Cantonese Tone Recognition Using the Hilbert-Huang Transform

by

LAM, Ying Fung

A Thesis Submitted to
The Hong Kong University of Science and Technology
in Partial Fulfillment of the Requirements for
the Degree of Master of Philosophy
in Computer Science and Engineering

January 2014, Hong Kong

Authorization

I hereby declare that I am the sole author of the thesis.

I authorize the Hong Kong University of Science and Technology to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize the Hong Kong University of Science and Technology to reproduce the thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

_____

LAM, Ying Fung

20th January 2014

Cantonese Tone Recognition Using the Hilbert-Huang Transform

by

LAM, Ying Fung

This is to certify that I have examined the above MPhil thesis
and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by
the thesis examination committee have been made.

_____

Thesis supervisor: Professor David ROSSITER

_____

Acting Department Head: Professor Siu-Wing CHENG

Department of Computer Science and Engineering

20th January 2014

Acknowledgements

I would like to thank Dr. David Rossiter for his support, guidance and patience, which made this thesis possible.

Also, I would like to thank Dr. Brian Mak and Dr. Andrew Horner for their help in forming my thesis defence committee.

In addition, I would like to thank Dr. Gibson Lam and Mr. Andrew Chung for giving me lots of valuable comments during the revision of my thesis.

Finally, I would like to thank my parents for their loving support.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF EQUATIONS

LIST OF TABLES

LIST OF LISTINGS

Cantonese Tone Recognition Using the Hilbert-Huang Transform

by Oz LAM Ying Fung

Department of Computer Science and Engineering

The Hong Kong University of Science and Technology

Abstract

Cantonese is a very popular spoken language/dialect, which is well known for its rich set of nine tones and the similarity in tone contours between its tones. Automated tone recognition of Cantonese is very challenging. Hilbert-Huang Transform (HHT) is an empirical algorithm that works on non-stationary and nonlinear signals. In this study, the performance of the HHT algorithm on the recognition of Cantonese tones for isolated syllables was examined.

In the first stage of this study, HHT was used as a frequency detection tool for syllables from the CUSYL corpus. The experimental results showed a 25% improvement in the accuracy of the fundamental frequency detection compared with peak picking the performance of the Fast Fourier transform. In the second stage of this study, the accuracy of the HHT on the CUSYL corpus was improved through experimentation with various parameters used by the core component of HHT, i.e. the Windowed Average-based Empirical Mode Decomposition (WA-BASED EMD). In the final stage of this study, Support Vector Machines (SVM) were used as binary classification tools. Pitch track information obtained by HHT together with tone information from the CUSYL corpus was used to train a set of 6 SVMs with more than 1,500 syllables. The experimental results showed a 79.08% speaker-independent tone recognition rate for isolated Cantonese syllables.

CHAPTER 1

INTRODUCTION

Speech recognition is the process of converting spoken words into patterns so that they can be used for further processing by computer. It is used for a variety of different tasks, including: controlling fighter aircraft by voice commands; training air traffic controllers; automatic translation from spoken inputs; data entry by speech; voice searching; and voice controls on mobile devices. Examples of modern and publicly accessible usage include the Google Voice Search Service [1] and the voice control system, called Siri, on iOS developed by Apple Inc. [2]. Tone recognition is part of the speech recognition process. It is required for tonal languages, like Chinese.

Cantonese is the native dialect of Chinese people that hail from the Guandong province. It is used by about 60 million people in more than 20 countries, which is about 0.9% of the world's population [3]. It is the 25th most used dialect/language in the world and it is also the 3rd most widely used dialect of Chinese, after Mandarin and Wu.

There are already many studies on speech recognition of the Mandarin dialect. But speech recognition of different languages/dialects usually requires different techniques due to the structural differences of the languages. Languages in different language families differ greatly in syllables, tone and word composition.

For example, when comparing Cantonese words with English words, their word composition is totally different. Cantonese, as a dialect of Chinese, is a descendant of the Sino-Tibetan language family. Cantonese words are formed using single syllables with a specific tone. On the other hand, English is a descendant of the Indo-European language family. English words are formed using one or more syllables without any tonal element involved. In Cantonese, the same syllable spoken with a different tone is usually a different word with a different meaning. While in English the tone of a single word does not change the lexical meaning of the word itself.

Different dialects of a language may also vary a lot in their structure. In Chinese, the set of initials, finals and tones of the words in Mandarin compared with Cantonese are significantly different. For example, there are four tones in Mandarin while there are nine tones in Cantonese.

A recent study tried to apply Hilbert-Huang Transform (HHT) to the Mandarin dialect of Chinese. The study showed that the use of HHT improved the frequency detection accuracy and the tone recognition rate of Mandarin syllables [4]. This study highlighted the potential applications of HHT and prompted the study behind this thesis. The focus of this thesis is the application of HHT for Cantonese tone recognition. As explained previously, there are significant differences between Cantonese and Mandarin in terms of the syllable and tone structures. That makes the reuse of the results of research on speech recognition of Mandarin not directly applicable to Cantonese. Furthermore, the HHT algorithm used [4] is empirical. This means the parameters and procedural details of the algorithm needs an extensive reconsideration when applying it to a different language/dialect.

In this study, we investigate the performance of the HHT algorithm to Cantonese tone recognition.

Firstly, we carry out an experiment on the different fundamental frequency detection and pitch tracking algorithms. Sine, triangle and sawtooth wave signals are used to evaluate the performance of four ‘traditional’ algorithms. The four algorithms are zero-crossing rate (ZCR), auto-correlation (AUTO-CORR), Fast Fourier Transform (FFT) and Cepstral Analysis (CEPS). FFT is found to be the best algorithm and is selected as the comparison base.

Secondly, voice samples from the CUSYL Cantonese spoken word corpus are used. The pitch tracks of some of the voice samples are manually measured for accurate reference. FFT and HHT are applied to those Cantonese syllables to obtain the fundamental frequency data. Experimental results show that the HHT has on average a 25% improvement on the mean absolute percentage error for fundamental frequency detection, when compared to the best result from the FFT comparison algorithm.

Thirdly, we examine the various arrangements of the parameters and procedural details of the Empirical Mode Decomposition (EMD). EMD is an important part of the HHT algorithm. EMD is used to decompose a voice signal into smaller components, called Intrinsic Mode Functions (IMFs), for further analysis. Its accuracy greatly affects the fundamental frequency detection. As a result, we carry out two experiments which help optimize the parameters needed by the WA-BASED EMD (an improved version of EMD) and hence improve its accuracy when applied to Cantonese voice samples.

By the design of the algorithm, WA-BASED EMD requires an initial guess of the fundamental frequency of the input signal. The IMFs decomposed by WA-BASED EMD depend a lot on the initial guess. A good guess usually produces a set of clean IMFs. Several methods for initial guessing of the fundamental frequency are compared, and the best among them is chosen as the default choice of the initial guessing method of WA-BASED EMD for the next experiment.

Another important procedural detail is the *stoppage criterion* of the *sifting process* of EMD. The sifting process is a sub-procedure used for obtaining the IMFs. Usually to decompose a signal into IMFs, the number of sifting needed is 10-100 times the number of IMFs extracted. Having a suitable stoppage criterion can reduce the sifting needed, hence reducing the computational cost. Different stoppage criterions are compared. The best is chosen as the default choice of the stoppage criterion of the WA-BASED EMD for the next experiment.

Lastly, we combine the results from the previous experiments and use HHT to obtain the pitch tracking of Cantonese voice samples. A set of Support Vector Machines (SVMs) is trained with the pitch tracking results as classifiers. The SVMs are then used to classify the tone of the voice samples from the CUSYL corpus. The results suggest an average of 79.08% correctness for Cantonese tone recognition.

In this thesis, background information concerning the human voice, the Cantonese dialect and speech recognition are covered in Chapter 2. Chapter 3 gives an overview of algorithms used in digital signal processing and speech processing. The details of how the different algorithms are used for fundamental frequency detection and pitch tracking are described in Chapter 4. Related work is reviewed in

Chapter 5. Chapter 6 describes the objective, methodology and the related experiments and their results. Conclusions and suggestions for future work are given in Chapter 7.

CHAPTER 2

BACKGROUND

2.1 Overview

In this chapter, we will look at the background information of the human voice and its production, the Cantonese dialect and speech recognition

2.2 The Human Voice

In 1970, Gunnar Fant proposed a simplified model for human voice production [5]. The model assumes the voice signal is a result of transformation of a periodic pulse train produced by air passing through the vocal cords by vocal tract resonances. Mathematically, a voice signal $s(t)$ could be formulated by:

$$s(t) = p(t) * h(t)$$

Equation 1 Simplified Model for Human Voice by Fant

where $p(t)$ is the periodic pulse train; (t) is the transform response; and where $*$ is the convolution operation.

In this model, the change of pitch is mainly related to the pulse train. Hence pitch detection in this model requires finding the fundamental period or fundamental frequency of the pulse train $p(t)$.

2.3 Cantonese

There are about 60 million native Cantonese speakers all over the world, which accounts for about 0.9% of the world population. Cantonese is the native dialect for the majority of people living in Guangdong Province, Hong Kong and Macau. Well-known for its richness in tones and comparatively more complicated tonal system, automatic speech recognition for Cantonese is far more difficult than that for other dialects of Chinese.

2.3.1 Syllable Structure

The structure of a Cantonese syllable is shown in Figure 1. In Cantonese there are 19 initials and 53 finals, which are listed in Table 1 and Table 2. Every syllable consists of an optional initial and a final, which form the sound of that syllable. A final consists of a vowel and an optional terminal. When a syllable is pronounced, a particular tone is used. In Cantonese, all 19 initials are consonants. There are about 620 initial and final combinations and about 1761 syllables in Cantonese [6].

<table>
<tr><td colspan="3">Tone</td></tr>
<tr><td rowspan="2">Initial
(optional)</td><td colspan="2">Final</td></tr>
<tr><td>Vowel</td><td>Terminal
(optional)</td></tr>
</table>

Figure 1 Cantonese Syllable Structure

Initial		
LSHK	IPA	Example Word
b	/p/	巴
p	/p^h/	怕
m	/m/	媽
f	/f/	花
d	/t/	打
t	/t^h/	他
n	/n/	那
l	/l/	啦
g	/k/	家
k	/k^h/	卡
ng	/ŋ/	牙
h	/h/	蝦
gw	/k^w/	瓜
kw	/k^{wh}/	誇
w	/w/	蛙
z	/ts/	渣
c	/ts^h/	叉
s	/s/	沙
j	/j/	也

Table 1 List of Cantonese Initials. LSHK is the Cantonese Romanization Scheme proposed by the Linguistic Society of Hong Kong. IPA is the International Phonetic Alphabet.

Final			Final		
LSHK	**IPA**	**Example Word**	**LSHK**	**IPA**	**Example Word**
aa	/aː/	沙	ing	/ɪŋ/	永
aai	/aːi/	乖	ip	/iːp/	蝶
aau	/aːu/	貓	it	/iːt/	鐵
aam	/aːm/	三	ik	/iːk/	的
aan	/aːn/	單	o	/ɔː/	拖
aang	/aːŋ/	丁	oi	/ɔːi/	才
aap	/aːp/	甲	ou	/ɔːu/	肚
aat	/aːt/	八	on	/ɔːn/	漢
aak	/aːk/	白	ong	/ɔːŋ/	康
ai	/ɐi/	西	t	/ɔːt/	割
au	/ɐu/	收	ok	/ɔːk/	學
am	/ɐm/	心	u	/uː/	虎
an	/ɐn/	新	ui	/uːi/	胚
ang	/ɐŋ/	等	un	/uːn/	寬
ap	/ɐp/	入	ung	/ʊŋ/	宋
at	/ɐt/	七	ut	/uːt/	闊
ak	/ɐk/	得	uk	/ʊk/	叔
e	/ɛː/	卸	oe	/œː/	靴
ei	/ei/	非	oeng	/œːŋ/	娘
eu	/ɛːu/	掉	oek	/œːk/	卻
em	/ɛːm/	舐	eoi	/ɵy/	佢
eng	/ɛːŋ/	鄭	eon	/ɵn/	潤
ep	/ɛːp/	夾	eot	/ɵt/	出
ek	/ɛːk/	石	yu	/yː/	鼠
i	/iː/	詩	yun	/yːn/	短
iu	/iːu/	消	yut	/yːt/	雪
im	/iːm/	閃	m	/m̩/	唔
in	/iːn/	先	ng	/ŋ̩/	五

Table 2 List of Cantonese Finals

2.3.2 Tones

In Cantonese, there are 9 tones in 6 contrastive contours. 3 of them are entering tones which only appear in syllables with the terminal /p/, /t/, /k/. Table 3 is a summary of the 9 Cantonese tones. Distinguishing the entering tones from the non-entering tones is relatively easy since they have specific terminals. Also syllables with entering tone are significantly shorter in term of duration when compared to syllables with non-entering tones. In this study, we focus on distinguishing between tones 1 to 6.

Tone Number	1	2	3	4	5	6	7	8	9
Tone Name	High level	Mid rising	Mid level	Low falling	Low rising	Low level	Entering high level	Entering mid level	Entering low level
Example Word	分	粉	訓	焚	奮	份	忽	發	佛
LSHK	fan1	fan2	fan3	fan4	fan5	fan6	fat1	faat3	fat6
IPA	/fɐn1/	/fɐn2/	/fɐn3/	/fɐn4/	/fɐn5/	/fɐn6/	/fɐt1/	/fa:t3/	/fɐt6/

Table 3 Summary of Cantonese Tones.

2.4 Fundamental Period and Fundamental Frequency

For a periodic signal, the smallest unit we needed to completely describe it is a complete period of the signal. For a periodic signal $x(t)$, the value of it is the same after a shift in unit time T, that is:

$$\boldsymbol{x(t) = x(t + nT)\ \forall n \in \mathbb{Z}}$$

Equation 2 Definition of Periodic Signal

T is called the period of $x(t)$. Obviously multiples of T must also be the period of $x(t)$. Most of the time we are interested in finding the smallest value of T, which we call T_0. This is called the fundamental period of a signal. From the definition, the frequency of a signal is the reciprocal of its fundamental period. Hence the fundamental frequency, F_0, is the reciprocal of the fundamental period. For a signal that is composed by multiple periodic signals with different frequencies, the fundamental frequency is the frequency of the component with the lowest frequency.

2.4.1 Missing Fundamental

The 'missing fundamental' is the phenomenon where a signal has the overtones (higher harmonics) but lacks the fundamental frequency (first harmonic) [7]. Pitch detection performed by the human brain is achieved not only by tracing the fundamental frequency but also the ratio of the higher harmonics. For example, the pitch perceived by the human brain of a periodic signal with peaks at 440Hz, 660Hz, 880Hz, 1100Hz is 220Hz even though there is no peak at 220Hz.

2.4.2 Voiced and Unvoiced Sound

A sound is classified as voiced if the vocal cords vibrate when the air pass through. Otherwise the sound is classified as unvoiced. In Cantonese, all the nasal initials (/m/, /n/, /ng/) and all the finals are voiced and all the non-nasal initials are unvoiced. As mentioned in section 2.1, the pitch of a sound is mainly related to the pulse train, therefore only the voiced part of a syllable contains useful pitch information.

2.5 Closing Comments

In this chapter, we have looked at the background information of human voice production of the Cantonese dialect. We have also discussed the definition of fundamental frequency and how it is related to tone in tonal languages. In the next chapter, we will have a look at the traditional algorithms that are used for speech processing. Also, the details of the Hilbert-Huang transform will be discussed.

CHAPTER 3

ALGORITHMIC TECHNIQUES FOR SPEECH PROCESSING

3.1 Overview

In this chapter, we review different algorithms that have been used in speech processing. The algorithmic techniques we consider are:

1. Zero-crossing rate
2. Auto-correlation
3. Fourier transform
4. Cepstrum
5. Wavelet transform
6. Hilbert-Huang transform

The first two and the last one, i.e. zero-crossing rate, auto-correlation and Hilbert-Huang transform, work in the time domain. The other three, i.e. Fourier transform, Cepstrum and Wavelet transform, work in the frequency domain.

In the list above, instead of grouping the algorithms by the domain (time domain/ frequency domain) they work in, the order of the technique is based on the year they are first proposed and used in signal processing, which is also the order of complexity of the algorithm.

We will consider the basic concept of each algorithm and their application to speech processing, fundamental frequency detection and pitch tracking. Accuracy, processing speed and complexity of implementation are the major consideration.

3.2 Zero-crossing Rate

Zero-crossing is a point where the sign of the amplitude of a signal changes e.g. from positive to negative. By the definition of fundamental frequency in section 2.4, counting the number of zero-crossings could be used as a method for estimating the fundamental frequency of a signal. In simple zero-mean periodic signal, the time interval between three adjacent zero-crossing points is equal to the period length of the signal in the ideal case. For example, Figure 2 shows a complete period of a sine function and its zero-crossing points.

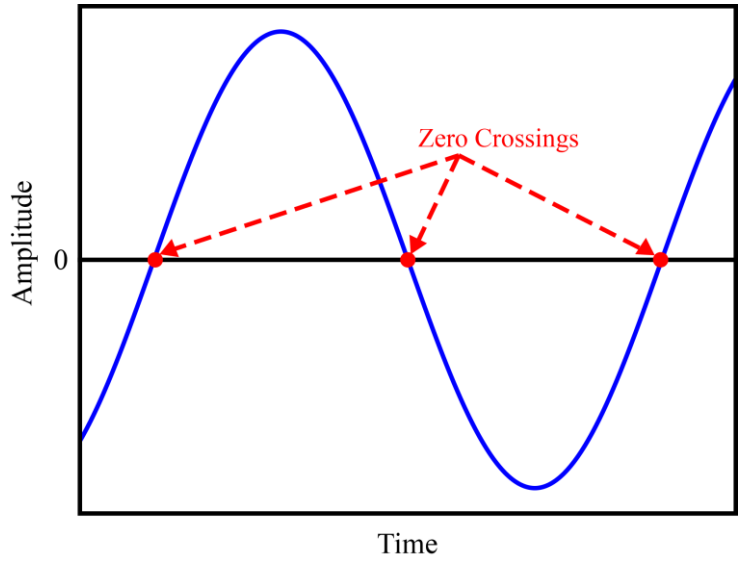

Figure 2 Zero-crossing Points of a Signal

This algorithm is simple, fast and easy to implement, but has a very high error rate when working with signal with noise present.

3.3 Auto-correlation

Auto-correlation (ACF) is a mathematical tool works by finding the similarity of a signal with itself. For a signal $x(t)$, the auto-correlation function $ACF_{x(t)}(\tau)$ is defined as:

$$ACF_{x(t)}(\tau) \stackrel{\text{def}}{=} \int_{-\infty}^{\infty} x(t)x(t-\tau)dt$$

Equation 3 Definition of Auto-correlation Function

where τ is the lag between the two copies of $x(t)$.

By definition, Equation 3 is the *convolution* of $x(t)$ and $x(-t)$ where convolution of two functions $f(t)$ and $g(t)$ is defined as:

$$(f * g)(\tau) \stackrel{\text{def}}{=} \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau$$

Equation 4 Definition of Convolution

The convolution of discrete functions $x[n]$ and $y[n]$ is defined as:

$$(x * y)[n] \stackrel{\text{def}}{=} \sum_{m} x[m]y[n-m]$$

Equation 5 Definition of Discrete Convolution

For example, x[n] is the pulse train and y[n] is vocal tract response in human voice production.

Hence the ACF on a discrete signal $x[n]$ is defined as:

$$ACF_{x[n]}(\tau) \stackrel{\text{def}}{=} \sum_{\tau} x[\tau]x[n-\tau]$$

Equation 6 Definition of ACF for Discrete Signal

Figure 3 shows a signal with 100 samples and its ACF. The signal is a sine function with random white noise added. In this example, the peaks are found at the lag $\tau = 0$ and approximately at the lag $\tau = 10, 20, 30, etc.$. That means the signal has high similarity with itself at those lag values. Under the assumption that the input

signal is a periodic signal, we can conclude that the period length of the input signal is 10 samples.

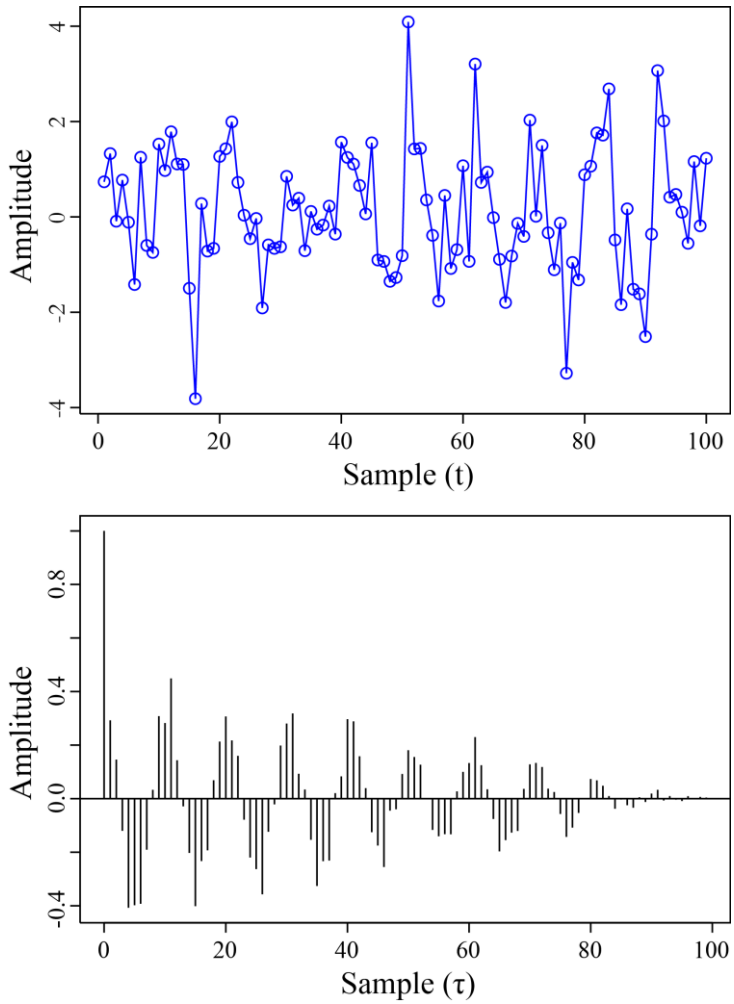

Figure 3 Plot of a Signal (top) and its ACF (bottom)

Calculating the ACF of a N points signal $x(t)$ can be done in $O(N^2)$ arithmetical operations. It can be simply implemented by using two loops. Advanced programming languages, such as MATLAB, provide a convolution function as a basic component.

Auto-correlation can be used to reveal the information about repeating events in a signal. For example, it can be used to determine the pitch of a musical tone. Similarly, it can be used to estimate the frequency of the periodic pulse train in a speech signal.

One advantage of ACF is that it is more resistant to noise compared to the zero-crossing method.

3.4 Fourier Transform

A periodic function could be expressed as the sum of a series of sine and cosine functions. These sine and cosine functions are the frequency composition of the signal. Fourier transform (FT) is one of the mathematical tools that decomposes a signal into its frequency components.

A frequency spectrum represents a time domain signal in the frequency domain. The result of an FFT is a frequency spectrum. A frequency spectrum can clearly show the composition of a signal in terms of the contribution of different frequency components.

3.4.1 Fourier Series

A periodic signal function $x(t)$ with period 2π, i.e. $x(t + 2\pi n) = x(t)\ \forall n$, when expressed as the sum of a set of simple harmonic oscillating function (sines and cosines) in the form of:

$$x(t) = \frac{1}{2}a_0 + \sum_{n=1}^{\infty}(a_n \cos(nt) + b_n \sin(nt)),$$

$$where, \quad a_0 = \frac{1}{\pi}\int_{-\pi}^{\pi} x(t)\,dt$$

$$a_n = \frac{1}{\pi}\int_{-\pi}^{\pi} x(t)\cos(nt)\,dt$$

$$b_n = \frac{1}{\pi}\int_{-\pi}^{\pi} x(t)\sin(nt)\,dt$$

$$n = 1, 2, 3, \ldots$$

Equation 7 Definition of Fourier Series of a Periodic Signal $x(t)$

is called the Fourier series of $x(t)$. The coefficients a_n and b_n measure the contribution from each harmonic.

3.4.2 Fourier Transform

Fourier transform is the generalization of the process of finding the Fourier series of a periodic signal with period 2π to any arbitrary periodic signal with period $2L$ and any arbitrary aperiodic signal. The Fourier transformed signal in the frequency domain, $X(\omega)$, of a time domain signal $x(t)$, is defined as:

$$X(\omega) = \int_{-\infty}^{\infty} x(t) e^{-2\pi i \omega t} dt$$

Equation 8 Definition of Fourier Transform of a Signal $x(t)$

When the independent variable t represents time, the dependent variable ω represents frequency.

In the rest of this thesis, $\mathcal{F}\{x\}$ will be used to denote the Fourier transform of a signal $x(t)$.

3.4.3 Discrete Fourier Transform

The discrete Fourier transform (DFT) is the Fourier transform applied to a finite discrete signal. It gives a result which is discrete in the frequency domain. For a discrete signal $x[n]$ with sampling rate f_s, its DFT is defined as:

$$X_k = \sum_{n=0}^{N-1} x[n] e^{\frac{-2\pi i}{N} kn}, \qquad k = 0, 1, \dots, N-1$$

Equation 9 Definition of Discrete Fourier Transform of a Discrete Signal $x[n]$

where N is the number of samples of the signal $x[n]$. The frequency spectrum also has N points. X_0 and X_{N-1} represent the contribution of 0Hz and f_s Hz component respectively. Each point is separated by f_s/N Hz.

X_k is even symmetrical around the central point $k = N/2$. So the useful region of X_k is only from $k = 0$ to $k = N/2$. Hence the DFT can only capture frequency components up to $f_s/2$ Hz.

For example, let us consider a 4,000 sample length signal with sampling rate 16,000Hz. Its DFT has 4,000 points, where X_{1999} represents the contribution of the 8,000Hz component. Only the first 2,000 points of the DFT is useful due to the symmetric property. This means that the DFT can only capture frequency components up to 8,000Hz.

Instead of FT, DFT is needed in signal processing in the processing of digital signals since the signals are always discrete.

3.4.4 Fast Fourier Transform

Fast Fourier transform (FFT) is the classification of the most efficient algorithm to compute the DFT of a discrete signal.

Computing a DFT of a N points signal by Equation 9 takes $O(N^2)$ arithmetical operations while FFT can compute the same result efficiently in $O(NlogN)$ operations.

There are many different algorithms of FFT. One of the most common FFT algorithms is the Cooley-Tukey algorithm [9]. It is a divide and conquer algorithm which takes a N points input where N is a power of 2. It first divides the input into two halves and applies FFT to each half separately and recursively. After that it combines the results from the two halves to produce the final result. A more detailed explanation of the Cooley-Tukey algorithm can be found in [9].

3.4.5 Short-time Fourier Transform

The Short-time Fourier Transform (STFT) can show the trend of the changes of the fundamental frequency and other higher harmonics. It can be used as a pitch-tracking tool. To increase the accuracy and the resolution of the frequency decomposition, a signal is usually divided into many short segments. FT is applied to each of them instead of applying FT to the entire signal. Every segment reveals local frequency information of the signal across its original time domain span. The result

is an energy-time-frequency spectrum. The segments may or may not be overlapped. The plot of the amplitude of the result of STFT is a *spectrogram*.

Figure 4 shows a human voice signal and its spectrogram calculated by STFT. The x axis in the plot in figure 4(b) is the n^{th} segment, the y axis is the frequency in log scale of the FT of each segment. The grey-scale displays the amplitude of the contribution of each frequency component. Dark grey represent a higher value while light grey represent a lower value.

There are many FFT implementations freely available for different platforms and programming languages [10][11][12]. There is also hardware developed for performing FFT [13]. STFT is just a simple loop which can be implemented within 20 lines of code if FFT is already available as a function.

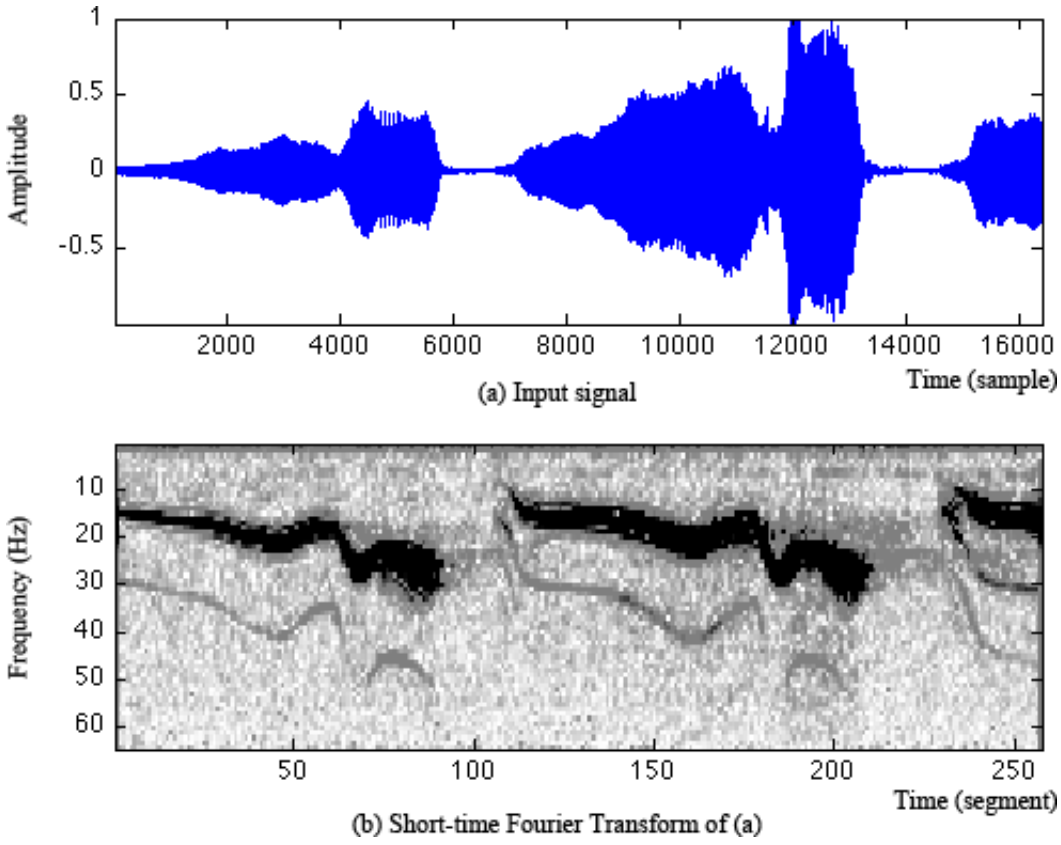

Figure 4 Plot of (a) a Human Voice Signal and (b) its spectrogram by Short-time Fourier Transform

3.4.6 Spectral Leakage

In the ideal case, the number of periods of a signal should be an integer. When the number of periods of the signal is not an integer, the spectrum obtained will show energy leakage around the peaks. Those unwanted components are called side lobes. Figure 5 shows the spectrums with and without leakage of a function of ω_0 Hz.

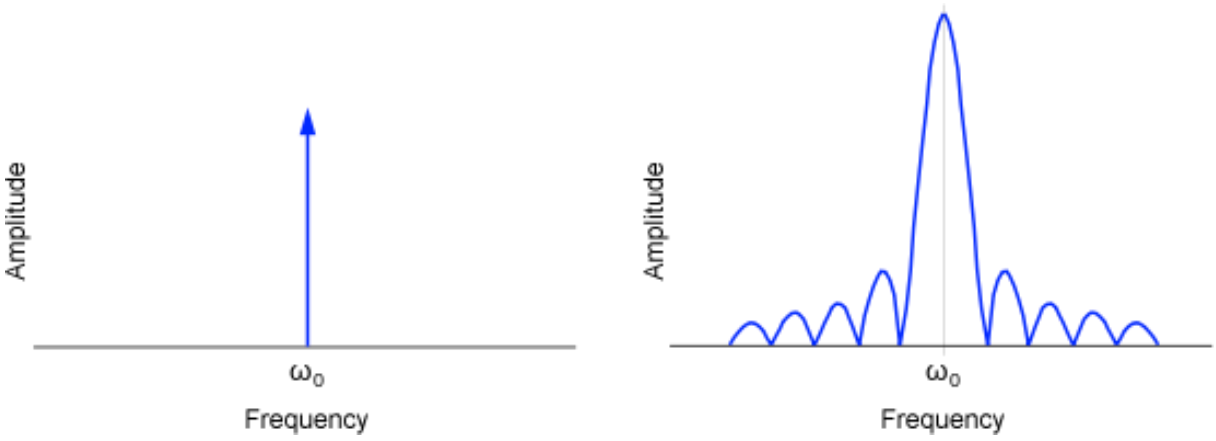

Figure 5 Spectrum of a 200Hz Sine Wave with Integer Number of Periods (left) and spectrum of a 200Hz Sine Wave with Non-integer Number of Periods (right)

For example, the spectrum of a 200Hz sine wave with an integer number of complete periods e.g. 7 shows a single stem at 200Hz and energy contributed by other frequencies are zero. In contrast, the spectrum of a 200Hz sine wave with a non-integer number of complete periods e.g. 7.3 periods has its peak at 200Hz, with energy leaked to the side lobes. This is illustrated in Figure 5(a) and 5(b) respectively.

3.4.7 Hamming Window

The Hamming window was proposed by mathematician Richard W. Hamming [14]. It is one period of a raised cosine function. The formula of the Hamming window is:

$$w(n) = 0.54 - 0.46\,cos\left(\frac{2\pi n}{N-1}\right)$$

Equation 10 Definition of the Hamming Window

where N is the width (in number of samples) of the Hamming window and $0 \leq n \leq N-1$.

The Hamming window can help reducing the spectral leakage problem by minimizing the side lobes in the spectrum. By applying the hamming window before performing the FFT on the signal segment, the leakage could be reduced. Figure 6 shows the plots of a Hamming window of 64 samples width in the time and the frequency domain on the left and right respectively.

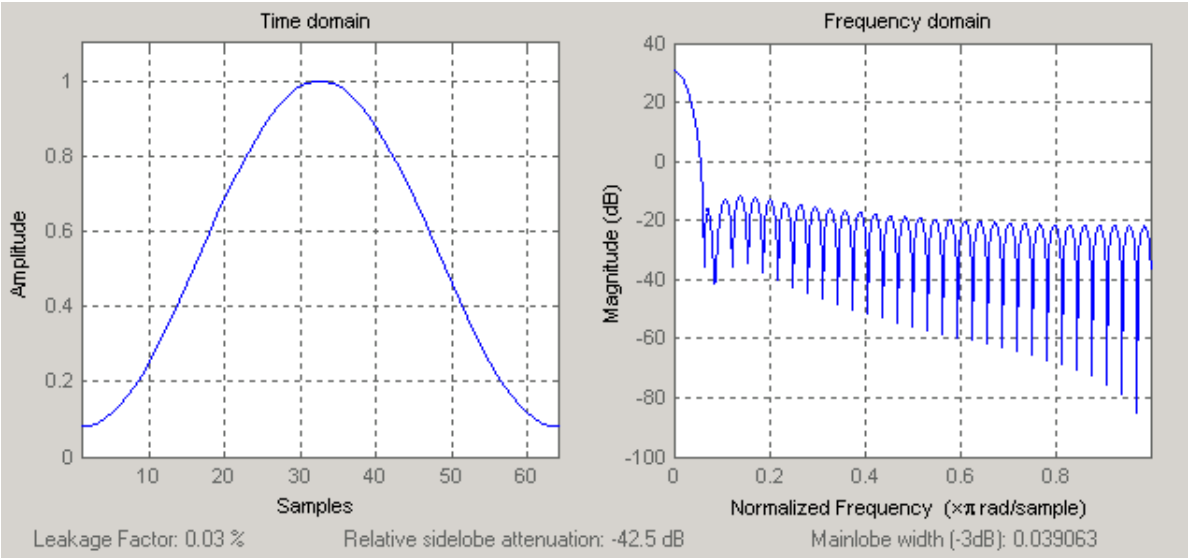

Figure 6 Plot of a 64 samples Hamming Window in Time Domain (left) and the corresponding Frequency Domain (right) [15]

3.5 Cepstrum

Cepstrum was first introduced by Bogert *et al.* in 1963 [16][17]. The basic idea of Cepstrum is to take a spectrum as input and apply FFT to obtain a spectrum of spectrum. By doing so, we can examine the properties that may be hidden in the

spectrum. For example, in the cepstrum, we could reveal the fundamental of a spectrum that has missing fundamental occurred.

By taking the Fourier transform of a spectrum in log scale as if it is a time domain signal, a cepstrum is obtained. Formally, the cepstrum $C(q)$ of a signal $x(t)$ is defined as:

$$C(q) = \mathcal{F}\{log|\mathcal{F}\{x(t)\}|\}$$

Equation 11 Definition of Cepstrum of a Signal $x(t)$

As mentioned in section 3.4.2, $\mathcal{F}\{x\}$ is the Fourier transform of the signal $x(t)$. The dependent variable q in the equation represents *quefrency*. Quefrency is a measurement of time in milliseconds (ms).

Let consider a specific example. For a simple periodic signal $x(t)$ with frequency $f_0 = 200$ Hz, $N = 8{,}192$ samples length and sampling rate $f_s = 16{,}000$ Hz, we can obtain a frequency spectrum $X(\omega)$ of $M = N/2 = 4096$ coefficients ranging from 0 to $f_s/2 = 8{,}000$Hz. Each pair of the adjacent coefficients is separated by $f_s/N \approx 1.95$Hz. Considering the frequency spectrum in log scale, $log|X(\omega)|$, as a periodic signal with period $F_0 = 100$ samples (which is ≈ 200 Hz), and applying Fourier transform on it gives the cepstrum $C(q)$ of $x(t)$. The sampling rate of this frequency spectrum is $2M/f_s$ samples/Hz (≈ 0.512 samples/Hz).

Same as spectrum discussed in section 3.4.3, only half of the coefficients in $C(q)$ are useful due to the even symmetric property of the cepstrum. Hence we have a cepstrum of $M/2 = 2{,}048$ coefficients ranging from 0 to 4,000 samples. When the quefrency is converted back to frequency, the frequency distribution is in a non-linear scale since $1/f = q * f_s$, hence $f = f_s/(q * N)$. For our example, $C(q)$ shows a large peak at $q_0 = 80$ samples, which indicates that the spectrum has a period every 80 samples. The fundamental frequency is $f_0 = f_s/q_0$, which for our example is 200Hz.

Figure 7 shows another example. The figure shows a 240Hz sawtooth wave signal, its frequency spectrum, its frequency spectrum in log scale and its cepstrum.

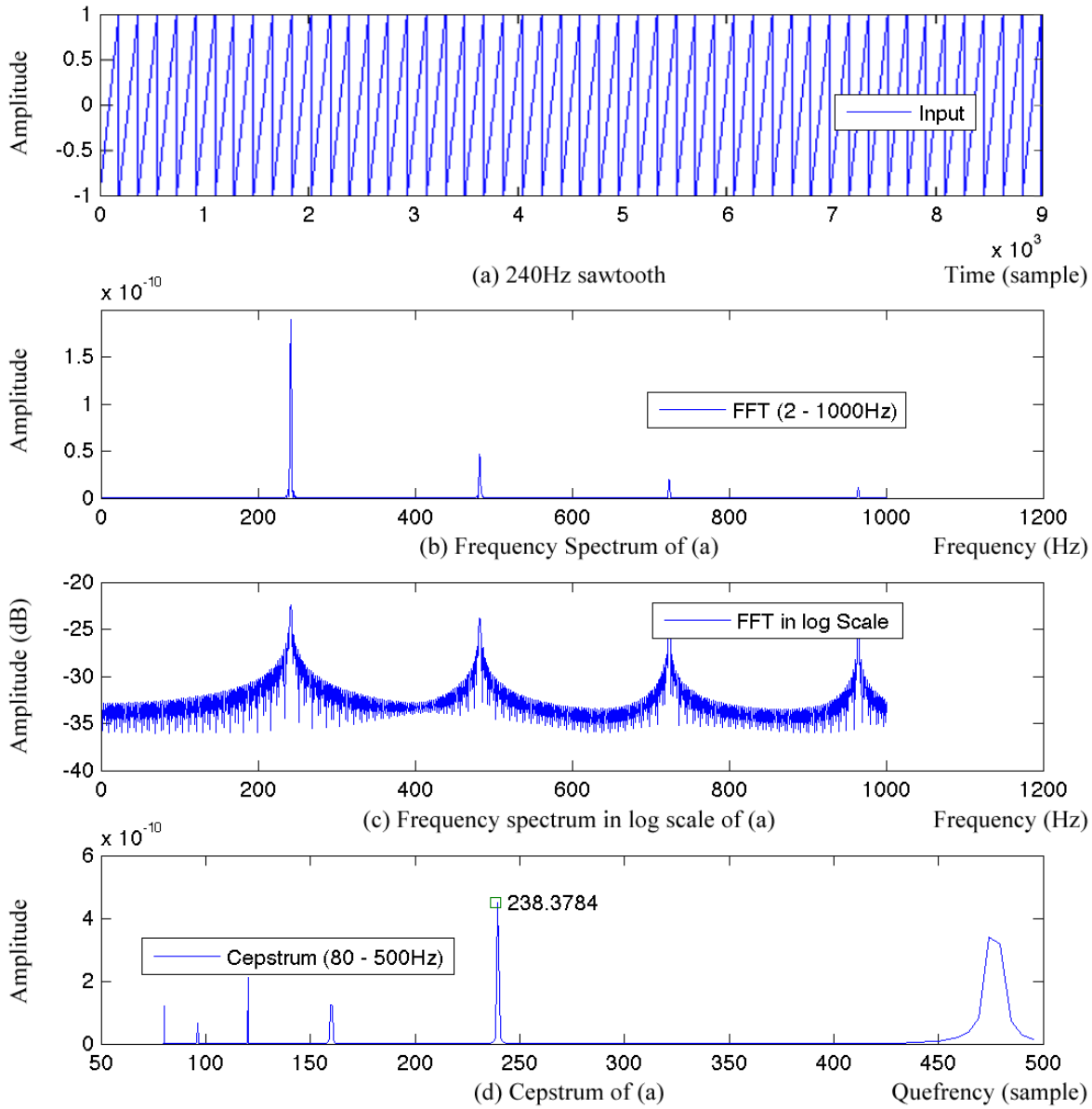

Figure 7 Plot of (a) a 240 Hz Sawtooth Wave, (b) its Frequency Spectrum, (c) its Frequency Spectrum in *log* Scale and (d) its Cepstrum

A cepstrum can be used to estimate the fundamental frequency for a signal which is missing its fundamental frequency component in the spectrum. Moreover, as described in section 2.1, a voice signal could be modeled as

$$x(t) = h(t) * p(t)$$

where $p(t)$ is the quasi-periodic signal of the pulse train. After Fourier Transform, the voice signal model becomes

$$\mathcal{F}\{x\} = \mathcal{F}\{h\} \cdot \mathcal{F}\{p\}$$

After applying log function to the magnitude and then the second Fourier Transform, the cepstrum model of the voice signal becomes

$$\boldsymbol{C\{x\} = \mathcal{F}\{log|\mathcal{F}\{h\}|\} + \mathcal{F}\{log|\mathcal{F}\{p\}|\} = C\{h\} + C\{p\}}$$

Equation 12 Cepstrum of the Human Voice Model by Fant

which is a superposition of the cepstrum of the impulse response and pulse train.

Since the impulse response $h(t)$ is relatively fast changing, the corresponding cepstrum component is expected to be located in the lower quefrency range of the cepstrum. In contrast, the pulse train $p(t)$ is relatively slow changing. The corresponding cepstrum component of the pulse train is expected to be located in higher quefrency range of the cepstrum. This property allows us to separate the pulse train from the impulse response.

Although the cepstrum method can separate the pulse train from the impulse response and also find the fundamental frequency of signal which has missing fundamental, its resolution is greatly limited by the resolution reduction in the two successive Fourier transform.

In the rest of this thesis, $C\{x\}$ will be used to denote the cepstrum of the time domain signal $x(t)$.

3.6 Wavelet Transform

Wavelet was first developed in 1910 by Alfed Haar as a mathematical tool [18].

3.6.1 Overview

The concept of wavelet transform (WT) was proposed by Jean Morlet in 1981. Wavelet transform is the process of producing wavelets from data. Wavelets are mathematical functions representing different frequency components of data. They

are each studied with a resolution matching their scale. Wavelet transform has the advantage in analyzing physical situations where the signal contains discontinuities and sharp spikes.

Comparing to STFT that uses fixed width windows, a wavelet transform scales according to the frequency of the current signal segment. The windows used in a wavelet transform scale with the frequency thus give a non-linear time domain resolution while the accuracy on the frequency domain improves. Figure 8 shows the difference of the division of the time and frequency domain of Fourier transform and wavelet transform. Every resolution cell in STFT is equally weighted in both time and frequency domain while those in wavelet transform varies. In wavelet transform, the area (number of samples) of each cell is the same while the height (span of frequency band) and width (span of time) of each cell is changing. The shorter the cell in the time domain, the lower the resolution in the frequency (scale-value) axis.

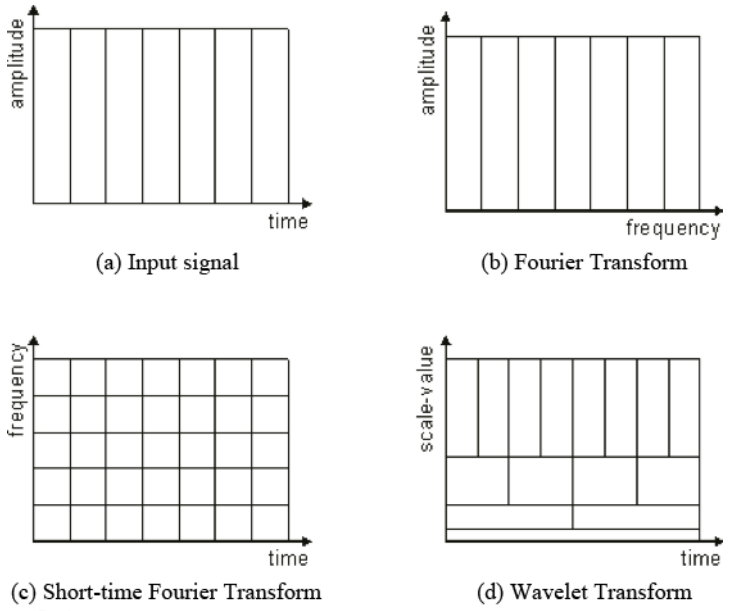

Figure 8 Resolution Cell of (a) Input Signal (b) Fourier Transform (c) Short-time Fourier Transform and (d) Wavelet Transform [19]

For example, in figure 8 (d), let us assume each cell has 32 samples and the scale-value is from 0 to f_s. The top row (i.e. the high scale-value) has 8 cells and hence 8 results in the time domain, where each cell has 32 divisions on the frequency resolution ranging from $f_s/2$ to f_sHz. Each division spans $f_s/64$ Hz. On the other

hand, the bottom row (i.e. the low scale-value) has 1 cell only and hence 1 result in the time domain. However, it has relatively many more divisions in the frequency axis. Specifically, there are 32 divisions on the frequency axis ranging from 0 to $f_s/8$Hz. Each division spans $f_s/256$Hz.

3.6.2 Wavelet Transform in Signal Processing

Wavelet transform is an improved version of Fourier transform. Fourier transform and frequency spectrum are very useful tools for signal analysis in the frequency domain, but they fail to handle non-stationary signals accurately. In contrast, a wavelet transform can decompose a non-stationary signal into elementary components at different positions with different scales.

Although it is useful in breaking down the signal into different compositions, the energy-time-frequency information is not always better than that of STFT in the accuracy of pitch detection, especially in speech analysis. The Wavelet transform of a speech signal only has good frequency resolution in the lower scale range. The time domain resolution is even worse than the STFT. Due to the fact that the range of the fundamental frequency of human voice is mainly located in the lower scale range, in this study we do not use wavelet transform since tone recognition needs a pitch track with reasonable time domain resolution.

When compared to FT, there is less available implemented code freely available. There is no hardware developed for performing WT at the time of writing.

3.7 Hilbert-Huang Transform

While many tools assume the signal is linear and stationary, Hilbert-Huang Transform (HHT) [20] does not. It provides a method for analyzing non-stationary and nonlinear data. It was first proposed in 1996.

3.7.1 Overview

Similar to spectrum and cepstrum methods which decompose a signal into sinusoids, the first step of HHT is to decompose the signal into finite and often small quantities of sinusoid-like components called *Intrinsic Mode Functions* (IMF) using the *Empirical Mode Decomposition* (EMD) before further analysis.

Figure 9 shows a signal with one set of possible IMF results produced by EMD.

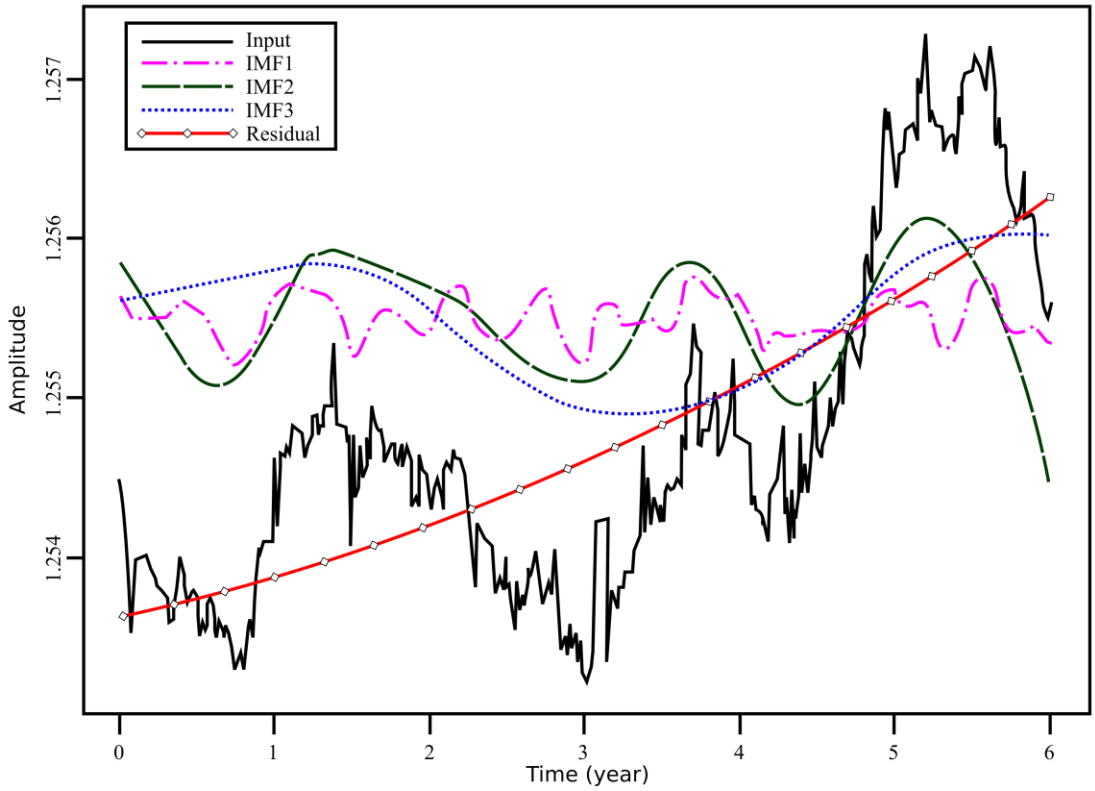

Figure 9 A Signal and One Set of Possible IMF Results Produced by EMD

3.7.2 Intrinsic Mode Functions

An IMF is defined as a function that satisfies the following two requirements:

1. The number of extrema and the number of zero-crossing points must

either be equal or at most differ by one

2. At any point, the mean value of the envelope defined by the local maxima and the envelope defined by the local minima is zero.

By the two requirements, an IMF can be considered as a generalized simple harmonic function, which allows variable amplitude and frequency along the time axis.

3.7.3 Empirical Mode Decomposition

Empirical Mode Decomposition (EMD) decomposes a signal into a finite number of IMFs. The process that extracts an IMF is called *sifting*. EMD consists of a series of *sifting processes*. The i^{th} sifting process produces two resulting components: the desired IMF, IMF_i, and a residual, $r_i(t)$, where $x_i(t) = IMF_i + r_i(t)$, $x_1(t) = x(t)$ and $x_{i+1}(t) = r_i(t) \forall i > 1$. The sifting process continues until the last residual, $r_n(t)$, becomes monotonic or with at most one extrema. Finally we get n IMFs. The input signal can be expressed in this form:

$$\boldsymbol{x(t) = \sum_{i=1}^{n} IMF_i + r_n(t)}$$

Equation 13 Definition of Empirical Mode Decomposition

The IMFs represent different oscillation mode. IMF_1 is the component with the shortest period/the highest frequency. IMF_n is the component with the longest period/the lowest frequency.

Compared to spectrum and cepstrum methods, HHT is more an empirical method rather than a theoretical tool. This is because the 'best' parameter configuration for controlling the sifting process could only be obtained through trial and error. This implies that the performance of the EMD process varies from spoken language to spoken language.

3.7.4 Sifting Process

As mentioned before, the procedure of extracting an IMF from an input signal is called sifting. A sifting process usually consists of several *sifting rounds*. Figure 10 shows a general idea of the envelopes and the local mean of a signal in a sifting round. With regards to the issue of convergence of the sifting process, this is an open question.

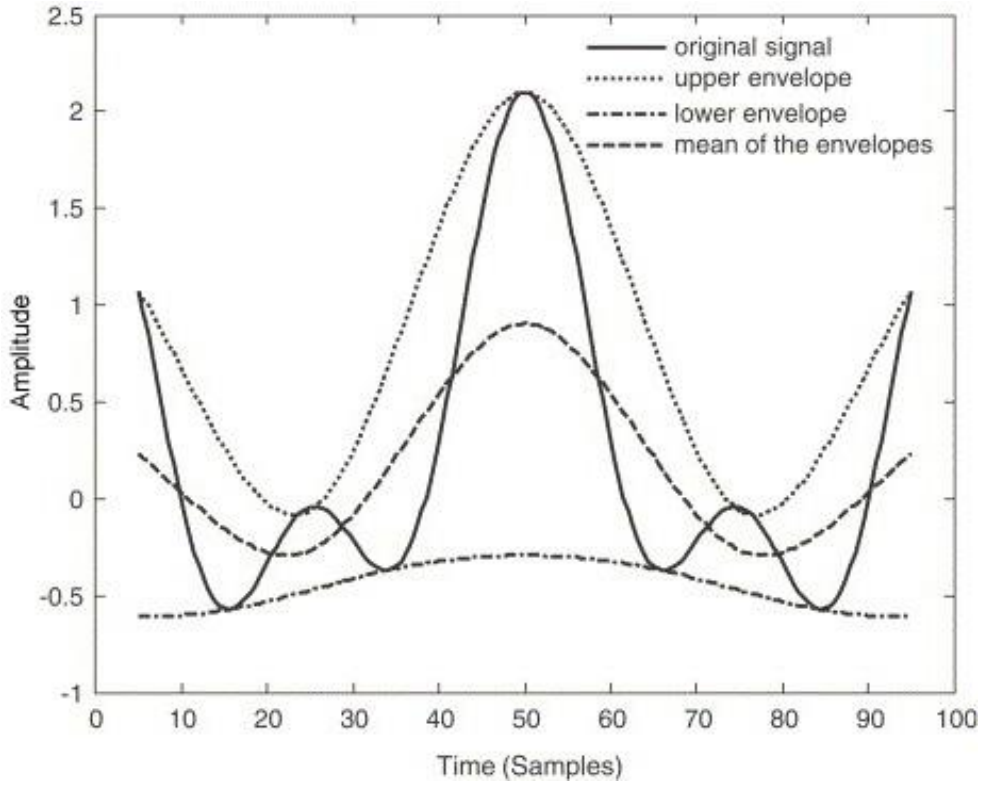

Figure 10 An Illustration of a Signal, its Upper Envelope, Lower Envelope and Local Mean [21]

Listing 1 The Detailed Steps of Sifting Round *j*

1. Find all the local extrema of the input signal $r_{i,j}(t)$
2. Connect all the local maxima by a cubic spline line as the upper envelope, e_{max}
3. Connect all the local minima by a cubic spline to produce the lower envelope, e_{min}
4. Find the local mean (the mean of the upper and lower envelope) of the signal $m_{i,j}(t) = \frac{e_{max}+e_{min}}{2}$
5. Obtain the residual $r_{i,j+1}(t) = r_{i,j}(t) - m_{i,j}(t)$

If the residual in round j, $r_{i,j+1}(t)$, meets one of the stopping criterion described in section 3.7.4.1 then it is an IMF and the sifting process finishes. In this situation, we have $IMF_i = r_{i,j+1}(t)$ and $r_i(t) = r_{i,0}(t) - r_{i,j+1}(t)$.

Otherwise, the sifting process continues to the next sifting round, round $j + 1$, with $r_{i,j+1}(t)$.

3.7.4.1 Stopping Criteria of the Sifting Process

The stopping criterion is one of the most important parts of EMD. It controls how many rounds are needed in one single sifting process. It also greatly affects the accuracy and quantities of IMFs in the final results since every IMF depends upon the residual produced by the previous sifting.

Listing 2 shows some of the possible stopping criteria that are used in traditional EMD sifting processes.

Listing 2 Some Possible Stoppage Criteria

1. The first criterion is proposed by Norden E. Huang [20], the original author of HHT. The sifting process finishes when the sum of difference, $SD_{i,j}$, is smaller than a pre-set threshold α.

   The definition of $SD_{i,j}$ for the sifting round j when sifting IMF_i is:

$$SD_{i,j} = \frac{\sum_t \left| r_{i,j}(t) - r_{i,j+1}(t) \right|^2}{\sum_t \left| r_{i,j}(t) \right|^2}$$

Equation 14 Definition of Sum of Difference for the Sifting Round j when Sifting IMF_i

2. Another stoppage criterion from the same author is to count the number of consecutive sifting rounds that have (i) residuals with equal numbers of zero-crossing points and extrema or (ii) residuals with zero-crossing points and extrema at most differs by one.

   The sifting process finishes when there are more than S consecutive sifting rounds that fulfilled the condition. S is decided empirically.

3. A fixed number of sifting rounds could also be used

3.7.5 Hilbert Spectral Analysis

Hilbert Spectral Analysis (HSA) is a method for examining the instantaneous frequency of an IMF.

By applying Hilbert transform to all the IMFs obtained, we can find the instantaneous frequencies. The input signal $x(t)$ can be expressed as:

$$x(t) = Real \sum_{j=1}^{n} a_j(t) e^{i \int \omega(t) dt}$$

Equation 15 Definition of the Hilbert Spectrum Representation of a Signal $x(t)$

where $a_j(t)e^{i \int \omega(t) dt}$ is the analytic representation of each IMF and $Real$ is a function that obtains the real part of a complex number. This produces the Hilbert spectrum representation of the original signal, which is an energy-time-frequency distribution representation. Figure 11 shows an example of applying EMD and Hilbert transform to an input signal $x(t) = sin(2\pi(35 + 60t)) + sin(2\pi(15 + 20t))$.

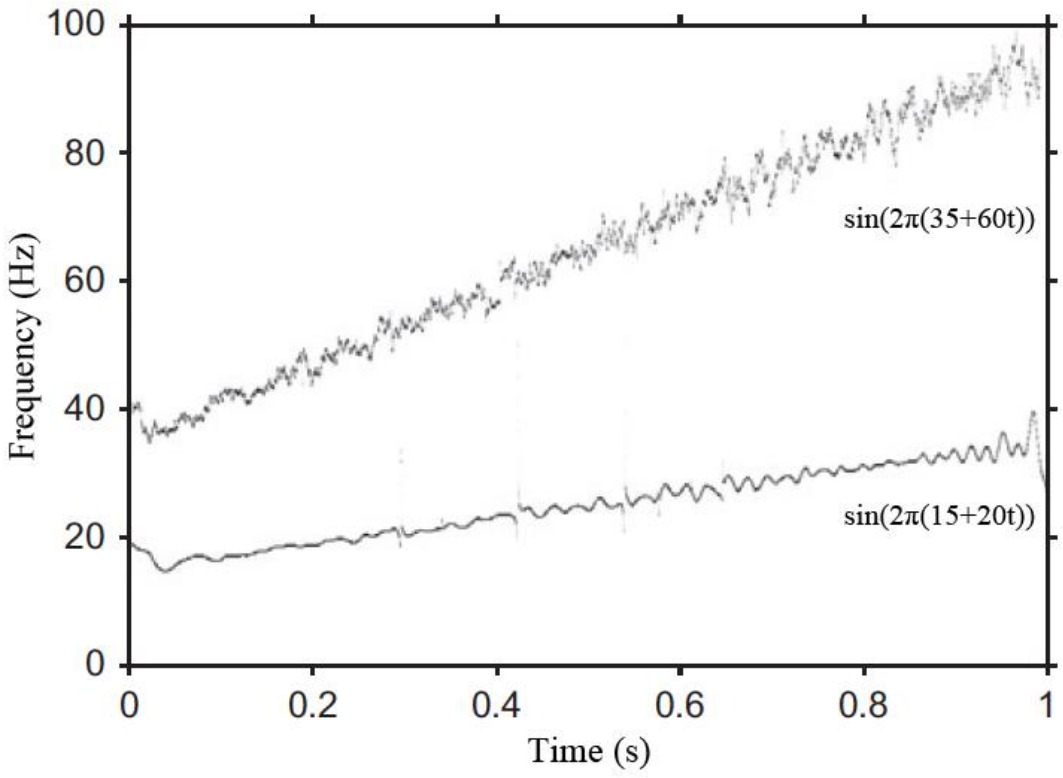

Figure 11 Plot of the Hilbert Spectrum of the Signal $x(t) = sin(2\pi(35 + 60t)) + sin(2\pi(15 + 20t))$ [4]

3.7.6 Mode Mixing Problem and Ensemble Empirical Mode Decomposition

Ensemble Empirical Mode Decomposition (EEMD) is proposed to solve the mode mixing problem which is defined in [22] as:

> *"Mode mixing" is defined as any IMF consisting of oscillations of dramatically disparate scales, often caused by intermittency of the driving mechanisms. When mode mixing occurs, an IMF can cease to have physical meaning by itself, suggesting falsely that there may be different physical processes represented in a mode.*

EEMD is an improved version of EMD. EEMD adds white noise to the input signal before performing EMD. The error introduced by the artificial white noise on the IMFs is cancelled out by taking average on a sufficient number (usually >100) of EMD with different white noise added to the input signal.

Although EEMD can greatly reduce the mode mixing problem in EMD, the computational cost of EEMD is very high when compared to FFT and original EMD. This is because EEMD requires hundreds of EMDs to be performed during the EEMD process, which is not practical for a real-time system.

There is a free MATLAB implementation of EMD/EEMD and instantaneous frequency calculation for HHT available at [23]. At the time of writing there is no hardware for performing HHT available.

3.7.7 Windowed Average Based Empirical Mode Decomposition

In [4], a better local mean estimation method for EMD is proposed. The EMD using the new method is named Window Average Based Empirical Mode Decomposition (WA-BASED EMD). The author tried to reduce the side effect of noise when estimating the local mean, hence eliminating the mode mixing problem found in the original EMD method. This new decomposition method also reduces the computational time. In WA-BASED EMD, the first sifting process sifts out the most noise-like part of the signal as the first IMF in WA-BASED EMD. Since most of the

noise is removed, the subsequent siftings only need to perform simpler interpolation when constructing envelopes. The WA-BASED EMD hence reduces the computational time because interpolation is very time consuming. The disadvantage is that one needs to specify the target frequencies as a parameter, which decreases the automaticity that the original proposed EMD method has. In this thesis, we use HHT with WA-BASED EMD.

3.8 Comparison of Fourier Transform, Wavelet Transform and Hilbert-Huang Transform

In this section, we will compare the three transforms we introduced in previous sections. The basic concepts of these three transforms are similar: breaking down the signal into something simpler and easier for analysis. Zero-crossing and ACF are thus excluded from the comparison since their methodology is very different.

Fourier transform uses sine and cosine of different frequencies as its basic components for decomposition while wavelet transform uses wavelets as its basic components and Hilbert-Huang transform uses IMFs. While sine/cosine and wavelets are pre-defined, IMFs are constructed on-the-fly. We say the basis of FT and WT are *a priori* while that for HHT is *adaptive*.

The frequency spectrum of FT and WT are obtained by convolution (multiplication) of the basic units (sines, cosines, wavelets) with the input signal. While FT applies the convolution to the entire signal, WT applies convolution to different subdivisions of the signal. On the other hand, HHT uses differentiation (subtraction) continuously to obtain the IMFs. Frequencies obtained by FT and WT suffer from the *uncertainty principle* while HHT can obtain instantaneous frequency which does not suffer the same problem.

A frequency spectrum generated by FT produces a 2-dimensional energy versus frequency distribution. A spectrum generated by WT and HHT produces a 3-dimensional energy distribution on the time-frequency plane. STFT can be used to obtain a similar 3-dimensional energy distribution.

FT cannot handle non-linear and non-stationary signals. WT is relatively better in the way that it can handle non-stationary signals but cannot handle non-linear signals. HHT can handle both nonlinear and non-stationary signals.

FT and discrete WT do not have the ability to act as a feature extraction tool while continuous WT and HHT have. Feature extraction, like identifying the gender of speakers, may help tightening the assumption of the fundamental frequency range.

A summary of comparison between Fourier transform, wavelet transform and Hilbert-Huang transform analysis is shown in Table 4.

	FT	**WT**	**HHT**
Basis	*a priori*	*a priori*	adaptive
Frequency	convolution: global	convolution: regional	differentiation: local
Presentation	energy-frequency (FT) energy-time-frequency (STFT)	energy-time-frequency	energy-time-frequency
Nonlinear	No	No	Yes
Non-stationary	No	Yes	Yes
Theoretical base	theory complete	theory complete	empirical

Table 4 Comparison between FT, WT and HHT Analysis

3.9 Comparison and Selection of Different Algorithmic Tools

In this section, we will have a comparison on all the algorithmic tools introduced in this chapter and have a selection of which tools to use in this thesis.

Zero-crossing rate is a very simple and fast method for roughly estimating the fundamental frequency. It is easy to implement and can be used when a precise estimation is not needed. This is used as a comparison reference.

Auto-correlation function is equivalent to convolution of two vectors if the input is a real discrete signal. This applies to our study of Cantonese recordings which are real and discrete signals. Hence ACF is also used in this study as another comparison reference.

Fourier transform is a very important tool used by many studies in signal processing, frequency detection, pitch detections and forensic science. Thus in this study we also implement the STFT using MATLAB. MATLAB has a built-in FFT available, which is fast and accurate (in terms of the calculation error, not the frequency detection accuracy).

Cepstrum is included in this study for the fundamental frequency detection accuracy experiment. One reason for this is that some female voice recordings in our samples were found to have missing fundamental.

Wavelet transform is an excellent tool for decomposing a signal into wavelets which can provide a better understanding of the non-stationary energy trend in the signal. But its poor time resolution in lower scale frequency is a big drawback for Cantonese pitch detection, which requires a high time and frequency resolution. So, wavelet transform is not included in this study.

The target in this study is to find out the performance of the HHT algorithm when applying to the isolated Cantonese syllables for tone recognition. The Hilbert-Huang transform using WA-BASED EMD is implemented and used in this study. Although there are only a few freely available implementations of HHT and none for WA-BASED EMD, their promising power of great time and frequency resolution for helping to examine the signal is worth trying.

3.10 Closing Comments

In this chapter we have looked at four traditional algorithms: the Zero-crossing rate, auto-correlation, FT and cepstrum. Their concepts and technical details are discussed, with a comparison on their strength and weakness on signal processing, speech processing and tone recognition. We have also looked at two relatively modern algorithms, the WT and the HHT. Except for WT, the other five algorithms are selected. They will be used in this study. In the next chapter, we will discuss how these selected algorithms can be used in fundamental frequency detection and pitch tracking. A binary classifier called support vector machine will also be introduced.

CHAPTER 4

APPLICATION OF THE SELECTED ALGORITHMS

4.1 Overview

In this chapter, we look into how the selected algorithms in section 3.9 are used for fundamental frequency tracking in this study. In addition, a binary classification algorithm called Support Vector Machine (SVM) will be discussed also. All the techniques discussed in this chapter are used later, in the experiments described in chapter 6.

4.2 Peak Picking Algorithm

Peak picking is an algorithm for obtaining global/local maxima/minima for a set of data. Figure 12 shows the terms used. For example, peak picking algorithm can be used in a spectrum to obtain the fundamental frequency.

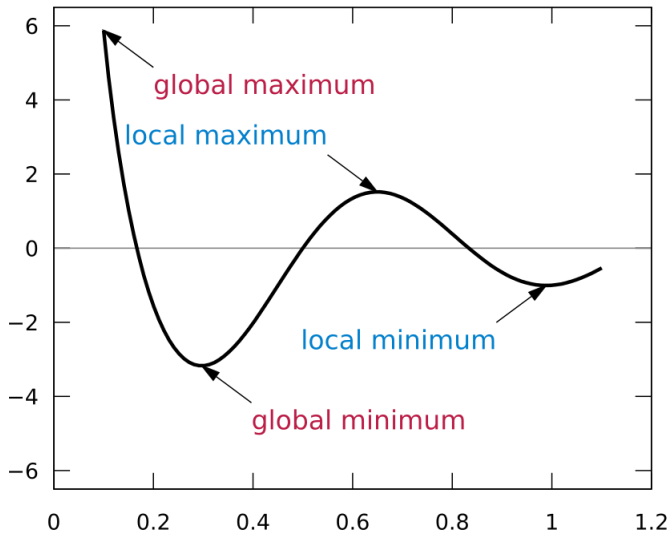

Figure 12 An Illustration of The Global/Local Maxima/Minima of a Signal

For 1-dimensional data like the FFT spectrum, peak picking for global maxima/minima is trivial. For local maxima/minima, a common way is to calculate the first derivative of the input, and find all the zero-crossing points. The downward-

going zero-crossing points correspond to the local maxima and the upward-going zero-crossing points correspond to the local minima.

Peaks produced by noise could be reduced by applying a threshold to the slope of the zero-crossing points, or by applying a threshold to the height difference of the peaks and their neighbors.

4.3 Pitch Tracking Algorithm

Pitch tracking is the process of determining the continuous trend of the fundamental frequency. In speech processing, a voice signal is usually segmented using a moving window before the frequency detection algorithm is applied to it. Peak picking is applied to each segment separately to obtain the fundamental frequency and/or the higher harmonics. The results from each windowed segment combine together to form the pitch track of the input signal.

4.4 Finding the Fundamental Frequency by Counting the Number of Zero-crossing Points

4.4.1 Overview

Counting the zero-crossing points of a signal is one of the simplest methods for estimating the fundamental frequency. By the definition of frequency and the assumption that the waveform of the human voice is *quasi*-triangular, we can find the fundamental frequency of a signal by counting the number of zero-crossing points. Since we expect the fundamental frequency of a voice signal may change over time, the input voice signal $x(t)$ is first segmented into short segments, x_i $(i = 1, 2, \dots, n)$. Each of the segments are the same length. A moving window is used with 80% overlap between 2 successive segments. The fundamental frequency of each segment is then calculated by diving the number of zero-crossing points by the duration of the segment in seconds. A pitch track could then be constructed. Figure 13 shows a flowchart illustrating this process.

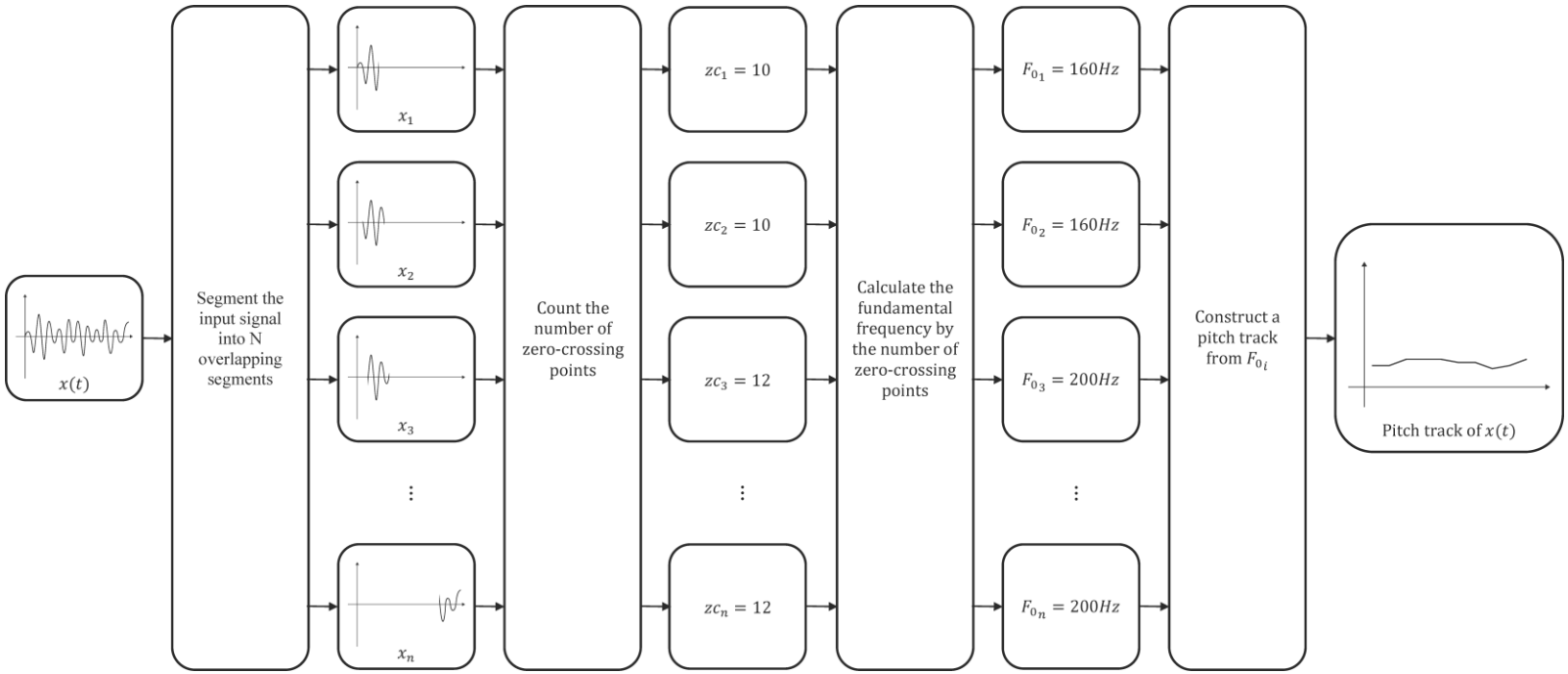

Figure 13 An Illustration of Pitch Tracking by Peak Picking Zero-crossing Rate

4.4.2 Issues Related to Accuracy

This method is not robust when handling noisy signals. For the signal with low signal to noise ratio (SNR), the zero-crossing rate is heavily affected. A noisy signal will generally have a zero-crossing rate much higher than that of a clean signal.

4.5 Finding the Fundamental Frequency by Peak Picking FFT Data

4.5.1 Overview

The spectrum obtained using FFT can be used to find the fundamental frequency of a signal. For an input signal $x(t)$ of length N samples, we first apply a Hamming window and then the FFT to obtain $X(\omega)$. The fundamental frequency F_0 is obtained by finding the frequency ω with the maximum amplitude in the range 80Hz to 350Hz (the range of fundamental frequency of Cantonese by native Cantonese speaker). Figure 14 shows a flowchart illustrating this process.

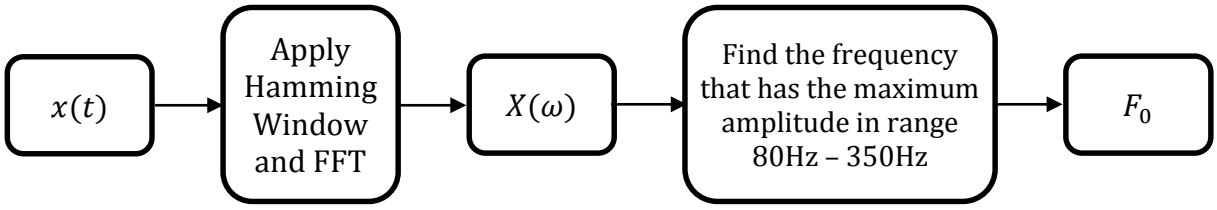

Figure 14 Flowchart of Finding F_0 by Peak Picking of FFT Spectrum

In order to obtain a pitch track of a voice signal, the input voice signal $x(t)$ is first segmented into short segments, x_i $(i = 1, 2, \dots, n)$, of the same length using a moving window with 80% overlap between 2 successive segments. For each segment x_i, a Hamming window and then the FFT are applied to obtain X_i. The fundamental frequency F_{0_i} is obtained by finding the frequency ω with the maximum amplitude in the range 80Hz to 350Hz for every X_i. Then the fundamental frequency track is constructed from F_{0_i}. Figure 15 shows a flowchart illustrating this process.

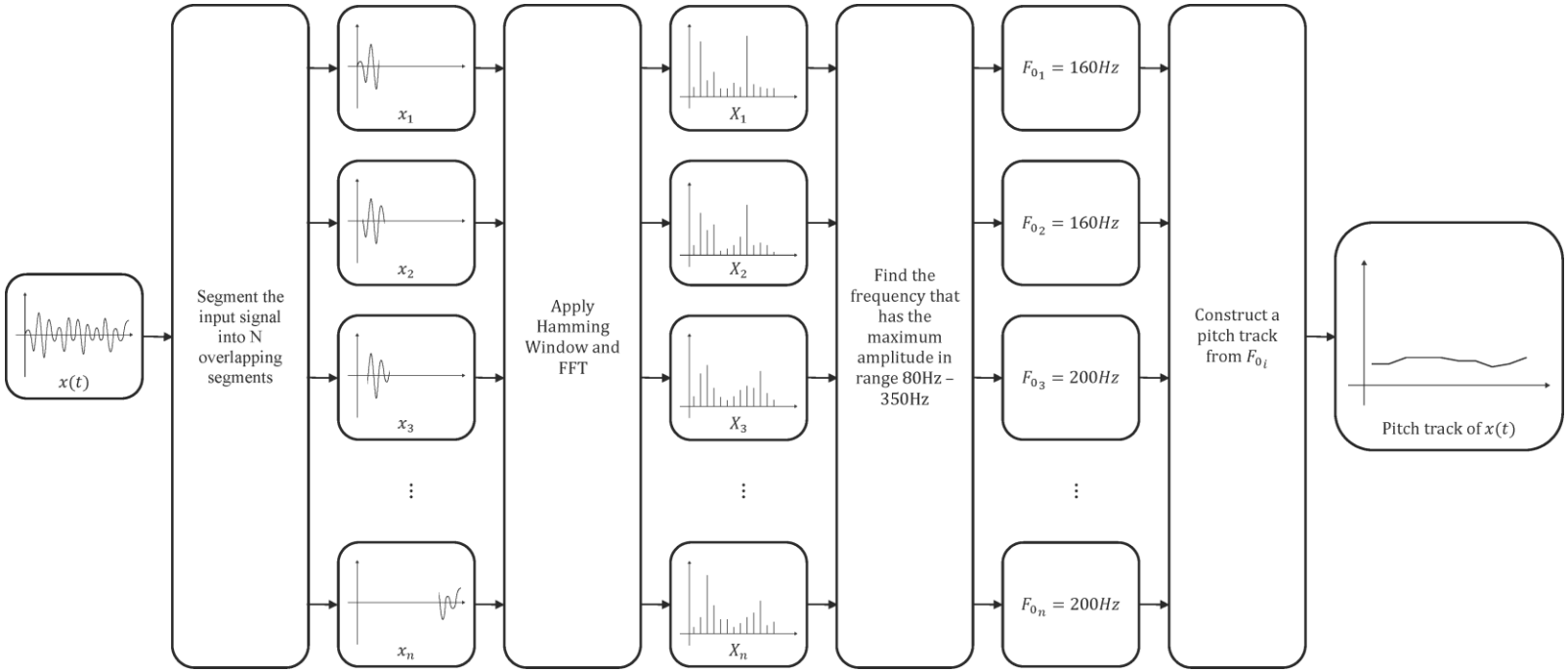

Figure 15 An Illustration of Pitch Tracking by Peak Picking FFT

4.5.2 Issues Related to Accuracy

The accuracy of this algorithm highly depends upon the time and frequency resolution of the FFT process. A windowed segment contains N samples, with sampling rate f_sHz. After Fourier Transform, this give N coefficients. Of these N coefficients, only the first $N/2$ are useful. They represent frequencies from 0 to $f_s/2$Hz. Two consecutive coefficients are spaced apart by f_s/N Hz. In general, the smaller the segment length we have, the larger the separations between two consecutive coefficients of the FFT result. For example, a signal of 4,096 samples with sampling rate 16,000Hz gives a FFT of 4096 coefficients, with two consecutive coefficients spaced apart by $16{,}000/4{,}096 \approx 3.91Hz$. Furthermore, a window of 4096 samples is a segment of length 0.256s for a signal with sampling rate 16,000Hz. In the time domain we can obtain $\left\lfloor \frac{N_x - N}{0.2N} \right\rfloor + 1$ segments from a voice signal $x(t)$ of N_x samples. On average, the duration of a single Cantonese syllable is about 380 milliseconds [8]. The resolution in time domain for this example will therefore be $\left\lfloor \frac{(16000)(0.380) - 4096}{(0.2)(4096)} \right\rfloor + 1 = 3$ segments. In order to increase the time resolution in the time domain, we can shorten the length of the segments, but this will decrease the frequency resolution of the FFT result.

In the case of a missing fundamental, this algorithm may fail to capture a correct value of F_0 due to the fact that it is actually not present or suppressed in the FFT result.

4.6 Finding the Fundamental Frequency by Peak Picking Cepstrum Data

4.6.1 Overview

Similar to the algorithm described in the section 4.5, we could estimate the fundamental frequency of a signal by locating the local peak in the quefrency domain. Figure 16 shows a flowchart of the process.

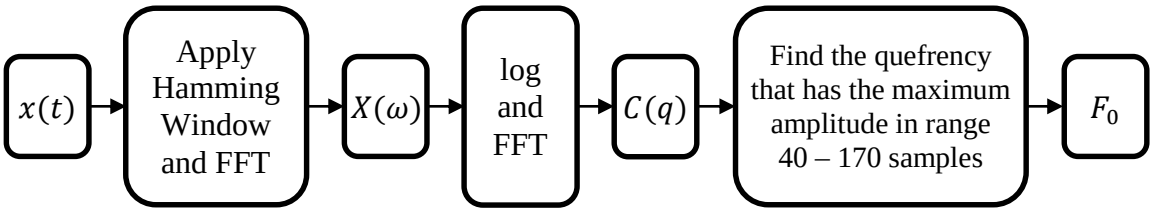

Figure 16 Flowchart of Finding F_0 by Peak Picking Cepstrum Data

To obtain the pitch track using peak picking of cepstrum of a voice signal $x(t)$, a process similar to the algorithm described in section 4.5 is applied. Instead of peak picking of the FFT spectrums X_i, a log function is applied to the magnitude of X_i and then FFT is applied to that to obtain the cepstrum C_i for every voice signal segment x_i. After that, F_{0_i} is obtained by a peak picking process applied to all the C_i. The pitch track is hence constructed. Figure 17 shows a flowchart illustrating this process.

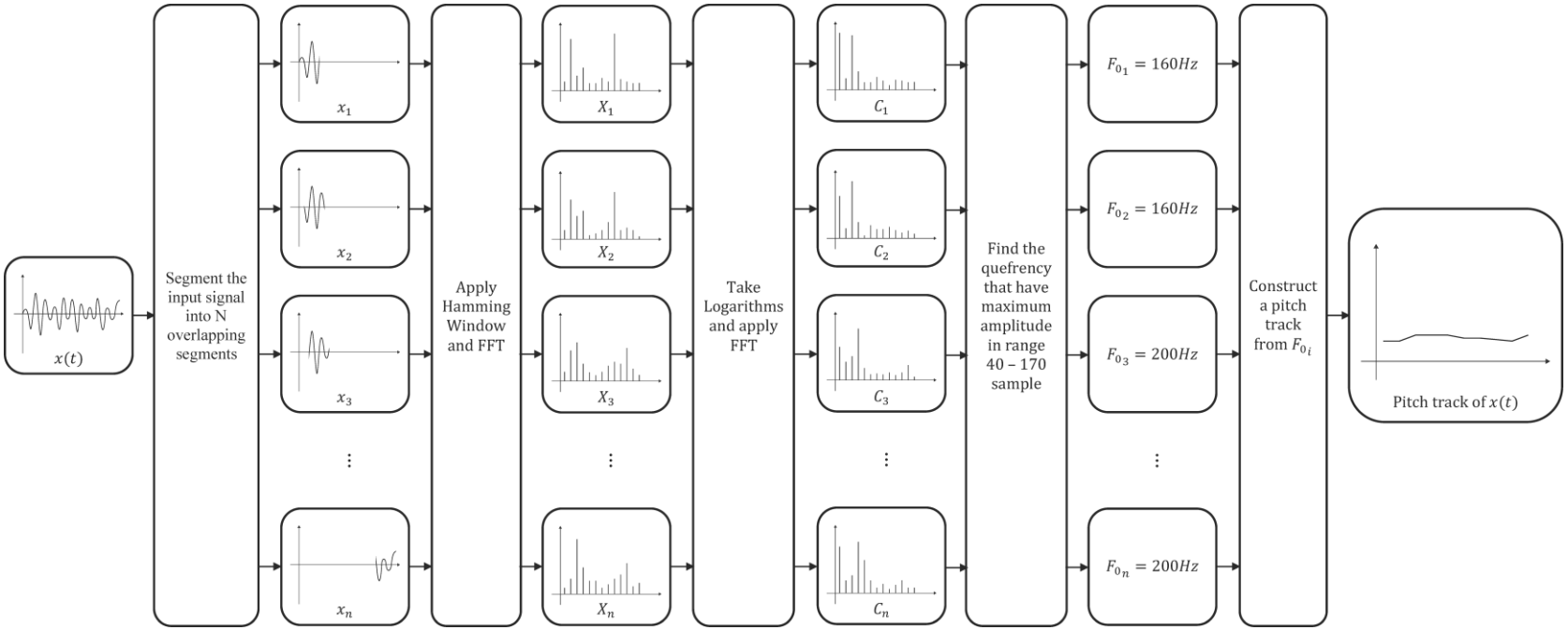

Figure 17 An Illustration of Pitch Tracking by Peak Picking Cepstrum

4.6.2 Issues related to pitch determination

As mentioned in previous section 3.5, the sampling space is only a quarter of the original voice sample size, which limits the bin size of the cepstrum. Added together with the log-scaled quefrency the process of converting quefrency-scaled result back into a frequency-scaled result is non-linear. The conversion is not trivial.

4.6.3 Issues related to accuracy

One of the advantages of cepstrum analysis over spectrum analysis is related to the missing fundamental. As cepstrum indicates the periodicity of a spectrum, it can handle signals that are missing the fundamental frequency. A reasonably accurate quefrency peak in the cepstrum is present for a signal that is missing its fundamental. This is due to the fact that the pattern of separations of higher harmonics provides information that is as important as the separation of the first two harmonics, i.e. the not existing fundamental frequency and the second harmonic.

Despite the ability of the cepstrum method to capture the missing fundamental frequency, the quefrency resolution is very limited. FFT with N coefficients is able to capture frequency components up to maximum of $N/2\,Hz$. A cepstrum could only detect periodicity of the spectrum up to maximum of $N/4\,Hz$.

4.6.4 Issues related to computational cost

The cepstrum algorithm requires two Fourier transforms before we can obtain the pitch track by peak picking. As a result, it requires about double amount of time for handling the same signal, comparing to the spectrum method. This could be a disadvantage if we want the algorithm to be applied on devices that have less computational power such as mobile phones.

4.7 Finding the Fundamental Frequency by HHT

4.7.1 Overview

EMD decomposes a signal into IMFs, which characterize the frequency components composition. To locate the IMF that most accurately represents the fundamental frequency component, we need to identify the IMF that has average frequency in the range 80Hz to 350Hz (the range of fundamental frequency of Cantonese by native Cantonese speaker) and has most of the energy in that range.

The first step is to obtain the IMFs of the input signal $x(t)$ using WA-BASED EMD described in section 3.7.7. Secondly, we count the number of zero crossing points in every IMF and roughly estimate the average frequency. IMFs with average frequency higher than 700Hz (twice the highest native Cantonese speaker fundamental frequency) are discarded. Then the percentage of power of the possible candidates relative to the original signal is calculated. A Hilbert transform is then applied to the candidate with the highest relative power percentage to obtain energy-time-frequency spectrum. The high-energy components form the fundamental frequency pitch track. Figure 18 shows a flowchart of the process.

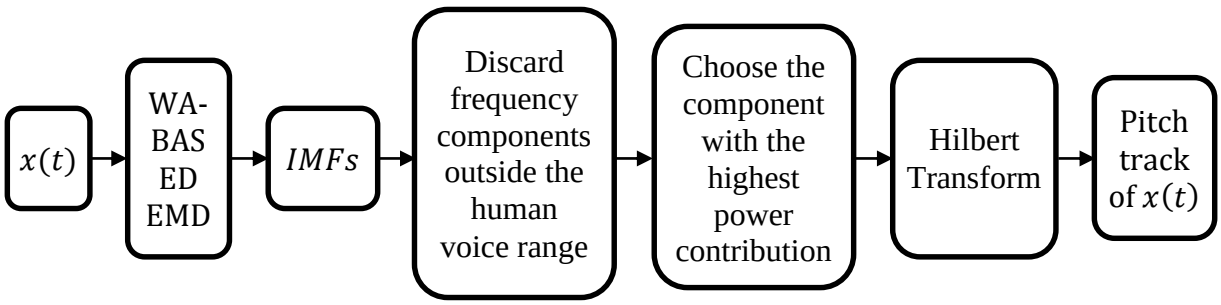

Figure 18 Flowchart of Finding Pitch Track of a Voice Signal by HHT

4.7.2 Issues related to accuracy

The resolution of the frequency track obtained with this method is much higher than those obtained with the spectrum method and the cepstrum method. By the nature of this decomposition, high frequency noise is usually extracted in the

first iteration as the first IMF, which has the highest average frequency. As the result, the sifting process is robust against noisy inputs.

As the EMD is empirical, the possible decompositions of a signal could be more than one depending on the choice of the envelope creation method, local mean estimation method and stoppage criteria. There is no unique decomposition for a signal, hence no best IMF selection algorithm available.

4.8 Support Vector Machine

Support vector machine (SVM) is a tool used in machine learning. SVM is a supervised learning model that was invented by Vladimir N. Vapnik in 1992 [24].

SVM takes a set of training data as input and returns a trained model that can classify new data into 2 classes: 1 or -1. The training set contains a group of n data points. Each data point is viewed as a p-dimensional vector and is labeled with either class 1 or class -1. Formally it is expressed as:

$$\mathfrak{D} = \{(\boldsymbol{x_i}, \boldsymbol{y_i}) | \boldsymbol{x_i} \in \mathbb{R}^{\boldsymbol{p}}, \boldsymbol{y_i} \in \{\mathbf{1}, -\mathbf{1}\}\}_{\boldsymbol{i=1}}^{\boldsymbol{n}}$$

Equation 16 A set of n p-dimensional points with label

By the definition above, the classification problem is transformed into a geometry problem which finds the existence of a (p-1)-dimensional hyperplane that can separate the n points into 2 classes. There may be more than one such hyperplane, among those the best choice is the one with largest margin between the 2 classes. This best hyperplane is called maximum-margin hyperplane, which has its margin maximized to the closest data points in each of the 2 classes.

Figure 19 shows an example of a group of 16 classified data points separated by 3 different hyperplanes. While H_1 failed to separate the points into 2 classes, both H_2 and H_3 separated the 16 points into 2 groups of 8 points as desired. H_3 is the maximum-margin hyperplane, since the distance is maximized to the two points from each class that is nearest to it.

A hyperplane could be expressed as:

$$\vec{w} \cdot \vec{x} - b = 0$$

where $\vec{w}$ is the normal vector of the hyperplane and $\cdot$ is the dot product operation.

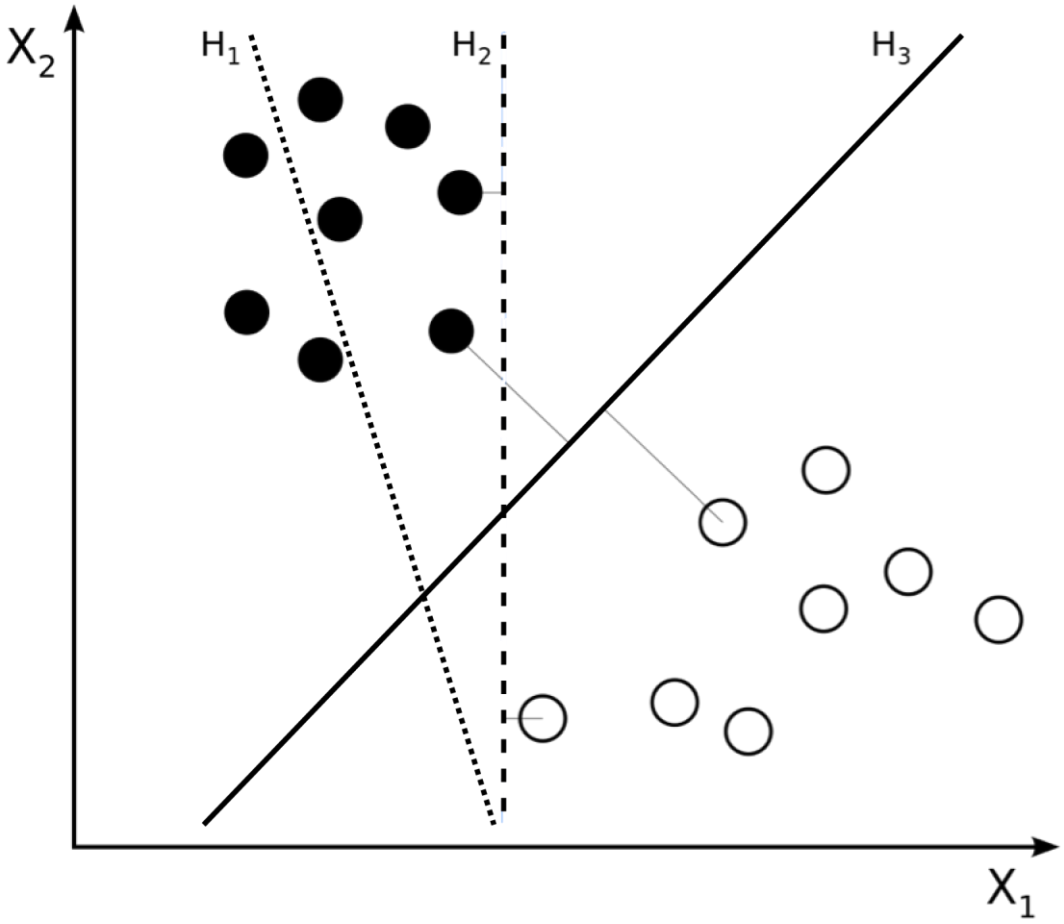

Figure 19 An example of 3 hyperplanes that separate a group of data points into 2 groups

The margins of the maximum-margin hyperplane are two hyperplanes expressed as:

$$\vec{w} \cdot \vec{x} - b = 1 \text{ and } \vec{w} \cdot \vec{x} - b = -1$$

The distance between the two margins is $\frac{2}{\|\vec{w}\|}$. No data points fall between the region (an n-dimensional subspace) bounded by the two margins.

We can form an optimization problem for finding the maximum-margin hyperplane with the constraints: no data points fall between the margins of the maximum-margin hyperplane.

Listing 3 Optimization Problem of Finding the Maximum-margin Hyperplane

Minimize $\|\vec{w}\|$

in $\vec{w} \cdot \vec{x} - b = 0$

subject to $y_i(\vec{w} \cdot x_i - b) \geq 1 \ (i = 1, \ldots, n)$

In order to solve the optimization problem, we transform $\|\vec{w}\|$ into $\frac{1}{2}\|\vec{w}\|^2$ to avoid the square root operation in evaluation of $\|\vec{w}\|$.

The trained model can now classify any new data points with the hyperplane found, by testing the result of $\vec{w} \cdot x - b$ which is the same as the region it falls into.

Then we can solve the problem with quadratic programming optimization. MATLAB has an implementation of SVMs: the functions `svmtrain()` and `svmclassify()`[25]. For the rest of this thesis, unless mentioned explicitly, it is assumed that the SVM model used is from MATLAB.

4.9 Closing Comments

In this chapter we discussed the techniques that will be used later in this study. Pitch tracking algorithm with zero-crossing rate, auto-correlation, FT, Cepstrum and HHT have been discussed. The binary classification tool SVM is introduced. In the next chapter, we will have a literature review of related work for Cantonese tone recognition and Hilbert-Huang transform.

CHAPTER 5

RELATED WORK

5.1 Overview

In this chapter, we will have a look at related work on Cantonese tone recognition and HHT.

5.2 Human Voice Model

One of the most important studies in speech processing is the simplified human voice model first proposed in 1970 by Gunnar Fant [5]. The study examined how the human voice is produced. Fant proposed a simple and concrete approximate mathematic model describing it. The model is called the source-filter voice model. Figure 20 is an illustration of the model. Fant proposed that the human voice is a combination of a slow-changing pulse train with a stable frequency and a filter response imposed by the vocal tract. For research in tone recognition, we are mostly interested in the pulse train. The tone of a syllable is based on the fundamental frequency.

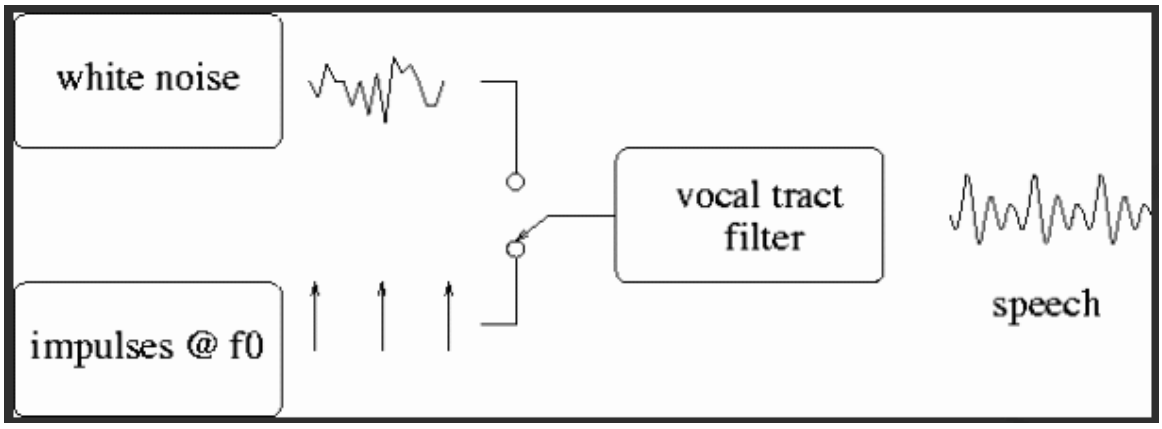

Figure 20 An Illustration of the Source-Filter Model proposed by Fant [26]

5.3 Cantonese Voice Samples

In 1998, the Digital Signal Processing & Speech Technology Laboratory of CUHK has developed a corpus for Cantonese [27]. The corpus is named as CUSYL. The corpus covers 1801 different Cantonese tonal syllables. The data is collected from 2 male and 2 female speakers [28].

5.4 Mandarin Tone Recognition of Isolated Syllables

In the 1990s, research by W.-J, Yang *et al* in tone recognition of isolated Mandarin syllables with Hidden Markov model and vector quantization achieved a speaker-independent recognition rate of 96% accuracy [29]. The high rate of tone recognition of Mandarin syllables is achieved in part because the pitch contours are quite different between the 4 Mandarin tones [30].

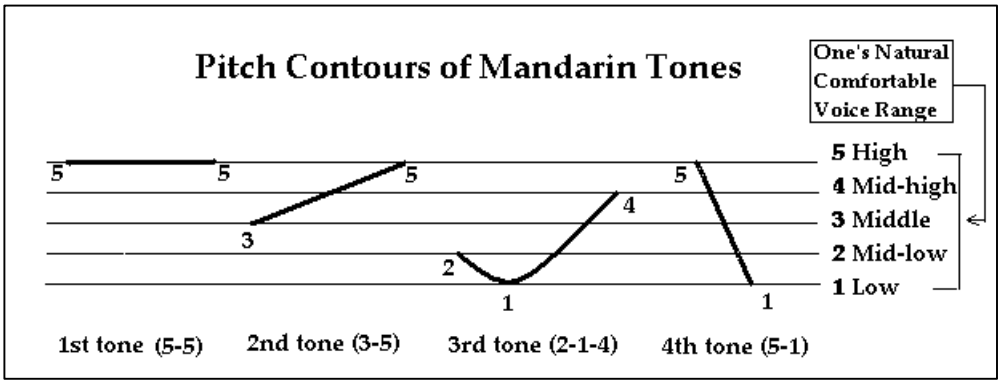

Figure 21 An Illustration of the 4 Mandarin Tones [31]

In 2006, L. Tang improved the speaker-independent recognition rate to 96.64% accuracy for 4 Mandarin tones in Mandarin tone recognition based on pre-classification [32].

5.5 Cantonese Tone Recognition of Isolated Syllables

Unlike Mandarin tone recognition, the accuracy of tone recognition of isolated Cantonese syllables is comparatively much lower. In 'Tone Recognition of Isolated Cantonese Syllables' by Tan Lee in 1995 [33], a 3-layer feed-forward neural network

is used to classify the suprasegmental feature parameters extracted from the voiced portion of a monosyllabic Cantonese utterance. With a training set of 234 syllables, the algorithm showed 87% accuracy for single speaker, speaker-dependent classification of the 9 Cantonese tones. The experiment only covered about 20% of the 1761 Cantonese syllables. The accuracy rate may drop if more syllables were considered.

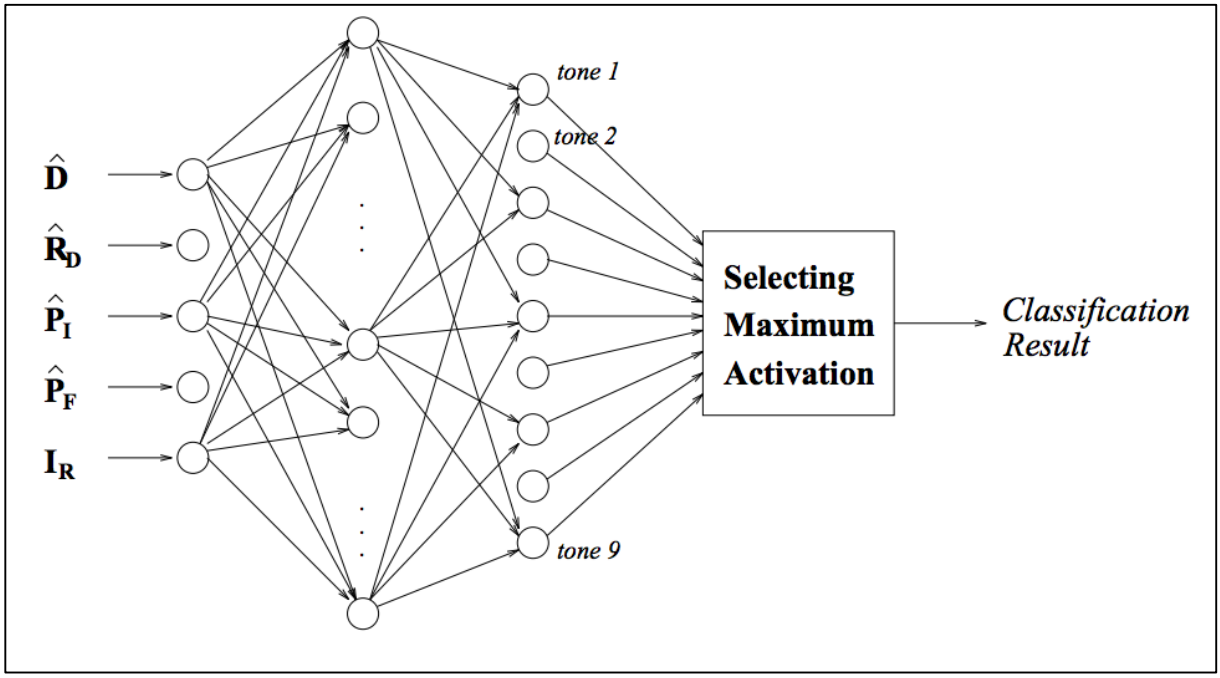

Figure 22 The Multi-layer Perceptron used by Tan Lee. The inputs from top to bottom are: normalized duration, normalized energy drop rate, normalized average pitch of initial, normalized average pitch of final and the pitch rising index respectively.

In 2004, in Tone recognition of continuous Cantonese speech based on support vector machines [34], Gang Peng and William S.Y. Wang proposed an adaptive log-scale 5-level F_0 normalization schema to reduce the tone-irrelevant variation of F_0. They achieved a Cantonese tone recognition rate of 71.5% for tones 1 to 6.

5.6 Cantonese Tone Recognition of Continuous Speech

In 2004, in Tone recognition in continuous Cantonese speech using supratone models, Y. Qian achieved 74.68% accuracy for 6 Cantonese tones speaker-independent recognition [35].

5.7 Hilbert-Huang Transform

The Hilbert-Huang Transform (HHT) was developed by Huang in 1998 [20]. It is an empirical algorithm instead of a theoretical tool. HHT is a general algorithm developed to examine a signal that is non-stationary and nonlinear. It decomposes the signal into so-called intrinsic mode functions (IMF) and obtains the instantaneous frequency. The decompose algorithm, Empirical Mode Decomposition (EMD), works in the time domain and is adaptive, which makes it very efficient.

In 2011, Li *et al.* improved the EMD algorithm and proposed the Windowed Average-based EMD [3]. It mainly improved the mode-mixing issues that occur in the IMFS in the original EMD. By reducing the mode-mixing that occurs in the IMFs, the resulting frequency tracks are more accurate.

5.8 Closing Comments

In this chapter, we have reviewed some related work on Cantonese tone recognition and HHT. In the next chapter, we will go through the main part of this study. The objective, methodology and experiment details and their results will be discussed.

CHAPTER 6

OBJECTIVE, METHODOLOGY AND EXPERIMENTS

6.1 Overview

In this chapter, firstly the objectives are explained in section 6.2. Then the methodology is discussed in section 6.3. After that, in section 6.4 to 6.7, we describe the details of the experiments, their results and analysis.

6.2 Objective

The main objective of this study is to explore the application of Hilbert-Huang Transform to Cantonese tone recognition. We will examine how the Hilbert-Huang Transform performs in fundamental frequency detection. We would also assess the use of the 'best' parameters and procedural details of the WA-BASED EMD of HHT, which may improve its performance on Cantonese voice samples. Lastly we will combine the HHT with the SVM to try to improve the tone recognition performance.

6.3 Methodology

To achieve the objective, a series of four experiments were devised. Here is a quick summary of the four experiments:

Experiment 1. Comparison of the accuracy of different pitch tracking algorithms to determine the best algorithm

Experiment 2. Comparison of accuracy for FFT, EMD and WA-BASED EMD when applied to Cantonese voice signals, to determine the best algorithm

Experiment 3. Varying various parameters for the WA-BASED EMD algorithm to improve its performance on Cantonese

Experiment 4. Assessment of Cantonese tone recognition using HHT and support vector machines

Experiment 1 was devised to test the accuracy of four traditional fundamental frequency detection algorithms when used in pitch tracking. In this experiment, sine, triangle and sawtooth waves with fundamental frequency ranging from 80Hz to 700Hz are used. The range 80Hz to 700Hz is chosen to simulate the typical human voice fundamental frequency range. A synthesized signal is used for testing instead of real voice samples because we want to focus on examining the accuracy of the algorithms with respect to the frequency. After that, the algorithm that performs the best is selected and is used in the second experiment as a comparison base.

The second experiment focuses on testing the accuracy of EMD and WA-BASED EMD compared to the traditional algorithms. The best algorithm in the first experiment will be used as a comparison base. In this experiment, we will examine the accuracy of the three algorithms when they are applied to real Cantonese voice samples.

The third experiment aims to further improve the accuracy of the WA-BASED EMD of HHT. In this experiment, we try varying various parameters and procedural details of WA-BASED EMD with Cantonese voice samples. The set of the ‘best’ parameters that gives the ‘best’ result will be used in the next experiment.

The fourth and final experiment aims to examine the performance of HHT with WA-BASED EMD as a tone recognition tool together with a simple binary classifier. This experiment uses support vector machines as binary classifiers. Random voice samples from the CUSYL corpus are used to trains six SVMs. Each of the SVM classifies one of the six tones. Then, random voice samples from the CUSYL are used to test the accuracy of the SVMs.

In the following sections, we will go through the details of the four experiments.

6.4 Experiment 1 Assessment of Traditional Pitch Tracking Algorithms

6.4.1 Details of Experiment 1

In this experiment, pitch tracking algorithm using four algorithms are assessed: zero-crossing rate, auto-correlation, FFT and Cepstrum. To compare the accuracy of the four different pitch tracking algorithms, 621 sets of simple sine wave, triangle wave and sawtooth wave ranging from 80Hz to 700Hz are generated for input test signals, producing a total of 1863 input signals. The four algorithms are applied to each of the test signals. The wave signals were generated using MATLAB with a sampling rate of 44100Hz. For auto-correlation, FFT and Cepstrum methods, the window size is 2048 samples, and the overlap is 80%. The mean absolute percentage error rates are determined and compared.

6.4.2 Experimental Results of Experiment 1

See Figure 23 for an example result. It shows a 200Hz sine wave with duration of 250ms together with the results of the pitch detection by autocorrelation (AUTO), FFT, Cepstrum (CEPS) and zero-crossing (ZC). REF is the frequency used to generate the signals. Table 5 gives a summary of the performance of the four algorithms for this specific sine wave example.

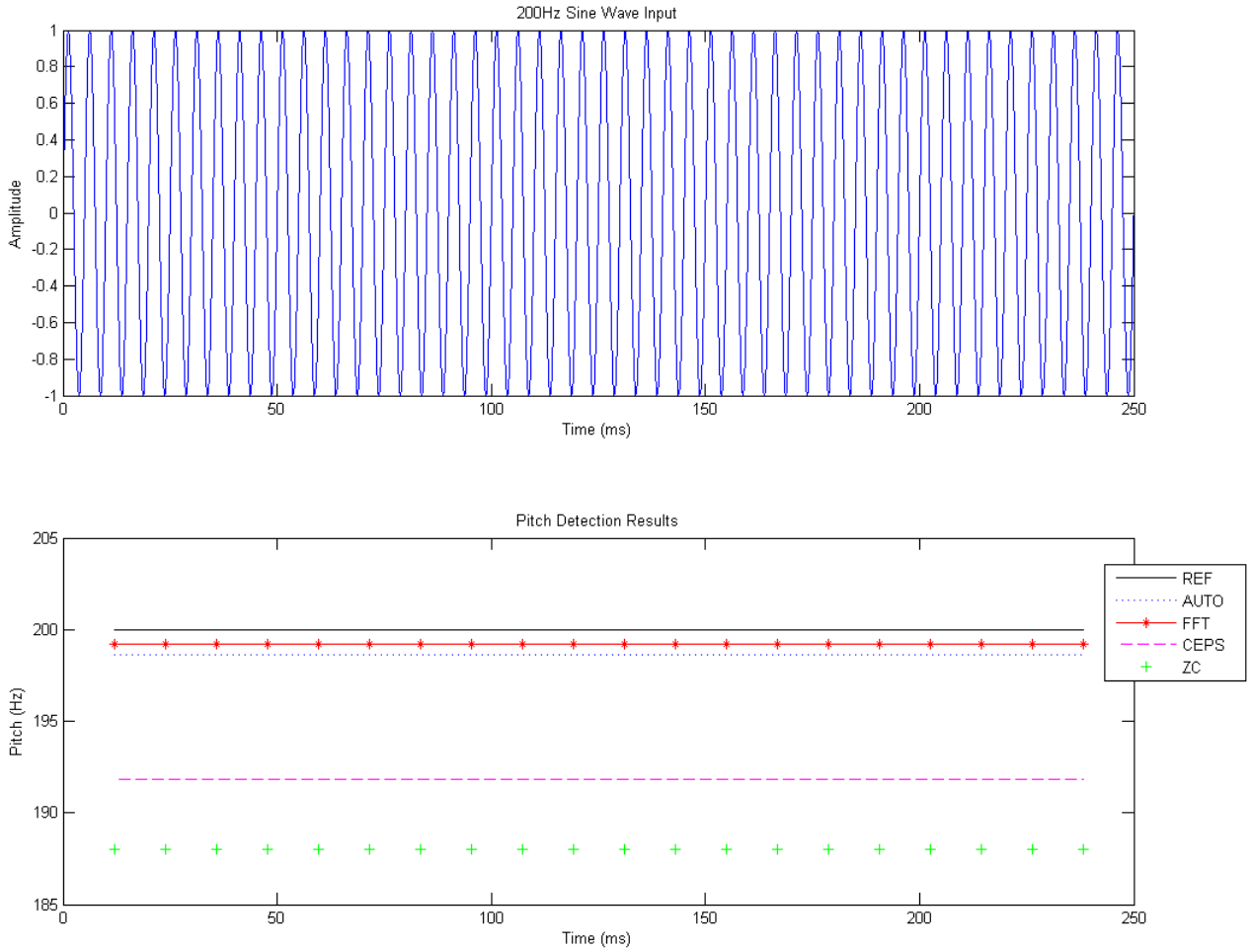

Figure 23 A 200 Hz sine wave in the time domain (upper diagram) and (lower diagram, from top to bottom), the reference (REF), and the pitch detection results using auto-correlation (AUTO), FFT, Cepstrum (CEPS) and zero-crossing (ZC)

Algorithm	Mean of Measured Frequency (Hz)	Mean Absolute Percentage Error (%)
AUTO	198.65	0.675
FFT	199.18	0.41
CPES	191.80	4.1
ZC	188.20	5.9

Table 5 Mean of Measured Frequencies of AUTO, FFT, CEPS and ZC and the Mean Absolute Percentage Error of Each Method. These Results are for a Specific Signal, which is a 200Hz Sine Wave.

Mean absolute percentage error rates of the four methods are shown in Table 6, Table 7 and Table 8 for the sine wave, triangle wave and sawtooth wave sets respectively.

Algorithm	Mean Absolute Percentage Error (%)
AUTO-CORR	1.06
FFT	0.89
Cepstrum	1.43
Zero Crossing	1.24

Table 6 The Mean Absolute Percentage Error of AUTO-CORR, FFT, Cepstrum and Zero-crossing Pitch Detection for 621 Sine Wave Input Signals

Algorithm	Mean Absolute Percentage Error (%)
AUTO-CORR	2.01
FFT	0.85
Cepstrum	1.37
Zero Crossing	1.32

Table 7 The Mean Absolute Percentage Error of AUTO-CORR, FFT, Cepstrum and Zero-crossing Pitch Detection for 621 Triangle Wave Input Signals

Algorithm	Mean Absolute Percentage Error (%)
AUTO-CORR	2.30
FFT	0.88
Cepstrum	1.57
Zero Crossing	1.33

Table 8 The Mean Absolute Percentage Error of AUTO-CORR, FFT, Cepstrum and Zero-crossing Pitch Detection for 621 Sawtooth Wave Input Signals

6.4.3 Conclusion of Experiment 1

From the experimental results, it can be seen that FFT scored the lowest mean absolute percentage error rate in fundamental frequency detection amongst the four algorithms assessed, in all the three categories of waveform tested. As a result, FFT is used as a comparison base for experiment 2.

6.5 Experiment 2 Assessment of Modern Pitch Tracking Algorithms

6.5.1 Details of Experiment 2

The best algorithm from experiment 1, FFT, together with EMD and WA-BASED EMD are further assessed in this experiment. A randomly selected 4 voice samples of each of the 6 Cantonese tones being considered, in total 24 voice samples, are taken from the CUSYL corpus. The ground truth fundamental frequency tracks of the 24 voice samples are firstly measured manually, to provide a comparison reference. The ground truth fundamental frequency tracks are obtained by importing the voice sample into MATLAB, and zooming in close enough so that the period pattern can be manually recognizing. For each period pattern recognized, the period length is measured 'by hand' and the frequency is calculated accordingly.

The three algorithms are applied to the 24 voice samples. For EMD and WA-BASED EMD, the IMF that has a frequency closest to the fundamental frequency of the input is selected for subsequent detailed analysis. The resulting frequency tracks are then compared to the ground truth frequency tracks obtained manually. The mean absolute percentage error rates are determined and compared.

For this experiment, the FFT used a window size of 2048 samples. Using an overlap of 80%, this results in approximately 39 windows per second. The stoppage condition for both EMD and the WA-BASED EMD are a fixed number of 12 iterations. The initial frequency guess needed for WA-BASED EMD is obtained by averaging the 6^{th} to 10^{th} windowed FFT's result of the voice sample. The reason to choose this range of windows is because all of the 24 voice samples have their voiced part starting there, confirmed by manual inspection.

6.5.2 Experimental Results of Experiment 2

Table 9 shows the performance of the three different algorithms, when compared to the manually measured fundamental frequencies of the 24 selected voice samples. The performance is calculated in terms of mean absolute error rate.

Algorithm	Mean Absolute Percentage Error (%)
FFT	1.03
EMD	0.99
WA-BASED EMD	0.76

Table 9 The Mean Absolute Percentage Error of FFT, EMD and WA-BASED EMD for 24 Cantonese Voice Samples

From the experimental results, WA-BASED EMD improved the accuracy of the pitch tracking of the fundamental frequency of Cantonese voice samples by an average of 26%. Similar to the results of English and Mandarin, WA-BASED EMD is relatively more accurate in fundamental frequency detection.

In order to further improve the accuracy of WA-BASED EMD on the Cantonese voice samples, the following study, experiment 3, is carried out for varying the initial conditions and stoppage conditions to determine the 'best' parameters.

6.6 Experiment 3 - Assessment of Various Parameters of WA-BASED EMD

6.6.1 Details of Experiment 3

In this experiment, various parameters of the WA-BASED EMD are assessed. The 24 voices samples used in experiment 2 are used again in this experiment. The WA-BASED EMD algorithm is applied to all of the voice samples with varying parameters. The 'best' set of parameters will be used in a following experiment, experiment 4.

The two components of the WA-BASED EMD that are assessed are:

Part 1. The method for deciding the initial guess of the signal fundamental frequency

Part 2. The stoppage condition for the sifting process mentioned in section 3.7.4

6.6.1.1 Part 1: Varying the method for deciding the initial guess of the signal fundamental frequency for WA-BASED EMD

In order to 'kick start' the WA-BASED EMD, an initial guess of the fundamental frequency is needed. In this experiment, the initial guess is automated by performing peak picking of FFT on the first few windows of the signal to obtain a rough value of the initial fundamental frequency of the whole signal. The choice of the initial guess affects the IMFs obtained.

Three different methods are tried, together with the method used in experiment 2, to obtain the 'best' method of initial guessing of the fundamental frequency needed for the WA-BASED EMD for Cantonese syllables. The guessing methods are:

Guessing Method 1. Use the average fundamental frequency of the 6^{th} to 10^{th} windows of the signal (used in experiment 2). For each window, the fundamental frequency is obtained by peak picking FFT of the signal, with a window size of 2048 samples

Guessing Method 2. Use a fixed value of 300Hz, which is about the middle of the human voice fundamental frequency range

Guessing Method 3. Use the average fundamental frequency of the first 5 windows derived by peak picking FFT of the signal, with a window size of 2048 samples

Guessing Method 4. Use the fundamental frequency of the central 5 windows of the voice sample. The fundamental frequency is obtained by a peak picking FFT of that part of the signal, with a window size of 2048 samples

Guessing method 4 is expected to be the best amongst the 4 proposed methods. This is because in general we believe the middle most part of a voice signal contains the most representative fundamental frequency information.

6.6.1.2 Experimental Results of Part 1 of Experiment 3

The 4 guessing methods described in the previous section are applied to the 24 voice samples and the resulting IMFs are obtained. It was found that the second IMF has a result closest to the actual fundamental frequency. So the second IMF is taken for detailed analysis. The performance of the 4 guessing methods is tabulated in Table 10.

Guessing Method	**Mean Absolute Percentage Error (%)**
Guessing Method 1	0.72
Guessing Method 2	2.95
Guessing Method 3	0.65
Guessing Method 4	0.95

Table 10 Performance of various guessing methods for initial guessing of fundamental frequency for WA-BASED EMD

In section 6.6.1.1 guessing method 4 was predicted to be the one that would give the most accurate resulting IMFs. It turns out this method is only a second runner up amongst the 4 proposed guessing methods. The result of guessing method 2 shows the great impact of the initial guess on the accuracy of the IMFs. Guessing method 3 indicates that, for Cantonese, instead of the frequency of the voiced part, the WA-BASED EMD algorithm works better when using the frequency of the first

part of the signal for the initial guess. As a result, guessing method 3 is chosen as the default approach used in part 2 of experiment 3 (see next section) and experiment 4.

6.6.1.3 Part 2: Varying the stoppage conditions for the sifting process in WA-BASED EMD

As mentioned in section 3.7.4.1, there are many different stoppage conditions that could be used in the sifting process. Different stoppage conditions will be tested and the effect of them on WA-BASED EMD will be studied in this part of the experiment. All the WA-BASED EMD applied in this experiment uses guessing method 3 (i.e., average of the first 5 windows of peak picking FFT) for initial guessing.

The following stoppage conditions are tested:

Stoppage Condition 1. A fixed number of 6 iterations of sifting rounds

Stoppage Condition 2. A fixed number of 12 iterations of sifting rounds (used previously in experiment 2)

Stoppage Condition 3. A fixed number of 24 iterations of sifting rounds

Stoppage Condition 4. The sum of difference threshold method proposed by the original author of HHT, Norden E. Huang [20]

Stoppage Condition 5. 6 consecutive sifting rounds that have (i) residuals with equal numbers of zero-crossing points and extrema or (ii) residuals with zero-crossing points and extrema at most differs by one, has occurred

Stoppage Condition 6. 12 consecutive sifting rounds that have (i) residuals with equal numbers of zero-crossing points and extrema or (ii) residuals with zero-crossing points and extrema at most differs by one, has occurred

The above stoppage methods are used in the WA-BASED EMD for the 24 voice samples and the resulted IMFs are obtained. The second IMF, which has the fundamental frequency closest to the fundamental frequency as mentioned in section

3.7.7, is selected for detailed analysis. Besides the mean absolute percentage error rate, the time used for each sifting process is also recorded.

6.6.1.4 Experimental Results of Part 2 of Experiment 3

The performance of the 6 stoppage conditions is tabulated in Table 11.

Stoppage Condition	Time Used for Sifting	Mean Absolute Percentage Error	Weighted Improvement Measure (col. 3 / col. 2)
Stoppage Condition 1	0.79s (1x)	0.98% (+0%)	0%
Stoppage Condition 2	1.34s (1.70x)	0.77% (+21.4%)	12.59%
Stoppage Condition 3	2.82s (3.57x)	0.78% (+20.4%)	5.71%
Stoppage Condition 4	3.74s (4.73x)	0.80% (+18.4%)	3.89%
Stoppage Condition 5	2.91s (3.68x)	0.79% (+19.4%)	5.27%
Stoppage Condition 6	5.21s (6.59x)	0.73% (+25.5%)	3.87%

Table 11 Performance of various stoppage conditions. (Nx) in the second column shows the amount of time used relative to the quickest result. (+M%) in the third column shows the percentage improvement relative to the worst result. The weighted improvement is the relative percentage improvement divided by the extra amount of time spent.

Amongst the 6 stoppage conditions tested, stoppage condition 1 was the fastest but had the worst accuracy. In contrast, stoppage condition 6 has the best accuracy but it takes more than 6 times more time than stoppage condition 1 to complete the sifting process. In order to compare the performance more accurately, a weighted improvement rate is used. This is shown in the last column of the table. The weighted improvement measure is calculated by dividing the relative percentage improvement by the extra amount of time spent compared to stoppage condition 1 (the fastest but least accurate condition). Stoppage condition 2 has the highest

weighted improvement measure. It clearly outperforms the other methods. As a result, stoppage condition 2 is chosen as the default parameter for the next experiment, experiment 4.

6.6.2 Conclusion of Experiment 3

4 initial guessing methods and 6 stoppage conditions are assessed. From the experimental results described previously, initial guessing using the first 5 windows of the peak picking FFT of the signal is the ‘best’ method. Among the 6 stoppage conditions, a fixed number of 12 iterations in the sifting round is the ‘best’. These two approaches are chosen as the default for WA-BASED EMD for experiment 4.

6.7 Experiment 4 Assessment of Cantonese Tone Recognition with HHT and SVMs

6.7.1 Details of Experiment 4

In this experiment, the method proposed in section 6.3 of using HHT and SVMs for Cantonese tone recognition is assessed. There are 272 syllables with an entering tone in the corpus. For each speaker, we have a total of 1530 voice samples from tone 1 to 6. We used data from 1 speaker as the training set and the other 3 speakers as the testing set.

HHT with the ‘best’ parameters obtained in experiment 3 (section 6.6) are applied to all the voice samples. Pitch tracking algorithm is applied accordingly. Figure 24 shows the averaged energy distributions of the resulting pitch tracks for each tone, with a normalized duration. In the figure, we can clearly see that the energy of the tracks is mainly in the middle third of the entire track. For all pitch tracks, 13 points of interest (POIs) are taken from each of the pitch tracks. The POIs are evenly distributed across the middle third of the pitch tracks, which covers most of the area that representing the useful contours.

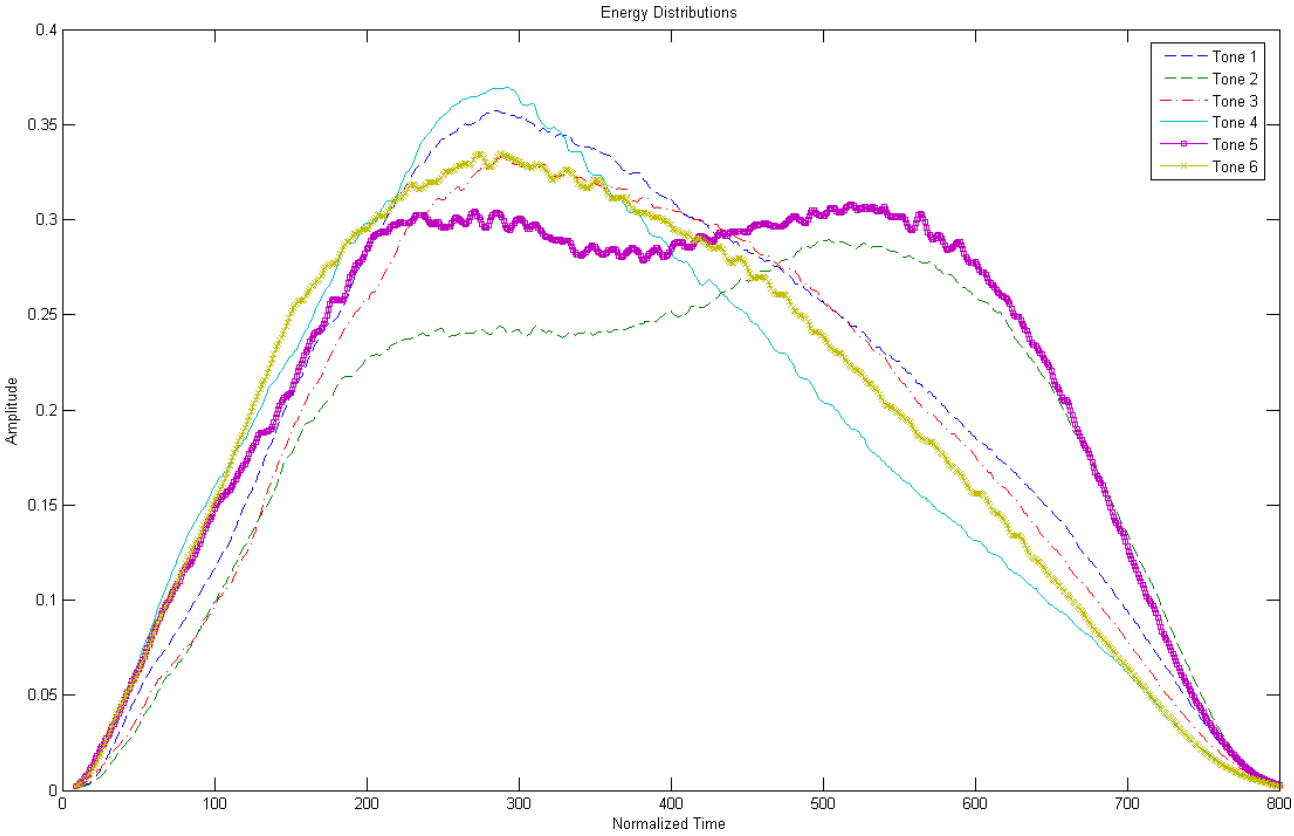

Figure 24 The Averaged Energy Distribution of the 6120 Pitch Tracks, for the 6 Cantonese Tones. The Duration is Normalized to 800 Samples.

The POIs together with their tone information are used as the training input for SVM considered in section 4.8. 6 SVMs are used to perform the training. The training consists of 6 classifications, where the inputs are classified as:

Class 1. Tone 1 and not tone 1

Class 2. Tone 2 and not tone 2

Class 3. Tone 3 and not tone 3

Class 4. Tone 4 and not tone 4

Class 5. Tone 5 and not tone 5

Class 6. Tone 6 and not tone 6

After the training of the 6 SVMs is completed, the 4590 voice samples from the other 3 speakers are used as the testing set.

Four runs of training and testing procedure is processed. Each speaker is used once as the training set. For each subsequent assessment, the speaker used for training is rotated. Classification accuracies of all the 4 runs are then averaged and recognized as the final result.

6.7.1.1 Experimental Results of Experiment 4

The result is shown in Table 12. The column 'Recognized As X' is the averaged number of testing voice samples that are classified by more than 1 SVM.

Tone	**Recognized As**							**Accuracy (%)**
	1	**2**	**3**	**4**	**5**	**6**	**X**	
1	732.25	4.0	164.75	1.0	6.0	70.0	20.0	75.38
2	0.25	782.25	0.5	21.5	129.0	20.5	10.0	81.15
3	90.75	1.0	606.5	1.0	0.5	64.75	19.5	77.36
4	15.5	9.5	19.25	644.25	20.25	17.0	5.25	88.13
5	18.0	20.0	13.75	8.25	339.0	10.5	4.5	81.88
6	49.25	2.0	130.25	2.25	2.75	493.25	19.25	70.57
								79.08 (Averaged)

Table 12 The Averaged Accuracy of the trained SVMs.

The recognition rate of the tone 2, 4 and 5 are comparatively higher than that of the other 3 tones. The most likely reason is that tones 1, 3 and 6 are all level tones, which are easily confused with each other. Tones 2, 4 and 5 have a contour shape which is sufficiently distinct from the other 5 tones, which gives an easier classification task. The average 79.08% accuracy is good but there is a big room for

improvement when compared to the outstanding tone recognition rate of 96% in Mandarin.

6.7.2 Closing Comments

In this chapter, we have defined the objective of this study. The proposed methodology with a series of 4 experiments is described. After that, the details of the 4 experiments and their experiment results are shown. In the next chapter, we will give a final conclusion for this study and suggest possible future directions.

CHAPTER 7

CONCLUSION

In this study, we have investigated the application of the HHT with WA-BASED EMD to Cantonese tone recognition. A series of 4 experiments have been designed and implemented. From experiment 1 and 2 results (discussed in section 6.4.2 and section 6.5.2), the WA-BASED EMD was found to have a higher accuracy for fundamental frequency detection when compared to FFT. It also has a higher time resolution.

From the results of experiment 3 (discussed in section 6.6), we have obtained a set of 'empirically the best' parameters for the Cantonese syllables needed for the WA-BASED EMD algorithm.

In experiment 4 (discussed in section 6.7.1), by using the HHT as a pitch tracking tool together with SVMs as binary classifiers, we have achieved a 79.08% accuracy for speaker-independent tone recognition for Cantonese syllables. Analyzing Cantonese tones is a very hard task due to the high similarity in the contours in the majority of the tones.

A possible improvement is to increase the number of points of interest (POIs) for the training set. By doing so we can increase the level of separation of the POIs in the hyperspace. As a result, one will have a higher chance to find a clearer hyperplane that separates the data. A further possibility could be to try out the optimal number of POIs. Generally speaking, increasing the POIs is an obvious way to increase the SVM's classification performance. However, the higher number of POIs used to train the SVMs, the much longer time is needed for the training of the SVMs to be completed. Also, the performance on the classification when applied in a real system may be impractical. If we use too many POIs, it implies an increase of computational power for classifying the input.

Another possible improvement is to change the way of choosing the POIs amongst the pitch tracks. In our experiment, we used 13 evenly distributed POIs in the middle third of the pitch tracks. This is because we believe the voiced part of the

signal is the main component of the tone contour. Many other possible schema are possible.

Although the accuracy we achieved is not very outstanding, the experiments on HHT applied to Cantonese syllables exhibits a promising improvement in the spectrum for both of the time and frequency resolution. It is definitely useful for other speech processing task, for example the Cantonese syllable recognition and the speaker identification.

BIBLIOGRAPHY AND REFERENCES

[1] Yun-hsuan Sung and Martin Jansche. (2010, December) Research Blog, The latest news from Research at Google. http://googleresearch.blogspot.hk/2010/12/google-launches-cantonese-voice-search.html Accessed on 10 December, 2013

[2] Apple Inc. (2013, September) Apple Inc. http://support.apple.com/kb/HT4992 Accessed on 10 December, 2013

[3] M. Paul Lewis, Ed., "Statistical Summaries", in *Ethnologue: Languages of the World*, Sixteenth ed. Dallas, Texas, USA: SIL International, 2009.

[4] C. Li, X. Wang, Z. Tao, Q. Wang and S. Du, "Extraction of time varying information from noisy signals: An approach based on the empirical mode decomposition", in *Mechanical Systems and Signal Processing*, vol. 25, no. 3, pp. 812-820, 2011.

[5] Gunnar Fant, *Acoustic Theory of Speech Production*, The Hague, 1970.

[6] Robert S. Bauer and Paul K. Benedict, *Modern Cantonese phonology*, Walter de Gruyter, 1997.

[7] Y. Matsuwaki, T. Nakajima, and T. Ookushi, J. Iimura, K. Kunou, M. Nakagawa, M. Shintani, H. Moriyama, T. Ishikawa, "Evaluation of missing fundamental phenomenon in the human auditory cortex", in *Auris Nasus Larynx*, vol. 31, no. 3, pp. 208-211, Sep 2004.

[8] Robert S. Bauer and Paul K. Benedict, *Modern Cantonese phonology*, Walter de Gruyter, 1997.

[9] James W. Cooley and John W. Tukey, "An algorithm for the machine calculation of complex Fourier series", in *Mathematics of Computation*, vol. 19, pp. 297-301, 1965

[10] Mark Borgerding. (2012, July) SourceForge. http://sourceforge.net/projects/kissfft/ Accessed on 1 August, 2013

[11] John Bowman and Malcolm Roberts. (2012, August) SourceForge. http://sourceforge.net/projects/fftwpp/ Accessed on 1 August, 2013

[12] MathWorks. (2012, August) MathWorks. Product Documentation. http://www.mathworks.com/help/techdoc/ref/fft.html Accessed on 1 August, 2013

[13] Atmel Corp. (2012, August) IC-ON-LINE. http://www.ic-on-line.cn/view_download.php?id=1137399&file=0070%5Cat40k-fft_583230.pdf Accessed on 1 August, 2013

[14] Richard Wesley Hamming, "Hamming Window: Raised Cosine with a Platform", in *Digital Filters*, 3rd ed. Englewood Cliffs, N.J., USA: Prentice Hall, 1989.

[15] MathWorks. (2012, August) MATLAB R2012a Product Documentation. http://www.mathworks.com/help/toolbox/signal/ref/hamming.html Accessed on 15 August, 2013

[16] B. P. Bogert, M. J. R. Healy, and J. W. Tukey, "The Quefrency Alanysis of Time Series for Echoes: Cepstrum, Pseudo Autocovariance, Cross-Cepstrum and Saphe Cracking", in *Proceedings of the Symposium on Time Series Analysis*, New York, 1963, pp. 209-243.

[17] D.G. Childers, D.P. Skinner, and R.C. Kemerait, "The cepstrum: A guide to processing" in *Proceedings of the IEEE*, vol. 65, no. 10, pp. 1428-1442, October 1977.

[18] Ivan W. Selesnick, "Wavelet Transform - A Quick Study", in *Physics Today*, October 2007.

[19] Christian U. Grosse, Hans W. Reinhardt, "Signal conditioning in acoustic emission analysis using wavelets", in *NDT.net*, vol. 7, No. 09, Septermer 2002.

[20] Norden E. Huang, Zheng Shen, Steven R. Long, Manli C. Wu, Hsing H. Shih, Quanan Zheng, Nai-Chyuan Yen, Chi Chao Tung and Henry H. Liu, "The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis," in *Proc. R. Soc. Lond. A*, vol. 454, no. 1971, pp. 903-995, 1998.

[21] H.G. Chen, Y.J. Yan, and J.S. Jiang, "Vibration-based damage detection in composite wingbox structures by HHT" in *Mechanical Systems and Signal Processing*, vol. 21, no. 1, pp. 307-321, January 2007.

[22] Z. Wu and Norden E. Huang, "Ensemble empirical mode decomposition: a noise-assisted data analysis method" in *World Scientific*, vol. 1, no. 1, January 2009.

[23] Research Center for Adaptive Data Analysis. (2010, August) Research Center for Adaptive Data Analysis. http://rcada.ncu.edu.tw/research1_clip_program.htm Accessed on 1 July, 2011

[24] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the fifth annual workshop on Computational learning theory*, New York, 1992, pp. 144-152.

[25] The MathWorks, Inc. (2013, Janaury) Documentation Center. http://www.mathworks.com/help/stats/support-vector-machines.html Accessed on 15 August, 2013

[26] Tony Robinson. (2011, August) The source filter model of speech. http://svr-www.eng.cam.ac.uk/~ajr/SA95/node15.html Accessed on 10 December, 2013

[27] Wai Kit Lo, Ka Fai Chow, Tan Lee, and Ching Pak Chung, "Cantonese Databases Developed at CUHK for Speech Processing" in *Proceedings of the Conference on Phonetics of the Languages in China*, pp. 77-80, 1998.

[28] Tan Lee, W.K. Lo, P.C. Ching, and Helen Meng, "Spoken language resources for Cantonese speech processing," *Speech Communication*, vol. 36, no. 3-4, pp. 327-342, March 2002.

[29] W.-J. Yang, J.-C. Lee, Y.-C. Chang and H.-C. Wang, "Hidden Markov model for Mandarin lexical tone recognition", in *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 36, No.7, July 1988.

[30] J. M. Howie, *Acoustical Studies of Mandarin Vowels and Tones*, Cambridge University Press, 2010.

[31] Jin Zhang. (2013, December) Mandarin Tones, Hanyu Pinyin for Mandarin Speakers. http://web.mit.edu/~jinzhang/www/pinyin/tones/ Accessed on 10 December, 2013

[32] L. Tang, "Mandarin Tone Recognition Based on Pre-Classification", in *Intelligent Control and Automation*, vol.2, pp.9468-9472, 2006.

[33] T. Lee, P. C. Ching, L. W. Chan, Y. H. Cheng, and Brian Mak, "Tone Recognition of Isolated Cantonese Syllables," in *IEEE Transactions on Speech and Audio Processing*, vol. 3, pp. 204-209, 1995.

[34] G. Peng, William S.-Y. Wang, "Tone recognition of continuous Cantonese speech based on support vector machines", in *Speech Communication*, vol. 45, pp. 49-62, 2004.

[35] Y. Qian, " Tone recognition in continuous Cantonese speech using supratone models", in *Journal of the Acoustical Society of America*, vol. 121, pp.2936, 2007.