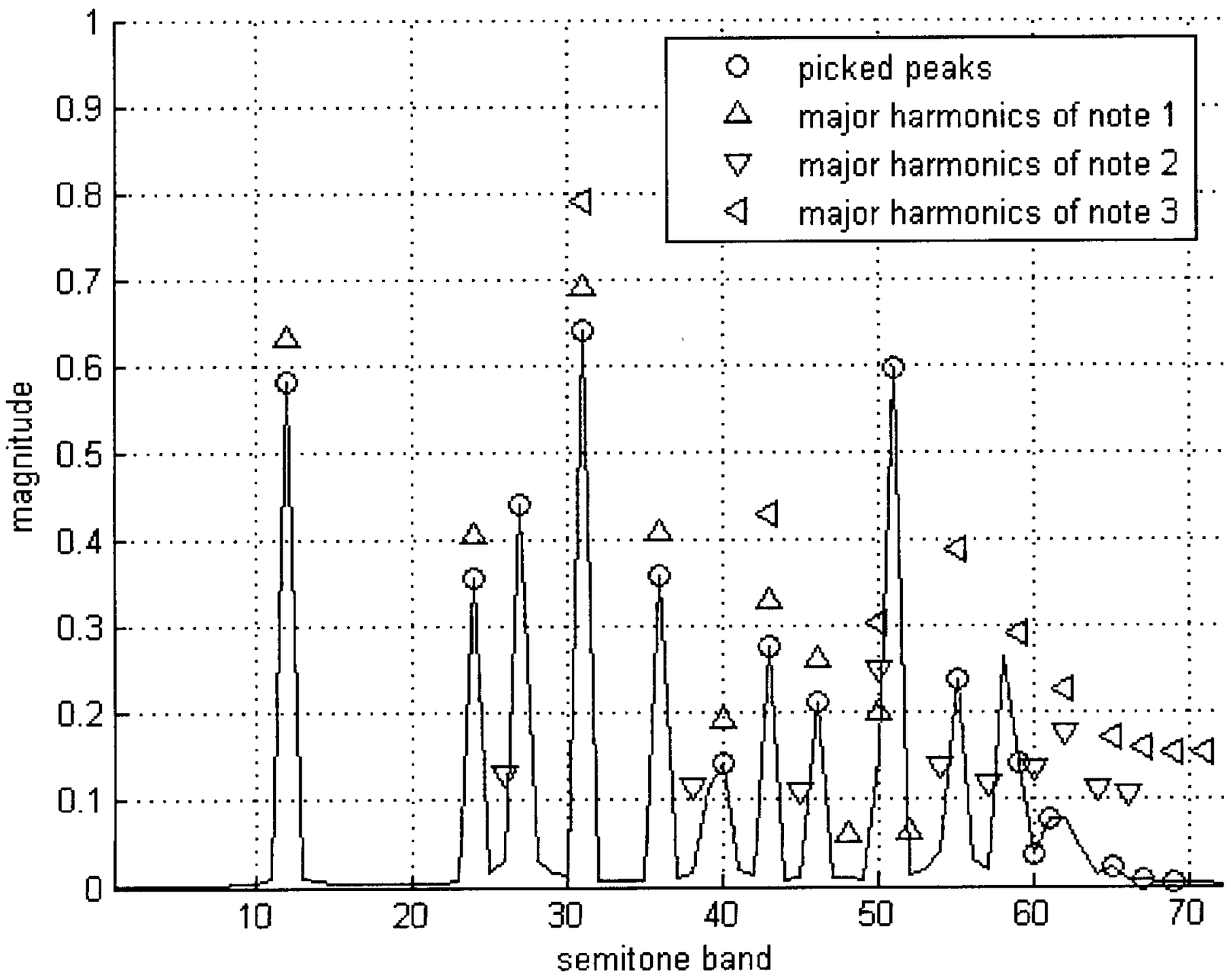

Figure 3.15: An example of three concurrent notes

means silent segment should have at least 5 fractions, each note should have more than 5 fractions. Three cases need to put into consideration. The first one is, both sides of un-identified fractions with a length less than 5 are already identified as silent fractions. In this case, these fraction will be identified as silent fractions based on our assumption. The second case is that the un-identified fractions have one side identified having note/notes, and the other side identified as silent fractions. In this case, these un-identified fractions will be identified having note/notes which is the same with the adjacent identified fractions. The third case is that the un-identified fractions have both sides identified having note/notes. In this case, these un-identified fractions will be identified having note/notes in both sides. This setting tries to identify as many as fractions as note/notes, but it will introduce some error and decrease the precision by increase the number of *falsePositive*. However, since our current experiment shows that the *recall* rate is relative low, and the above setting will increase the *recall* rate, this setting is more proper. Afterall, for this case, there is a tradeoff to balance the *recall* rate and the *precision*.

3.2.3 Evaluation Method

To evaluate the performance of a system and to compare the performance between different systems, an effective evaluation method is needed to be established. Several evaluation methods had been proposed in previous researches. Dixon [36] proposed a frame-level version named Overall Accuracy to measure the general accuracy of the

system. Poliner[37], on the other hand, proposed a group of error measure functions to measure the errors. He discriminates the errors into three categories: substitution errors(mislabeling note/notes), "miss" errors(when note/notes is/are present in the fractions but missed in the estimated transcript result), and "false alarm" errors(when note/notes is/are reported without any underlying source). He states this three-way decomposition avoids the problem of 'double-counting' errors.

In previous Y. Ma's work[2], she did not propose a proper method to evaluate the result. She compared the result of estimated score to the real score by plotting them together with the same axes setting. By observing the amount of overlap, people can generally tell if the system is doing good.

In this thesis, we proposed an evaluation method which is widely used in the realm of machine learning. We believe this method is effective and comprehensive, evaluating not only the general accuracy of the system, but also its error rate.

Some notions are used in the evaluation method. TP("true positive") is the number of correctly identified notes, FN("false negatives") is the number of notes which are present in the music but failed to be identified, and FP("false positive") is the number of notes which are identified by the system but is actually not present in the music.

The first evaluation function is Recall rate, defined by

$$Recall = \frac{\sum_t TP(t)}{\sum_t (TP(t) + FN(t))} \tag{3.13}$$

The recall rate is the ratio of the number of correctly identified notes to the number of total reference notes(the number of notes present in the music) at fraction t, reflecting the ability of the system to pick up notes. The higher the value, the more capable the system is to pick up notes from the mixture.

The second one is Precision, which is defined as

$$Precision = \frac{\sum_t TP(t)}{\sum_t (TP(t) + FP(t))} \tag{3.14}$$

The precision is the ratio of the number of correctly identified notes to the number of total estimated notes(the number of notes estimated by the system) at fraction t, reflecting the system's accuracy in identification [35]. The higher the value, the more accurate the system will be.

The third one is total error rate, defined as

$$e_{tot} = \frac{\sum_t max(TP(t) + FN(t), TP(t) + FP(t)) - TP(t)}{TP(t) + FN(t)} \tag{3.15}$$

The total error rate counts all possible errors in the system, and compares them to the number of notes actually present in the fraction at that time.

music	instrument	length(s)	concurrent notes
A-Major	violin	41.6	1
Arpeggion	clarient	23	1
Child and Star	flute	11.6	1
Angels	two violins	30	2
God Rest	two clarinet	20.1	2
Mozart K.487,No.1	two oboe	47.2	2
Sonata F Major,Op.1,No.1	two violin and cello	55.8	3
God Rest Ye Merry	choir(soprano, alto, tenor and bass)	33.5	4
Childful	clarinets, oboe, flute and violin	82	4
string quartet	2 violins, viola and cello	39.2	4

Table 3.3: Data set of the midi music in the experiment

3.2.4 Experiment and Results

The experiment is carried out under MATLAB version 7.0.4. The data set we choose are all midi music samples. The reason we choose synthesized music over real music is that, we know exactly the reference score of the synthesized music. Actually we also implement our algorithm for real music, but since we could not find the ground-truth pitch score for music played by the instrument we are interested at. We cannot determine the performance of our system.

We test our algorithm on 9 pieces of midi music with a total length of 384 seconds as illustrated in Table 3.3, including 3 pieces of solo music, 3 pieces of music played by two of the same instruments or two different instruments, 1 piece of trio music, and 3 pieces of quartet music. Those music are played by a group of instrument including violin, flute, clarinet, and oboe, with their frequency spectrum within the spectrum coverage of the system.

The identification results are shown in two ways. The first is to plot the estimated

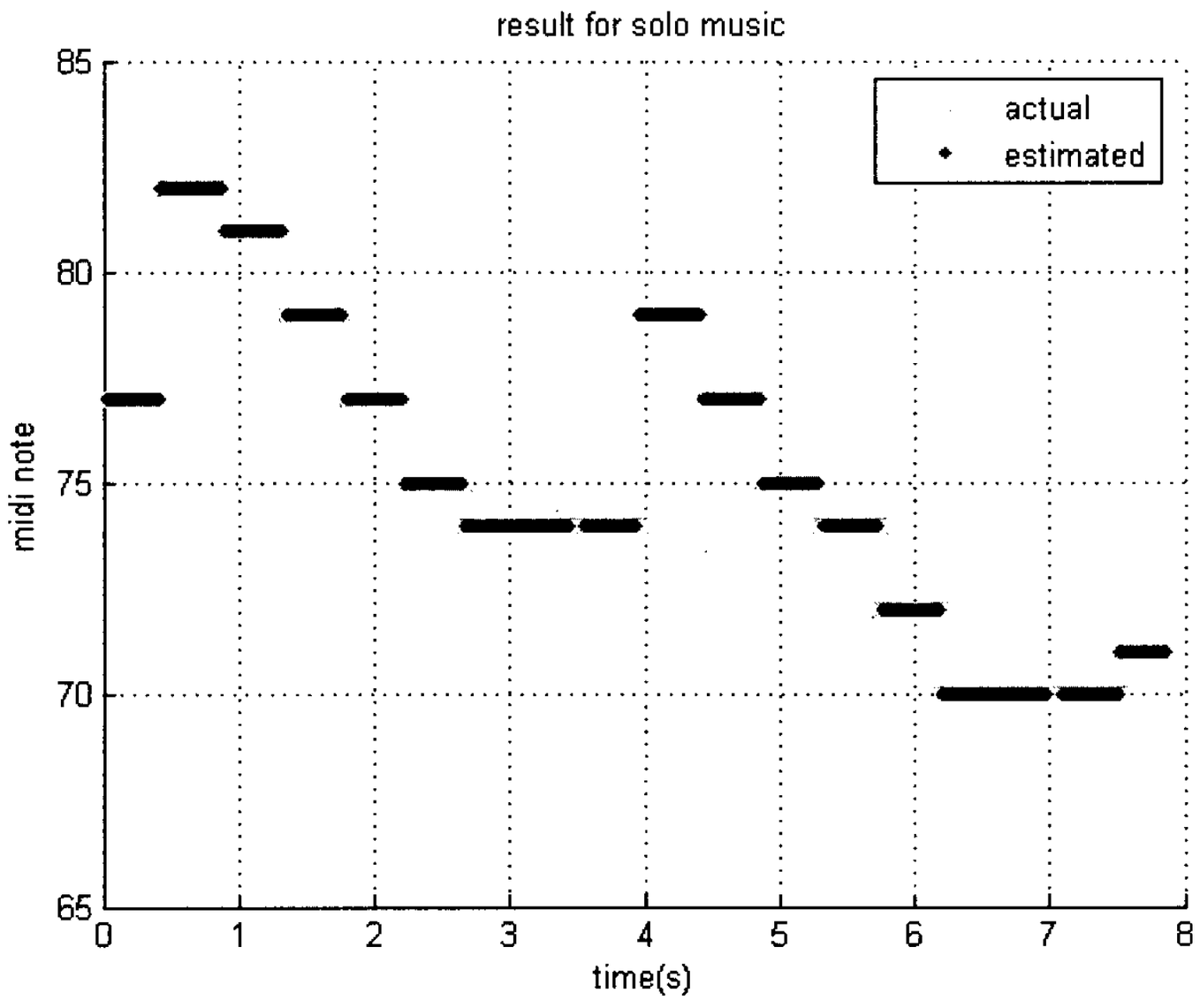

Figure 3.16: Estimated result for solo music

notes and the reference notes together to get a general impression of the system performance. The second is to use the evaluation method proposed in Section 3.2.3 to get more accurate evaluation of the system performance.

Figure 3.16, 3.17, 3.18, and 3.19 illustrate the results of plotting the estimated results and the actual notes together for solo, duo(with at most two concurrent notes), trio(with at most three concurrent notes), and quartet(with at most four concurrent notes) music. We can see from the plot that, the main error of the system is octave error(two concurrent notes n and $n + 12$ identified by the system to be one note n). The main reason for this is that, the harmonic structures for two notes with one octave distance are highly overlapped so that when the fractions are rated by the matching probability functions, the probability of one note is higher than that of two

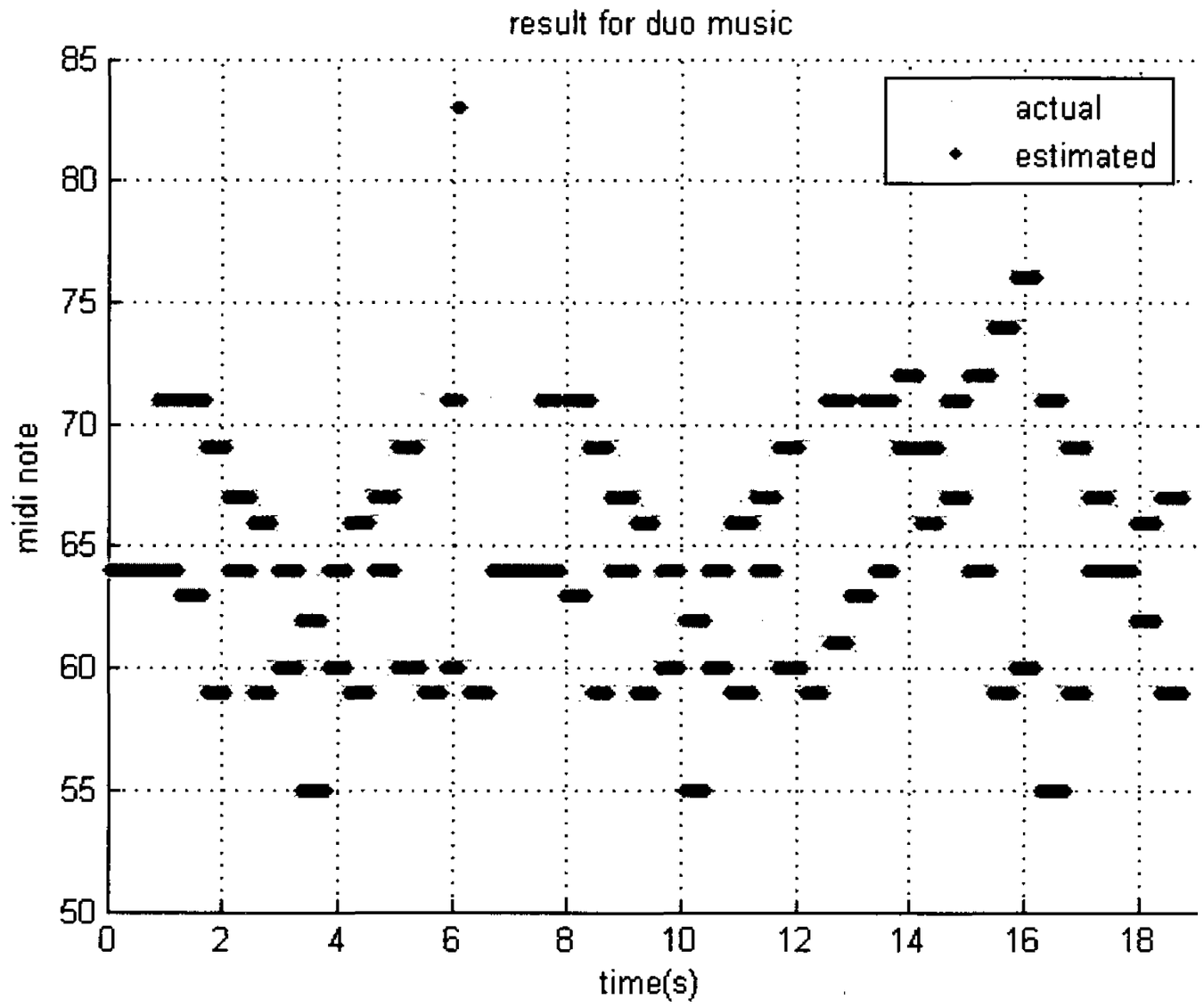

Figure 3.17: Estimated result for duo music

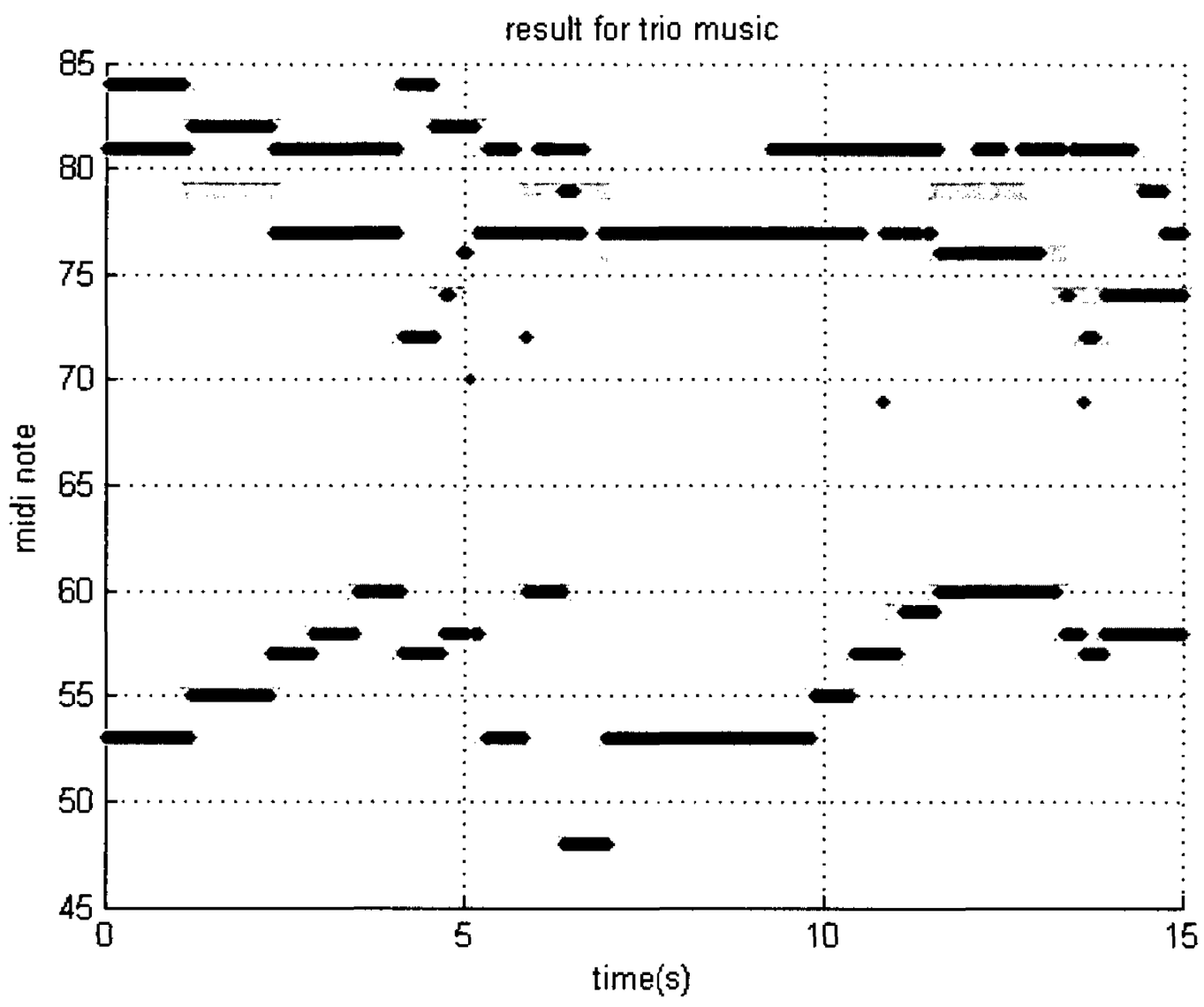

Figure 3.18: Estimated result for trio music

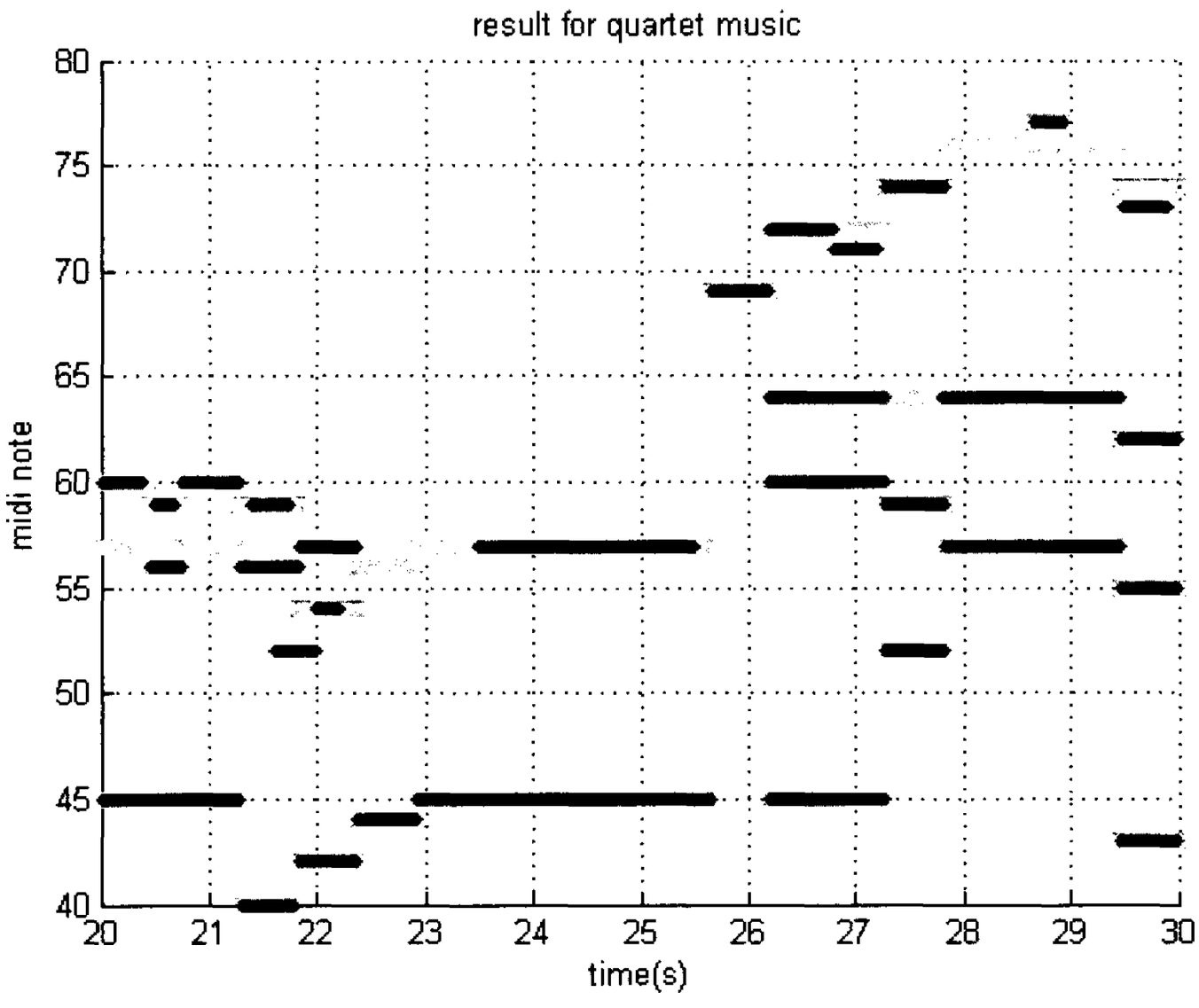

Figure 3.19: Estimated result for quartet music

notes. This situation happens more frequently when one note in the music has several lower harmonics missing in the harmonics structure.

Table 3.4 shows the final results of the whole data set using the evaluation methods proposed in this paper. It shows that, as the number of concurrent notes increases, the recall rate drops dramatically, from 96.1% for solo music to 68.4% for quartet music. This is understandable because for our system, as the number of concurrent notes increase, the peak-semitone band representation is increasingly complex. Also since our system can not pick two adjacent peaks, some real harmonics of the combination of notes in the music can not be picked, thus rendering the matching probability score for that combination decreases so that the correct combination does not have the highest score. The table also shows that, although the recall rate drops fast when

type	solo	duo	trio	quartet
Recall	96.1%	86.6%	73.2%	68.4%
Precision	98.2%	95.8%	94.8%	92.2%
Error total	6.9%	13.1%	22.3%	30.7%

Table 3.4: Result using proposed evaluation method

the number of concurrent notes increase, the precision remains stable(from 98.2% for solo music to 92.2%) and stays above 90% for all cases. This shows the ability of the system to exclude possible *FPs*.

Chapter 4

Modification of Frequency Tracking based on Adaptive Internal Model Control Theory

4.1 Introduction

Active suppression of noise is an interesting and challenging problem, and many different approaches have beed used to cancel disturbances, such as adaptive control techniques[38], Kalman filter based approaches[39], least-mean-square(LMS) gradient approximation approach[40], and recursive least square(RLS) based approach[41]. An overview of these approaches can be found in [42]. Another approach, called Internal Model Principle[6], was proposed by Francis and Wonham in 1976. It states that, perfect disturbance rejection is achieved if a replicate model of the disturbance is contained in the stable closed-loop system(the detail is discussed in 2.1). However, in most cases the disturbance model is not a known prior, and the disturbance properties are not constant over time. To overcome this limitation, an adaptive version

of the internal model controller(IMC) was proposed by L.J. Brown and Q. Zhang in [4, 15] for cancelation of periodic signals with uncertain frequency. They developed a function to map the time-varying states of the internal model to the time-invariant frequency of the disturbance signals. An integral controller is implemented to update the parameter of the internal model such that the controller can converge to the actual frequencies present in the signal. The convergence and stability of this adaptive control system is verified by singular perturbation theory and averaging theory. Jin [43] extended the algorithm for canceling a disturbance composed of a sum exponentially damped sinusoidal signals. Zhao [44] implemented the algorithm to extract the pitch from synthesized monophonic signals, realizing excellent temporal resolution to discriminate rapid musical passages. Sun [8] proposed a time-frequency analysis theory: Instantaneous Fourier Decomposition based on this theory, and applied it to analyze experimental weld signals having at most 3 harmonics. Yan [2] further developed an iterative system based on IFD for violin music decomposition, capable to deal with at most 2 concurrent pitches.

One problem of the adaptive internal model theory is that, its exponential convergence means that it cannot estimate the signals properties during the initial transient. Further, due to robustness and numerical stability issues, the greater the number of internal models, the slower the algorithm's convergence. For several applications such as the welding examples conducted by Sun, this initial information is crucial. In the welding problem, the goal is to measure the energy supplied to the weld by an AC

power supply controlled using silicon control rectifier (SCR) technology. The problem is the voltage cannot be directly measured as a result of the large magnetic fields produced by the $10-20$ KA AC currents used for welding. When measurement wires are connected to the weld electrodes, the quantity measured is a sum of the weld voltage and a term proportional to the derivative of the current. Further, SCR controlled power signals are not pure sinusoids but contain high levels of the odd harmonics. the real power supplied to the weld will be the part of the voltage signal in phase with the current. Thus by decomposing the voltage and current into phaser representations as the IFD does, it is possible to calculate the energy supplied to a weld after the initial convergence of the algorithm. Unfortunately, the maximum variance in the process occurs during this convergence period. Further since welds are typically only $10-15$ cycles in duration, it is imperative, that this initial energy be calculated as quickly as possible. Another example where the initial information is crucial is in identifying and modeling the attack characteristics of a musical instrument.

In this thesis, we proposed a method of running the adaptive internal model theory based algorithm twice, with the first time as normal, and the second time backwards-in-time. In the second time running, the input signal is reversed to backwards-in-time. With proper initialized value of the system's parameters, perfect tracking with no transition in the signal will be achieved. Also, in order to analyze signals with multiple harmonics, multiple Internal Models are incorporated in parallel in the feedback loop, with each IM tracks one harmonic contained in the signal. The algorithm

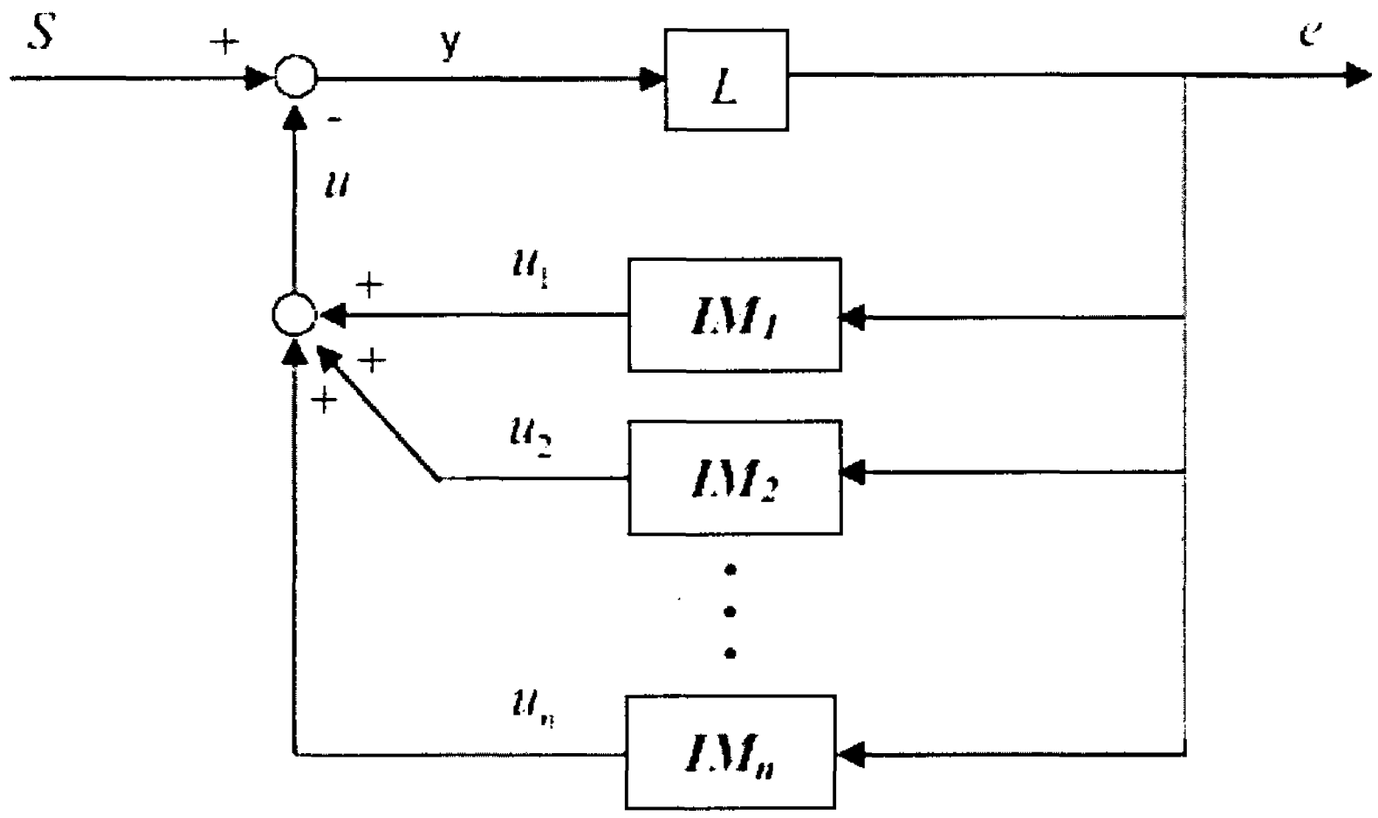

Figure 4.1: Block diagram of the frequency tracking system

diagram is illustrated in Figure 4.1. With this approach, the system is able to analyze monophonic signals with multiple harmonics, or polyphonic signals with multiple harmonics. Our ultimate goal is to implement the algorithm to analyze real music. As illustrated in Figure 4.2, real music normally has transition periods[45], the energy in which rises from zero to a sustained value at the beginning of a certain pitch and falls to zero at the end of the pitch. The transition period, especially the rising part of the signal(the attack period in figure4.2), is very difficult to analyze because the period is usually very short but with a considerably large energy change. Since our algorithm is able to track the initial period of a signal, thus we can use it for transient modeling of the music signal.

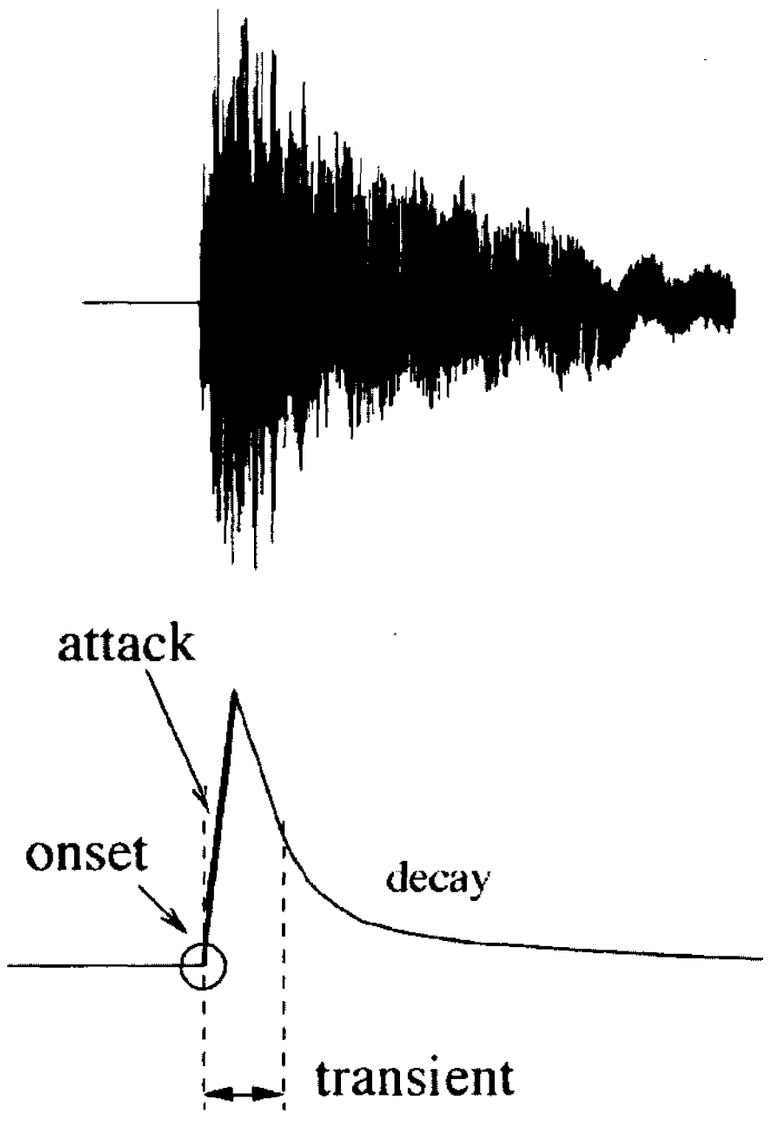

Figure 4.2: The transient period in an ideal case of single pitch

4.2 Full-length Frequency Tracking based on Adaptive Internal Model Control Theory

As discussed in 2.2.4, Instantaneous Fourier Decomposition transform the input signal into a magnitude semitone channel spectrogram. The desired feedback control system behaves like a band-pass filter with multiple notches. The advantage of this method is that, the number of the notches is configurable(as much as 72 channel). However, since IFD is inadaptive and treats the energy in the semitone channel as the energy of the central frequency of that channel, we can not tell the exact frequency of the signal at that moment. Also, the speed of convergence in IFD is not controllable, and sometimes the transient time is longer than the whole duration of the signal,

rendering the system unable to identify the frequency.

In this thesis, we proposed a method of designing the pole locations of the system directly. By placing the poles of the closed loop system near the origin in the $z-$ *domain*, we can control the convergence speed, making it relatively faster than IFD.

4.2.1 Design of the Plant L

The plant L in the diagram is chosen as a simple low-pass filter with its transfer function in the $z-$ *domain* as $L(z) = \frac{z-1}{z+b}$. One variant b in the denominator is decided by matching the coefficients of the actual system to the desired closed-loop system that will be explained in the next section.

4.2.2 Design of Pole Location

The method in this thesis is to place the closed-loop poles in an arch within the unit circle in $z-$ *domain*. Given the plant transfer function $L(z) = \frac{z-1}{z+b}$, the closed-loop poles should be in an arch in the right semicircle of the unit circle, with one pole in the real axis, and others are pairs symmetric about the real axis. Figure 4.3 shows the poles location of a $7-$ *order* system with 3 paralleled IMs. The radius of all the desired poles are the same value of $r = 0.8341$. In fact, since the pole location in our method is totally configurable, the desired poles can be placed anywhere within the unit circle. However, in order to increase the convergence speed and to provide a sufficiently large frequency variation range for the adaptation, the 7 poles are placed

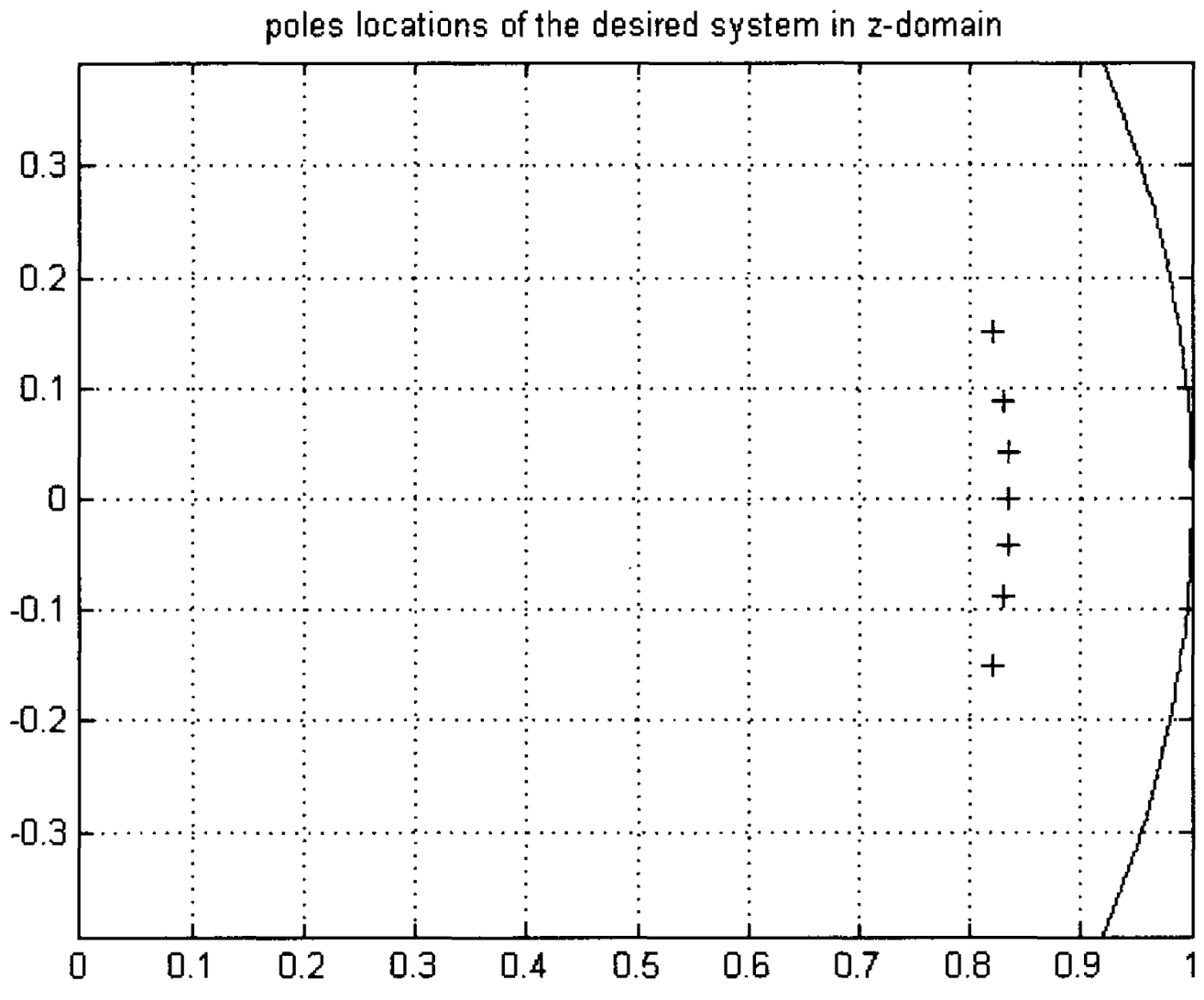

Figure 4.3: Desired poles location of a 7-order system

in an arch with the root locus(pole location versus the frequency ω) illustrated in Figure 4.4. Randomly placed poles will drive the lines in the root locus easily going out of the unit circle, rendering the system unstable.

The adaptive internal model principle uses an internal controller to force the estimated frequency ω to the actual frequency of the signal. As ω changes in the adaptation process, the poles location of the closed-loop system varies as well. To guarantee the system is always stable, we need a sufficiently large range of ω so that the closed loop poles of the system remain in the unit circle in $z - domain$. The root locus in our example 4.4 has a frequency range $[220Hz, 660Hz]$, which is sufficiently large for current adaptation speed.

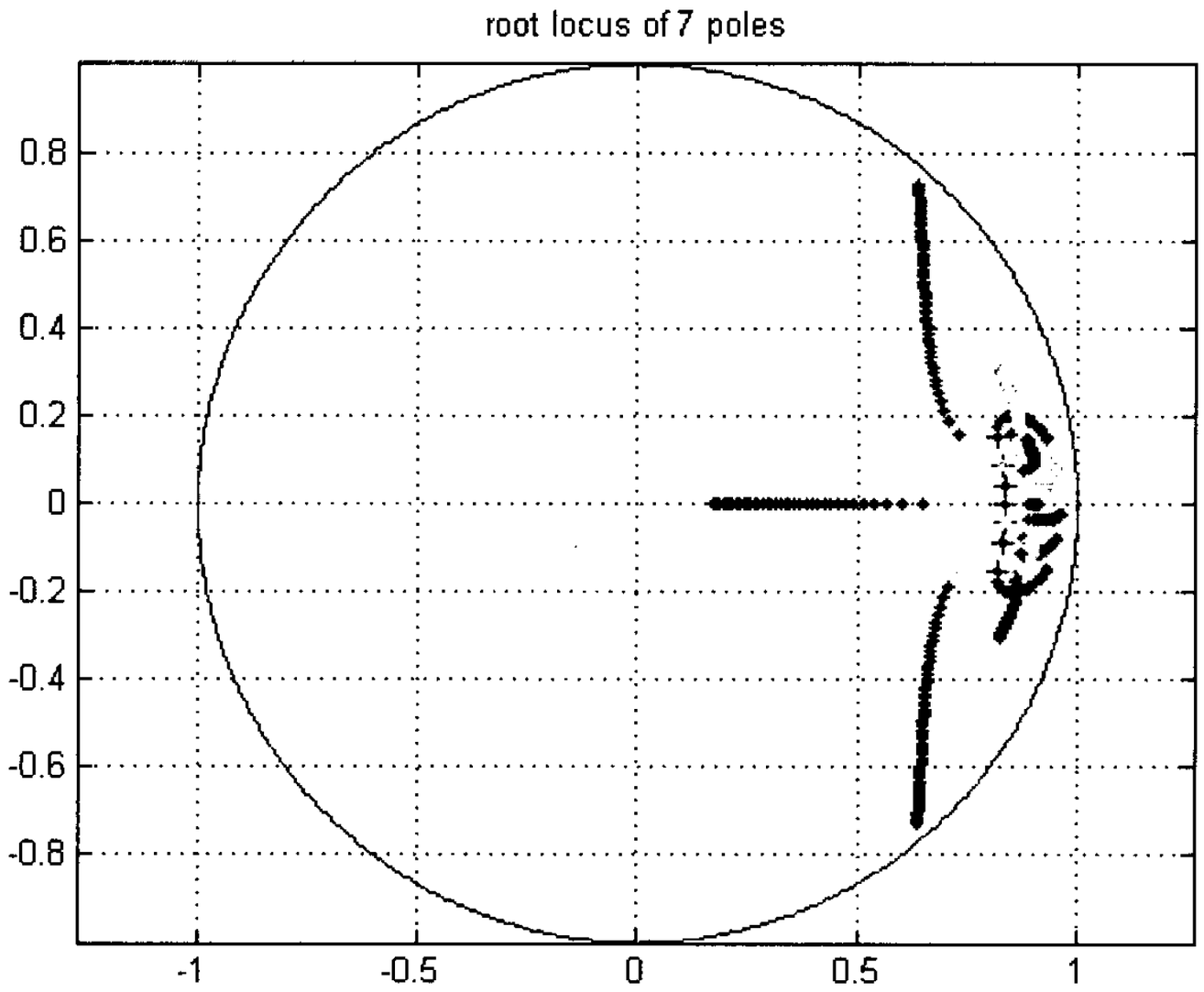

Figure 4.4: root locus of 7 poles with frequency range $\omega \in$[220Hz,660Hz]

The continuous-time state space form of the internal model in the system diagram 2.2 is as follows

$$\begin{bmatrix} \dot{x}_{1i} \\ \\ \dot{x}_{2i} \end{bmatrix} = \begin{bmatrix} 0 & w_i \\ \\ -w_i & 0 \end{bmatrix} \begin{bmatrix} x_{1i} \\ \\ x_{2i} \end{bmatrix} + \begin{bmatrix} 0 \\ \\ 1 \end{bmatrix} e \tag{4.1}$$

$$u_i = \begin{bmatrix} k_{1i} & k_{2i} \end{bmatrix} \begin{bmatrix} x_{1i} \\ \\ x_{2i} \end{bmatrix} \tag{4.2}$$

with the transfer function from e to u_i

$$T_{e \to u_i} = \frac{k_{2i}s + k_{1i}w}{s^2 + w^2} \tag{4.3}$$

By mapping the state space equations from continuous-time to discrete-time, the discrete-time state space form of the internal model is as follows

$$\begin{bmatrix} x_{1i}(T+1) \\ \\ x_{2i}(T+1) \end{bmatrix} = \begin{bmatrix} \cos(w_i) & \sin(w_i) \\ \\ -\sin(w_i) & \cos(w_i) \end{bmatrix} \begin{bmatrix} x_{1i}(T) \\ \\ x_{2i}(T) \end{bmatrix} + \begin{bmatrix} 0 \\ \\ 1 \end{bmatrix} e_i \tag{4.4}$$

$$u_i(T) = \begin{bmatrix} k_{1i} & k_{2i} \end{bmatrix} \begin{bmatrix} x_{1i}(T) \\ \\ x_{2i}(T) \end{bmatrix} \tag{4.5}$$

The feedback signal $u(t)$ is the summation of the output of each internal model

$$u(t) = \sum_{i=1}^{n} u_i(t) = \sum_{i=1}^{n} (k_{1i}x_{1i}(t) + k_{2i}x_{2i}(t)) \tag{4.6}$$

where n is the number of internal model(3 in our case). The transfer function from e to u_i is

$$T_{e \to u_i} = \frac{k_{2i}z + k_{1i}z\sin(w_i) - k_{2i}z\cos(w_i)}{z^2 - 2\cos(w_i)z + 1} \tag{4.7}$$

with the 7 desired poles placed in the z-domain, the denominator for the 7-order desired closed loop system is

$$DEN_{desired} = \prod_{i=1}^{n} (z - p_i(z)) \tag{4.8}$$

where n is the number of paralleled internal models. With the transfer function of the plant is $L(z) = \frac{z-1}{z+b}$, the denominator of the actual system is

b	k_{11}	k_{21}	k_{12}	k_{22}	k_{13}	k_{23}
-0.3179	0.7906	1.2497	1.5937	0.1710	0.2472	-1.0062

Table 4.1: The parameters in a 7-order closed-loop system

$$DEN_{actual} = (z-1)\cdot D(z) + (z+b)\cdot \prod_{i=1}^{n}(z^2 - 2cos(\omega_i)z + 1) \qquad (4.9)$$

with $D(z) = \sum_{i=1}^{n}((k_{2i}z + k_{1i}\sin(w_i) - k_{2i}\cos(w_i))\Pi_{l=1,\ldots,n}^{l\neq i}(z^2 - 2\cos(w_l)z + 1))$.

By matching the coefficients of the two equations 4.8 and 4.9, we can set the feedback gain $(k_{1i}, k_{2i})(i = 1, \cdots, n)$ for each IM and the unknown coefficient b of the plant $L(z)$. The value of these parameters in our experiment are given in Table 4.2.2.

4.2.3 Setting of Initial Values in Backwards-In-Time Running

The state variables in the system state-space equation include the state variables x_{1i} and $x_{2i}(i = 1, \cdots, n)$ and the estimated frequency ω. We obtain the final value of these state variables in the first time normal forwards-in-time running. Based on the methods discussed below, the correct initial values of these state variables are set, and the algorithm is run second time backwards-in-time, with the original reference signal reversed as the input. The following section will discuss how to calculate the correct initial values of the state variables.

Consider the plant L has a transfer function $L(s) = \frac{s+1}{s+a}$, following Equation 4.1, 4.3,

the transfer functions from the reference signal r to x_{1i} and x_{2i} are

$$\begin{aligned} TF_{s\to x_{1i}} &= \frac{T_{yx_1}}{1+L(s)T_{e\to u_i}(s)} \\ &= \frac{T_{ye}\cdot T_{ex_1}}{1+L(s)T_{e\to u_i}(s)} \\ &= \frac{\frac{s+1}{s+a}\frac{w}{s^2+w^2}}{1+\frac{s+1}{s+a}\frac{k_{2i}s+k_{1i}w}{s^2+w^2}} \\ &= \frac{w(s+1)}{(s+a)(s^2+w^2)+(s+1)(k_{2i}s+k_{1i}w)} \end{aligned} \tag{4.10}$$

$$\begin{aligned} TF_{s\to x_{2i}} &= \frac{T_{yx_2}}{1+L(s)T_{e\to u_i}(s)} \\ &= \frac{T_{ye}\cdot T_{ex_2}}{1+L(s)T_{e\to u_i}(s)} \\ &= \frac{\frac{s+1}{s+a}\frac{s}{s^2+w^2}}{1+\frac{s+1}{s+a}\frac{k_{2i}s+k_{1i}w}{s^2+w^2}} \\ &= \frac{s(s+1)}{(s+a)(s^2+w^2)+(s+1)(k_{2i}s+k_{1i}w)} \end{aligned} \tag{4.11}$$

Substituting $s = j\omega$ into the above equations, we have

$$\begin{aligned} TF_{s\to x_{1i}}(s=jw) &= \frac{1}{k_{1i}+jk_{2i}} \\ &= |K_i|\angle(tan^{-1}(\frac{-k_{2i}}{k_{1i}})) \end{aligned} \tag{4.12}$$

$$\begin{aligned} TF_{s\to x_{2i}}(s=jw) &= \frac{jw}{k_{1i}+jk_{2i}} \\ &= |wK_i|\angle(tan^{-1}(\frac{-k_{2i}}{k_{1i}})+\frac{\pi}{2}) \end{aligned} \tag{4.13}$$

where

$$|K_i| = \frac{1}{\sqrt{k_{1i}^2+k_{2i}^2}} \tag{4.14}$$

Assuming the reference signal is a pure sinusoid $s = |s|cos(w_c t + \varphi_s)$, with the magnitude $|s|$ is a constant. In the steady state, the the two state variables $x_1(t_s)$ and $x_2(t_s)$ are given as follow

$$x_{1s} = x_1(t_s) = |x|\cos(w_c t_s + \varphi_x) \tag{4.15}$$

where t_s is the time in steady state.

$$x_{2s} = x_2(t_s) = |x|\cos(w_c t_s + \varphi_x + \frac{\pi}{2}) \tag{4.16}$$

Following Equation 4.12, the extra phase from φ_s to φ_x is given by

$$\delta = \varphi_x - \varphi_s = tan^{-1}(\frac{-k_2}{k_1}) \tag{4.17}$$

In the second time running the algorithm backwards in time, the reference signal is reversed $s_b = |s|\cos(-w_c t + w_c t_s + \varphi_s)$, and the two state variables are given by

$$x_{1b}(t) = |x|\cos(-w_c t + w_c t_s + \varphi_s - \delta) = |x|\cos(w_c t + (w_c t_s + \varphi_x) - 2(\varphi_s - \varphi_x)) \quad (4.18)$$

$$x_{2b}(t) = |x|\cos(-w_c t + w_c t_s + \varphi_s - \delta - \frac{\pi}{2}) = |x|\cos(w_c t + (w_c t_s + \varphi_x) - 2(\varphi_s - \varphi_x) - \frac{\pi}{2}) \quad (4.19)$$

Substituting Equation 4.15, 4.16 and 4.17 into the above two equations, and setting $t = 0$, we have the correct initial values of the two state variables in the backwards-in-time running

$$x_{1b}(0) = |x|\cos(cos^{-1}(\frac{x_{1s}}{|x|}) - 2\delta) \quad (4.20)$$

$$x_{2b}(0) = |x|\cos(cos^{-1}(\frac{x_{1s}}{|x|}) - 2\delta - \frac{\pi}{2}) \quad (4.21)$$

4.2.4 Experiment and Results

The algorithm is implemented in MATLAB Simulink version 6.2. The reference signal is a synthesized signal with three harmonics, one is the fundamental frequency, the other two are the second and third harmonics:

$$s(t) = m_1\cos(2\pi\int w_1(t)dt + \varphi_1) + m_2\cos(2\pi\int w_2(t)dt + \varphi_2) + m_3\cos(2\pi\int w_3(t)dt + \varphi_3) + n \quad (4.22)$$

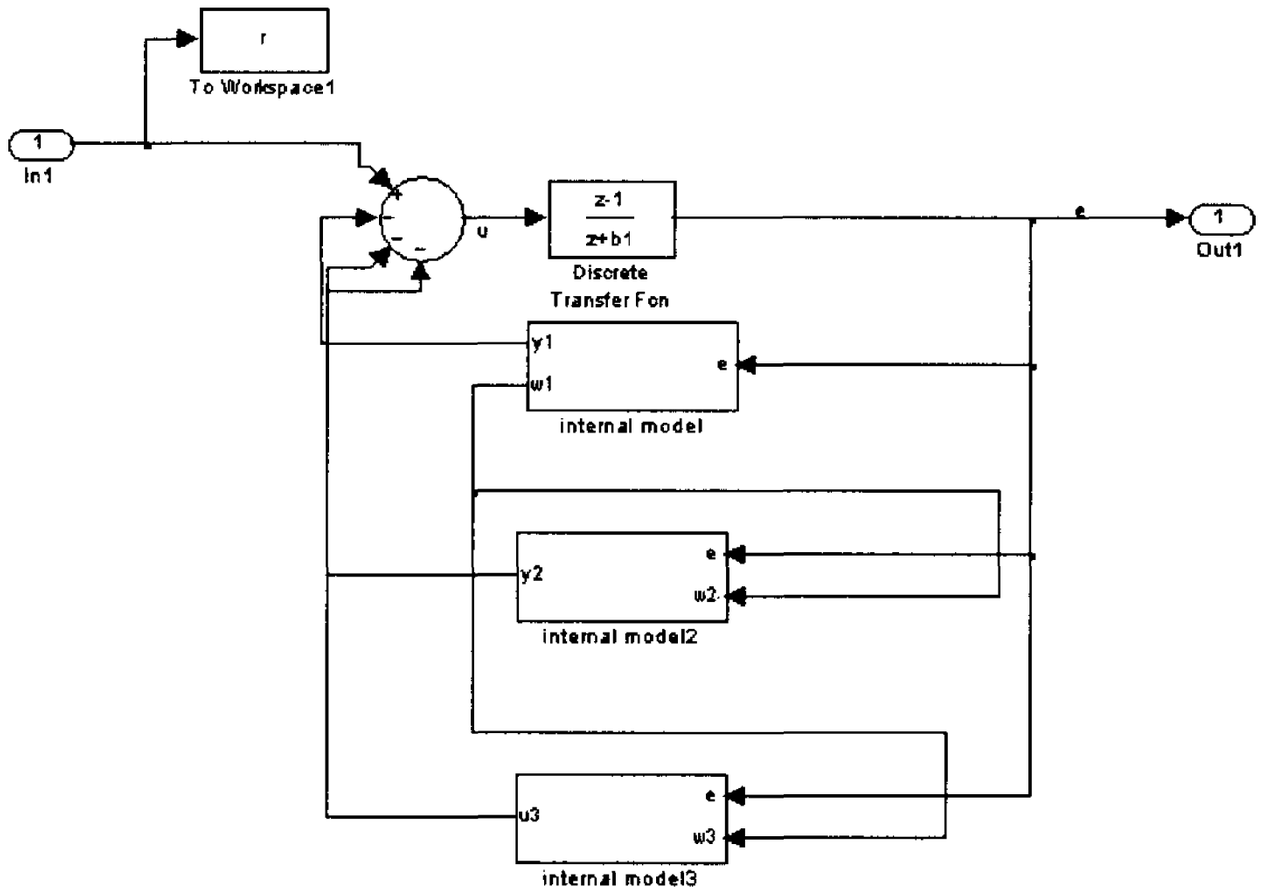

Figure 4.5: Diagram of a 7 order system in Simulink

where the three initial phase $\varphi_1, \varphi_2, \varphi_3$ are set the same value of 0.5, the magnitudes of three harmonics are set three different constant $m_1 = 1, m_2 = 0.5, m_3 = 0.25$, the frequencies of the three harmonics are time-varying $w_1(t) = 220 + 160t$, $w_2(t) = 440 + 320t$, and $w_3(t) = 660 + 480t$, and n is a Guassian white noise with PSD 0.001. In order to track the three harmonics of the reference signal, the number of paralleled IMs is set 3, and the adaptation gain K_e is set 0.06. Since the signal is a monophonic signal, by tracking the fundamental frequency, the second and third harmonics will also be tracked by multiplying the fundamental frequency by 2 and 3, thus the adaptation is only applied for the fundamental frequency. The system diagram in Simulink is illustrated in 4.5. The results for the first time forwards-in-time running is shown in Figure 4.6. As seen from the figure, the adaptive internal model based closed-loop system could track each harmonic with time-varying frequency but with delays. The delay is $0.04s$ for $w_1(t) = 220 + 160t$, about $9(0.04/(1/220) \simeq 8.8)$ cycles of the

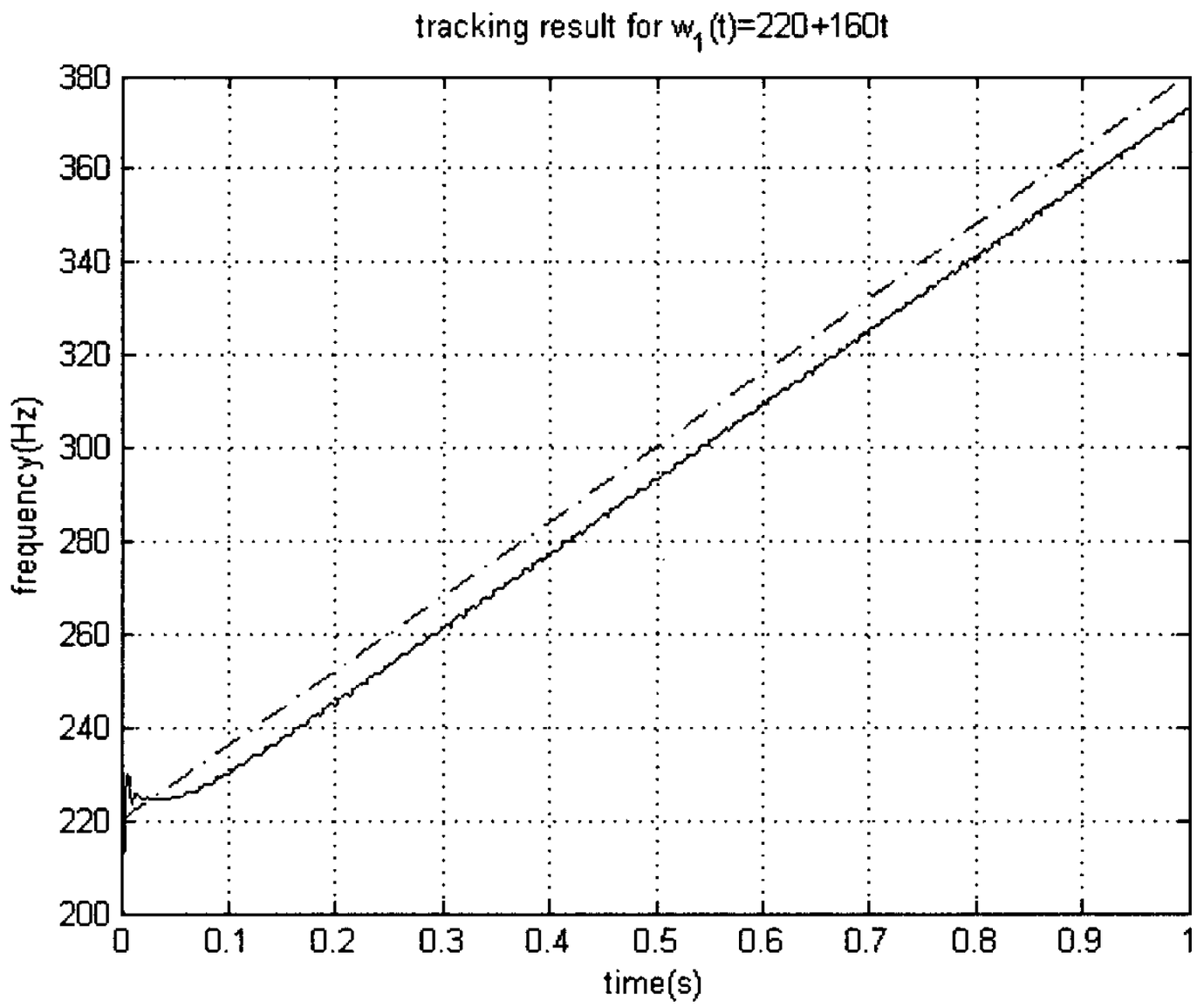

Figure 4.6: Estimated frequency for $w_1(t) = 220 + 160t$

$x_{11}(0)$	$x_{21}(0)$	$x_{12}(0)$	$x_{22}(0)$	$x_{13}(0)$	$x_{21}(0)$
-0.2913	-0.5967	0.2966	-0.1167	-0.1342	-0.2225

Table 4.2: Initial values of state variables in backwards-in-time running

signal. This delay is inevitable, since signals going through any linear system would generate delay. The transition period for the adaptation is about $0.058s$ (12.5 cycles of the signal). In the second time backwards-in-time running, the initial values of the state variables are calculated according to the methods discussed in Section 4.2.3 and illustrated in Table4.2. The initial value of the adaptation frequency is set as frequency at the ending state $w_{backwards}(0) = w_{forwards}(end)$. The results for the second time backwards-in-time running is shown in Figure 4.7. In the backwards-in-time running, the fundamental frequency component should be $w_{1b}(t) = 380 - 160t$. As seen from the figure, with correct initial value setting, the system is able to track

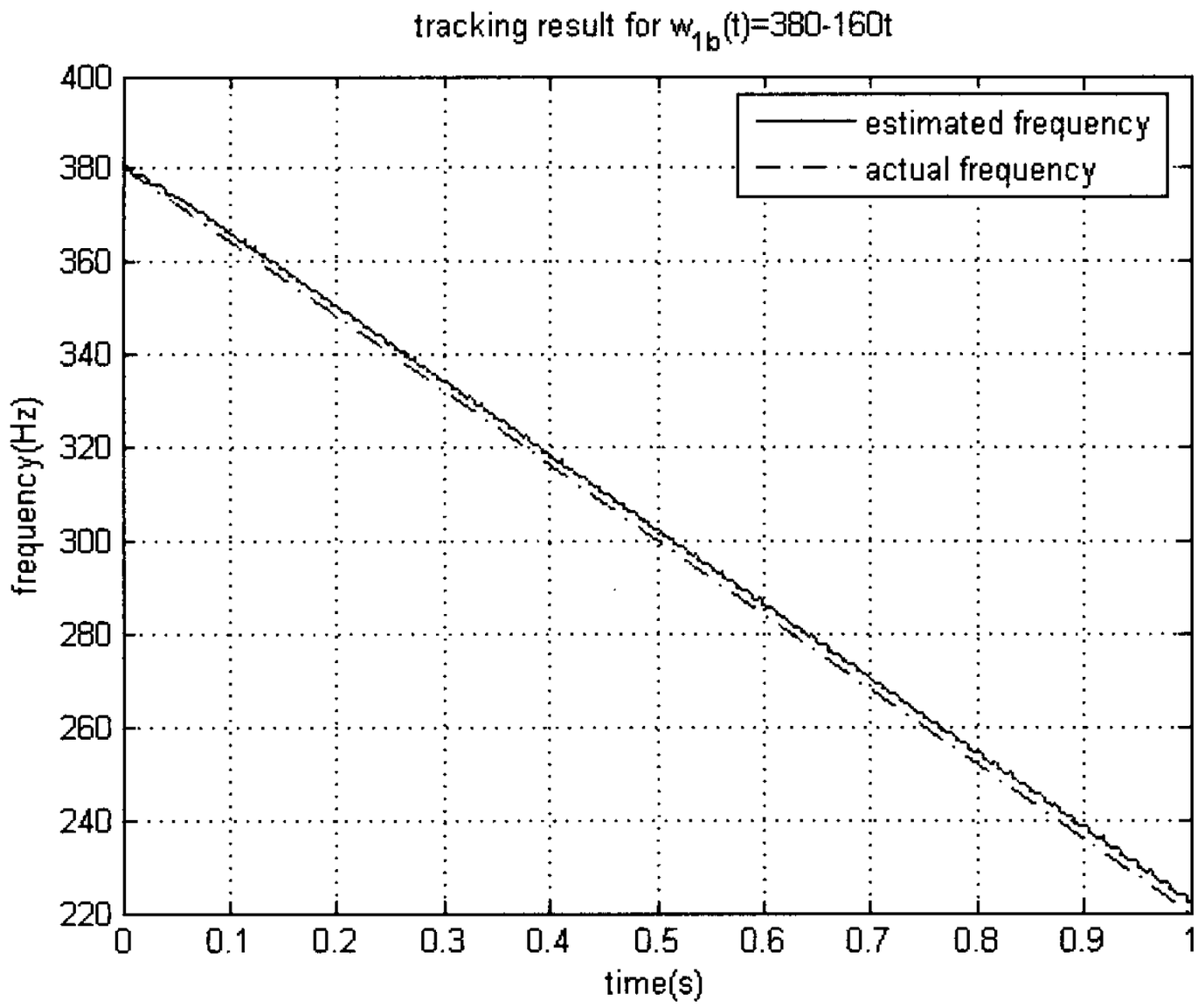

Figure 4.7: Estimated frequency for $w_{1b}(t) = 380 - 160t$

the time-varying frequency from the beginning of the signal with no transition. The delay is about 3 cycles of the signal.

Chapter 5

Conclusions and Future Work

Two applications have been presented. The first one is a modified multiple fundamental frequency estimation method based on recursive CASA. The second one is a frequency tracking system able to capture the initial part of a monophonic signal with multiple harmonics. Experiments and simulations are conducted under MATLAB and SIMULINK environment for algorithm validation.

5.1 Conclusions

A Modified Multiple Fundamental Frequency Estimation Method based on Recursive CASA is proposed in Chapter 3. The algorithm is based on a previous peer Yan's work. In order to process music signal with more than 2 concurrent notes and to improve the system performance, multiple modifications are made. For Time-Frequency analysis, the Chebyshev type I band-pass filter is changed to Chebyshev type II filter in order to reject possible noise in other frequency bands. For Note Extraction procedure, the modified Matching Probability Functions are proposed to process polyphonic signals, the flow-chart for the note extraction is also modified. A

proper evaluation criteria based on machine learning is also proposed to evaluate the performance of the system. Experiments are carried out and the results are evaluated using the proposed criteria. The system achieved an averaged 68.4% recall rate for signals with 4 concurrent notes, 73.2% for trio music, 86.6% for duo music, and 96.1% for solo music, and the precision above 90%.

A Modified Frequency Tracking system based on Adaptive Internal Model Control Theory is presented in Chapter 4. In order to track the whole period of a monophonic signal, the algorithm is run two times, with the first time forwards-in-time as normal, and the second time backwards-in-time with the reversed reference signal as input. By properly setting of the initial state variables in the second time running, perfect tracking through the whole period of the signal is achieved. Experiments are carried out on synthesized monophonic signal with 3 harmonics. The results show that the system could track the whole period of the signal in the backwards-in-time running with a slight delay.

5.2 Future Work

For modified multiple-f0 estimation algorithm, the following work is needed to be performed in future

- The frequency range of signals the algorithm is able to process is needed to extend.

  Current algorithm is able to process signal with frequency range from $116.54Hz$

to $7.04kHz$(72 semitone channels), while for real music, the frequency range is from $27.5Hz$ to $20kHz$(120 semitone channels). In order to analyze music played by other instruments, more channels need to be added into the system. However, when the number of channels is increased to 80, division by zero error would happen in MATLAB. To solve this problem, the parameter setting strategy needs to be modified.

- The current experiments are only conducted for synthesized MIDI music signals because we lack the ground-truth score of real music samples. The future work is to process real music signals, and evaluate the result with the ground-truth score.

For the frequency tracking system based on adaptive internal model theory, current work is just a start. Much more work needs to be done.

- Current algorithm sets the magnitude of each harmonic to be constant, while for real world signal, such like music signal, the magnitude is time-varying. Current system will fail to track when the energy of the harmonic is small, thus rendering the system unable to track the parts of signal with small energy.

- Current algorithm is only able to track monophonic signal with multiple harmonics. For polyphonic signals with multiple harmonics, the adaptive internal model theory needs to be modified. Another problem is that current algorithm tracks only the fundament frequency of the signal, multiplying the fundamental

frequency with integers to track other harmonics. But real world signals have more complex harmonic structure. For example, the fundamental frequency in music may not exist, and harmonics may not be integer multiple of the fundamental frequency.

Bibliography

[1] "pitch class," wikipedia, *http : //en.wikipedia.org/wiki/Pitch_class.*

[2] Y.Ma, "An iterative approach to automatic music transcription and musical signal decomposition," Ph.D. dissertation, University of Western Ontario, London, Ontarion, Canada, 2009.

[3] L. Rabiner and B.H. Juang, *Fundamentals of Speech Recognition.* Prentice Hall PTR, April 1993.

[4] L.J. Brown and Q. Zhang, "Indentification of periodic signal with unknow frequency," in *IEEE Transactions on Signal Processing*, vol.51, no.6, pp.1-9, 2003.

[5] Q.Zhang, "Periodic disturbance cancellation with uncertain frequency," M.E.Sc.thesis, The University of Western Ontario, London, Ontario, Canada, 2001.

[6] B.A.Francis and W.M.Wonham, "*The Internal Model Principle of Control Theory,*" Automatica, vol.12, pp.457-465, 1976.

[7] N.Malhotra, "Online tip voltage and dynamic resistance measurement in RSW process," Master's thesis, the University of Western Ontario, London, Ontario, Canada, 2005

[8] Y.Sun, "Instantaneous fourier series estimation," Master's thesis, the University of Western Ontario, London, Ontario, Canada, 2006.

[9] G. Zheng and S. Liu, "Automatic transcription method for polyphonic music based on adaptive comb filter and neural netword," in *IEEE International Conference on Mechatronics and Automation*, Harbin, China, 2007, pp.2592-2597.

[10] A.P.Klapuri, "Multipitch analysis of polyphonic music and speech signals using an auditory model," *IEEE Transactions on Audio, Speech, and Language Processing*, vol.16, pp.255-266, 2008.

[11] K.L. Goldstein, "An optimum processor theory for the central formation of the pitch of complex tones," *ournal of the Acoustical Society of America*, Vol 54, Dec 1973, 1496-1516

[12] M. Marolt, "Transcription of polyphonic piano music with neural network," in *IEEE Mediterranean Electrotechnical Conference*, 2000, vol.2, pp.512-515.

[13] N.E.Huang, M.Wu, W.Qu, S.R.Long, S.S.P.Shen, and J.E.Zhang, "Applications of hilbert-huang transfomr to non-stationary financial time series analysis," *Applied Stochastic Models in Business and Industry*, vol.19, pp.245-268, 2003.

[14] N.Huang, Z.Shen, S.R.Long, M.Wu, H.Shih, Q.Zhen, N.Yen, C.Tung, and H.Liu, "The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis," *The Royal Society*, vol.454, pp.903-995, 1998.

[15] Q.Zhang and L.J.Brown, "Designing of adaptive bandpass filter with adjustable notch for frequency demodulation," in *American Control Conference*, vol.4, Denver, pp.2931-2936, 2003.

[16] K.Grochenig, *Foundations of Time-Frequency Analysis*, 1st ed. Birkhauser Boston, 2000.

[17] A.P.Klapuri, A.J.Eronen, and J.T.Astola, "Analysis of the meter of acoustic musical signals," *IEEE Transactions on Speech and Audio Processing*, vol.14, pp.342-355, 2006

[18] N.Bertin, R.Badeau, and G.Richard, "Blind signal decompositions for automatic transcription of polyphonic music: Nmf and k-svd on the benchmark," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol.1, Honolulu, HI, 2007, pp.65-68.

[19] R.Keren, Y.Y.Zeevi, and D.Chazan, "Multiresolution time-frequency analysis of polyphonic music," in *IEEE-SP International Symposium on Time-Frequency and Time-Scale Analysis*, Pittsburgh, PA, 1998, pp.565-568.

[20] "Automatic transcription of polyphonic music using the multiresolution fourier transform," in *IEEE Mediterranean Electrotechnical Conference*, vol.1, TelAviv, May 1998, pp.654-657.

[21] Cohen, A Kovacevic, "Wavelets: the mathematical background", Proceedings of the IEEE Volume 84, Issue 4, Pages 514-522, 1996.

[22] D.Wang and G.J.Brown, *Computational Auditory Scene Analysis: Principles, Algorithms, and Applications*, D.Wang and G.J.Brown, Eds. IEEE Press, 2006.

[23] A.S.Bregman. *Auditory Scene Analysis: The Perceptual Organization of Sound*, 1st ed. Cambridge, MA: the MIT Press, June 1990.

[24] A.P.Klapuri. "Multiple fundamental frequency estimation based on harmonicity and spetral smoothness," *IEEE Transactions on Speech and Audio Processing*, vol.11, no.6, pp.804-816, 2003.

[25] D.K.Mellinger, "Event formation and separation in musical sound," Ph.D. Dissertation, Stanford University, Stanford, CA, December 1991.

[26] K.Kashino and H.Tanaka, "A sound source separation system with the ability of automatic tone modeling," in *Proceeding of International Computer Music Conference,* Tokyo, Japan, August 1993, pp. 248-255.

[27] K.Kashino, K.Nakadai, T.Kinoshita, and H.Tanaka, "Organization of hierarchical perceptual sounds: Music scene analysis with autonomous precessing modules and a quantitative information integration mechanism," in *International Joint Conference on Artificial Intelligence*, Montreal, Canada, 1995

[28] D.Godsmark and G.J.Brown, "A blackboard architecture for computational auditory scene analysis," *Speech Communication*, vol.27, pp. 351-366, April 1999.

[29] A.D.Sterian, "Model-based segmentation of time-frequency images for musical transcription," Ph.D. dissertation, University of Michigan, Ann Arbor, MI, 1999.

[30] C.N.dos Santos, S.L.Netto, L.W.P.Biscainho, and D.B.Graziosi, "A modified constant-Q transform for audio signals," in *IEEE international Conference on Acoustics, Speech, and Signal Processing*, vol.2, 2004, pp.469-472.

[31] Y.R.Chien and S.K.Jeng, "An automatic transcription system with octave detection," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol.2, Orlando, FL, 2002, pp.1865-1868.

[32] J.C.Brown and M.S.Puckette, "An efficient algorithm for the calculation of a constant Q transform," *The Journal of the Acoustical Society of America*, vol.92, no.5, pp.2698-2701, 1992.

[33] "midi" wikipedia, *http* :

$en.wikipedia.org/wiki/Musical_Instrument_Digital_Interface$

[34] B.C.J.Moore, *An Introduction to the Psychology of Hearing*, 5th ed. Academic Press, January 2003.

[35] A. Inesta. J.M. "Multiple Fundamental Frequency estimation using Gaussian smoothness". *Proc. of the IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, ICASSP 2008, Las Vegas, USA, 2008.

[36] S.Dixon. On the computer recognition of solo piano music. In *Proc. Australasian Computer Music Conference*, Brisbane, 2000.

[37] G.E. Poliner and D.P.W. Ellis, "A discriminative model for polyphonic piano transcription," *EURASIP Journal on Advances in Signal Processing*, 2007

[38] M. Bodson and S.C. Douglas, "Adaptive Algorithm for the Rejection of Sinusoidal Disturbances with Unknown Frequency," *Automatica*, vol.33, no.12, pp. 2213-2221, 1997.

[39] J.L. Crassidis and J.L. Junkins, *Optimal Estimation of Dynamic Systems*. Boca Raton, FL: Chapman and Hall/CRC, 2004.

[40] P.A. Nelson and S.J. Elliot, *Active Control of Sound*. New York, NY:Academic Press, 1992.

[41] F. Ben Amara, P.T. Kabamba, and A.G. Ulsoy, "Adaptive Band-Limited Disturbance Rejection in Linear Discrete-Time Systems," in *Proceedings of the American Control Conference*, Seattle, WA, 1995, pp.582-586.

[42] C.R. Fuller and A.H. von Flotow, "Active Control of Sound and Vibration," *IEEE Control Systems Magazine*, vol.15, pp.9-19, 1995.

[43] J. Lu, "Control and Identification of Narrow-band Disturbances and Signals," M.E.Sc.thesis, The University of Western Ontario, London, Ontario, Canada, 2006.

[44] Z.Y. Zhao, "Modifications and Applications to a Frequency Estimation Algorithm," M.E.Sc.thesis, The University of Western Ontario, London, Ontario, Canada, 2004

[45] J.P. Bello, L. Daudet, etc, "A Tutorial on Onset Detection in Music Signals", *IEEE Transactions on Speech and Audio Processing*, vol.13, no.5, September, 2009.

Curriculum Vitae

Name	**Quan Wen**
Place of Birth	**Linfen, Shanxi Province, China**
Year of Birth	**1985**
Post-secondary Education and Degrees	**2003-2007 B.Sc(Engineering)** **Electrical and Computer Engineering** **Peking University, Beijing, China**