8-17-2018 10:00 AM

Signal Identification In Discrete-Time Based On Internal-Model-Principle

Jie Chen, *The University of Western Ontario*

Supervisor: Lyndon Brown, *The University of Western Ontario*

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

Abstract

This work presents an implementation of a signal identification algorithm which is based on the internal model principle. By using several internal models in feedback with a tuning function, this algorithm can decompose a signal into narrow-band signals and identify the frequencies, amplitudes and relative phases. A desired band-pass filter response can be achieved by selecting appropriate coefficients of the controllers and tuning functions, which can reject the noise and improve the performance. To achieve a result with fast transient characteristics, this system is then modified by adding a low-pass filter. This work is based on the previous work in continuous time. However, a discrete implementation should be much more practical. The simulation result shows a good tracking of the original signal with minimal response to measurement noise.

Keywords: Signal Identification, Internal Model Principle, Frequency Estimation, Band-pass and low-pass filter.

Acknowledgements

First, I would like to thank my supervisor Dr. Lyndon J. Brown for his support and help. I began to do a project with Dr. Brown in my sencond semester. After getting along with him for several weeks, I felt like that he is the best professor I have ever met and a respectful inductor whom I want to learn from. After I began my research, he definately gave me so many guaidances about how to do research and how to learn from the previous work. No matter what questions I asked him, he always explained to me patiently or guided me correctly. I really appreciate everything that he did for me and for my research. He not only provides me the knowlege but also the courgrage to keep doing research.

Last but not the least, I would also like to thank my families for their support both on matter and spirit. Studying abroad alone is not an easy thing at first, it is them who gave me a lot of supports to overcome difficulties.

Table of Contents

List of Tables

List of Figures

List of Abbreviations, Symbols, Nomenclature

HHT	Hilbert Huang Transform
$\bar{K_{1pq}}, \bar{K_{2pq}}$	Feedback controller gains in this work
ϵ_{pq}	Small real number
$\hat{\omega}_p$	Estimated frequency
ω_c	Fudamental frequency
ϕ	Relative phases
d	Signal
G	Tuning Function
K_{1ccpq}, K_{2ccpq}	Continous-time Feedback controller gains in another previous work
K_{1cpq}, K_{2cpq}	Continous-time Feedback controller gains in previous work
K_{1dpq}, K_{2dpq}	Discrete-time Feedback controller gains in previous work
K_a	Tuning Gain
m_p	Number of harmonics

n	Number of fundemental waves
$p\omega_q$	qth $Frequency with p^{th} harmonic$
T_{bpn}	Transfer function of bandpas filter with notches
T_{bp}	Transfer function of bandpas filter
T_{Lpn}	Transfer function of lowpass filter with notches
T_{Lp}	Transfer function of lowpass filter
v	Noise
x_{1pq}, x_{2pq}	State spaces states
AFC	Adaptive feedforward cancellation
ASTFT	Adaptive Short-Time Fourier Transform
DFT	Discrete Fourier Transform
EKF	Extended Kalman Filter
EMD	Empirical Mode Decomposition
FD	Frequency discriminator
FM	Frequency modulated
FT	Fourier Transform
IFD	Instantaneous Fourier Decomposition
IMF	Internal Model Function
IMP	Internal Model Principle
MLSE	Maximum-likelihood sequence estimator

RC	Repetitive control
STFT	Short-Time Fourier Transform
TFR	Time-frequency representation
TPAS	Time-phase amplitude specture
VVS	Virtual variable samplings
WT	Wavelet Transform

Chapter 1

Introduction

1.1 Background

Signals are widely used in our life, such as military, radar, satellite, commercial field and communication. In signal processing field, including communication, the frequency of a sinusoidal wave needs to be identified. When it comes to disturbance cancellation and signal estimation problems, these signals can be modeled as linear combinations of sinusoidal waves, which has become a popular subject since they are predictable and periodic. This issue occurs in musical pitch tracking, computer disk drives and continuous casting of steel.

It is usual to predict signals based on their past values, especially when their characteristics vary slowly. Specifically, in control, communication and mechanical research fields, signals can be represented by:

$$d(k) = \sum_{p=1}^{n} \sum_{q=1}^{m_p} A_{pq} \cos \phi_{pq}(k) + v(k) \tag{1.1}$$

where

$$\phi_{pq}(k) = q \sum_{i=1}^{k} \omega_p(k) + \phi_{pq}(0) \tag{1.2}$$

v is noise and the signal is the sum of n time-varying waves composed of m_p harmonics each. It is assumed that the amplitudes vary slower than the reciprocal of the frequencies $q\omega_p$.

1.2 Motivation

In real life, almost all events involve signals. However, the vast majority of collected signals contain noises or are the mixture of several signals. Due to external environmental interference, it is inevitable that noise is always mixed in signals during signal acquisition and transmission process. And the noise is an important factor affecting the target signal detection and identification performance, especially in some high-precision data analysis. Even a very weak noise may have a tremendous impact on the analysis result. In order to denoise in the signal processing, signal recognition is the first step which also has become a very important discipline. Instantaneous frequency has been a classical issue in signal processing and system controlling fields.

The research of speech recognition technology is a hot topic in the modern era. The speech recognition system has entered into people's life extensively. For example, the speech recognition system of vehicle instrumentation brings great convenience to people. The human-machine voice communication has always been an urgent desire to achieve. After obtaining the voice signal, it is necessary to identify the speech sound out of the mixed signal. A similar problem is encountered in the control field where it is desired to perfectly track reference signals or reject predictable disturbances. Control algorithms that adaptively achieve this goal can be seen to perfectly identify either the disturbance and/or reference signal. So, identifying this signal is the very first and main step of all of these procedures. In this work, these algorithms will be turned from signal processing problem to control problem by using a tuning function to control the process. Our goal is to get the algorithm that can work on audio signals at 20kHz.

1.3 Structure of this Thesis

The organization of this thesis is arranged as follows:

In Chapter 2, the relative literature is reviewed.

In Chapter 3, the previous work is shown.

In Chapter 4, the algorithm is introduced in detail.

In Chapter 5, simulation results verify the feasibility of the algorithm and by comparing it with other methods, the pros and cons of this algorithm are clearly shown.

In Chapter 6, a brief conclusion and the proposing work in the future are illustrated.

1.4 Main Contributions of this Thesis

The main contributions of this thesis are assigned as follows:

First, an improved method for the calculations of the final 4 parameters for the algorithm presented in [32] were developed.

Second, the implementation of the real-time algorithm proposed by Mohsen is successfully converted to discrete time, which is much more easier for a computer to process, and the result shows that it uses only half of the computational power of previous work.

Compared with the related works before, this thesis finds out a way to completely identify signals effectively.

Chapter 2

Review of Literature

2.1 Overview

In this chapter, a literature review of related background of our research is presented in both basic control field and signal processing field.

At present there are many techniques for identifying or canceling a periodic signal. Most of them can be classified as either time-frequency representation-based methods or filter-based methods. The main approaches will be analyzed in this chapter.

Methods of signal identification in signal processing field include: Fourier Transform, which is the most traditional technique, wavelet analysis, Gabor analysis and the approaches that are based on adaptive notch filter and output regulation. Another approach that has been widely used and discussed is Hilbert-Huang Transform (HHT).

When it comes to control field, the same problem can be converted to signal tracking or disturbance rejection. Two of the methods of this problem are repetitive controller and adaptive feedforward cancellation(AFC).

2.2 Control Theory

2.2.1 Internal Model Principle

It is known that Internal Model Principle(IMP) was proposed by Francis and Wonham in 1976 [12]. This principle states that the internal model, which is a dynamic model of the disturbance, is placed in a stable feedback loop to cancel the disturbance or track reference signals. One interpretation is that the linear feedback system will cancel a particular frequency if the controller has the gain of infinity at that frequency. Sinusoidal disturbances have drawn an amount of interest both in estimation and in rejection issues. In this case, the controller must have a pair of poles on the imaginary axis in the s-plane at a location corresponding to the frequency of the disturbance. Internal model is used in order to supply transmission zeros.

In discrete time, the controller's poles at $e^{\pm j\omega_{pq}}$ make the controller's gains at ω_{pq} infinite. In order for the algorithm to be stable, the input at these frequencies must be zero. Further, the gain of the plant needs to be known sufficiently accurate to ensure negative feedback. IMP control suffers significant degradation of performance unless the frequency of the signals is known perfectly. Also, small errors in this model can lead to significant degradation in effectiveness.

2.2.2 Repetitive Control

The basic idea of repetitive control comes from the internal model principle in control theory [15]. Repetitive control is used specifically in dealing with more complex periodic signals than a simple sinusoid. Fourier Series theory states that any periodic signal can be represented by a sum of sinusoids composed of a fundamental signal, *i.e.* a sinusoid with the same period as the periodic signal, and the fundamental harmonics, *i.e.* sinusoids whose frequencies are integer multiples of the fundamental. Thus, in continuous time, a perfect model of any periodic signal with period T would be a model with its poles at $2j\pi/T$. This infinite number of poles can be generated by a feedback loop about a pure delay of T. Typically, in order to stabilize the system, a low-pass filter is incorporated in this feedback loop causing the higher frequency poles, representing lower

energy modes to move to the left of the imaginary axis, and increasing the stability of the closed loop system. In our work, as an alternative, we keep the poles of the controller on the $2j\pi/T$ [15, 48]. The internal model principle is to embed a dynamic model of the system's external signal into the controller, and a mathematical model of the external input signal is included in a stable closed-loop system. This can ensure a high precision of the feedback control system. The repetitive controller is actually a cycle-by-cycle superposition of the input signal. When the input decays to zero, the output still repeats the same signal as what it is during the previous period. If the repetitive controller is placed in the forward channel of the control system and the input error exists, the output of the repetitive controller will increase periodically until the error is completely eliminated, that is, no-error-tracking can be achieved. This method is effective for periodic signal tracking, disk drives [8, 14], satellite control [4], servo hydraulics [47], robotic manipulators [9], etc.

However, due to the existence of the pure delay e^{sT} or z^{-N} in the repetitive controller, its output is delayed by T or N beats with respect to the input. Therefore, in the transient process, the repetitive controller can only respond after delaying N beats. However, it is difficult to stabilize a large number of marginally stable poles and thus a low-pass filter is needed for stabilization. Also, the convergence is very slow and the system may converge to a local minimum.

In [15], a model to generate periodic signal is implemented into the closed-loop system to achieve asymptotic tracking. To stablize the system, a low-pass filter is added as an appropriate proper stable rational function. A synthesis algorithm is developed by using state-space approach and factorization approach. This algorithm can also be implemented in a non-linear system.

In [48], the paper uses repetitive control schemes to track periodic signals with unknown or slowly varying period. Although the signal to identify is in continuous time, the author uses a recursive scheme in discrete time.

In [50], a modified repetitive-control system is used to improve the disturbance rejection performance. The system is composed of two subsystems. The first subsystem is used to reduce the conservativeness of stability condition, the other is to design the gains of the state observer.

In [27], a frequency adaptive discrete Fourier transform(DFT) based repetitive control(RC) scheme for dc/ac converters is presented. Different from the traditional one, this scheme is not that sensitive to frequency fluctuation. In this paper, the virtual variable sampling(VVS) method is used to enable the DFT-based selective harmonic RC to be frequency adaptive. To decrease the off-order harmonic and halve the number of sampling delay in the DFT filter, an off order harmonic DFT filter is used.

2.2.3 Adaptive Feedforward Cancellation

The method using adaptive feedforward cancellation(AFC) was first proposed by M Bodson, A Sacks and P Khosla in [2], and is widely used [30]. Instead of using the control scheme based on the internal model principle, this method is based on the phase-locked loop which is widely used in frequency modulation in communication systems. It can not only reject the certain frequencies but also reduce higher-order harmonics. However, this algorithm is not reliable since it needs to calculate the gain of every frequency, and the number of frequencies to identify is also limited.

Two methods are presented in [1] to reject the sinusoidal disturbances with unknown frequency. In the indirect approach, the frequency is estimated by an adaptive notch filter and then the result is used in a separate algorithm to cancel the disturbance. However, in the direct approach, the estimation and cancellation of frequency is carried out simultaneously. The convergence is ideal for the direct algorithm but not the indirect one, while the indirect one has larger capture region for the parameter estimates. Thus, each one has its own benefit. The direct approach is also used [37] to cancel harmonics.

2.3 Signal Processing

2.3.1 Fourier Transformation

Fourier Transformation(FT) [3] is a traditional signal processing method in pure frequency domain. It uses the original function with the superposition of complex sine components with different frequencies. The Fourier spectrum can be written as:

$$x(t) = \sum_{i=1}^{\infty} a_i e^{j \int \omega_i dt}$$

From the Fourier Transform formula, we know that to get the Fourier transform (spectrum) of a signal, we must take an infinite period of time $(-\infty, +\infty)$, that is, we must get all the information from the time domain. However, if you want to use the spectrum to describe the signal, no matter how short the signal is, you need to describe it in the entire frequency domain. The Fourier transform cannot give you the spectral information in a certain time period $[t_1, t_2]$, which, however, is often of great interest to us. For music signals, we often care about what and when frequency notes are sent out. For seismic signals, we are concerned with what frequencies occur in the spatial locations. These signals are all non-stationary signals. Their characteristics in the frequency domain vary with time. So, the characteristics of the frequency domain of the signal at any moment are very important, and we cannot completely separate the time and frequency domain. That is, the Fourier transform has no locality. It is only applicable to deterministic signals and stationary signals.

Due to the lack of temporal local information, there is a serious shortage of Fourier frequency analysis for time-varying signals and non-stationary signals. It cannot tell us which frequency occurs in which time, and cannot represent the distribution of the signal spectrum at one moment. If the signal changes within a small neighborhood of a certain moment, the entire frequency spectrum of the signal will be affected. However, the change of the frequency spectrum cannot fundamentally determine the temporal position of the change and the severity of the change. The time domain and frequency domain of the Fourier transform are completely separated. In fact, the fundamental

reason is that the trigonometric basis of the Fourier transform is global, and its localized nature is not that good. So, we can only obtain the entire spectrum of the signal, and it is difficult to determine the local characteristics of the signal in any small range in any limited frequency band. Also, this method is quite sensitive to noise [46].

2.3.2 Short-Time Fourier Transform

To obtain the information of when these frequencies occur and the amplitude of the instantaneous frequency at each time, we need time-frequency analysis. Hence, Short-Time Fourier Transform(STFT) is put forward [39]. The signal $x(t)$ is quasi-stationary, and STFT for $x(t)$ is

$$S_x(t,\omega) = \int_{-\infty}^{+\infty} x(\tau)\, h(t-\tau)\, e^{-j\omega\tau} d\tau$$

where $h(t)$ is the analysis window. The window function is used to divide the whole process into small processes. Each of them are approximately stationary. Then, we use FFT to analyze every segment and we can know what and when frequency occurs. However, the width of this window has to be determined in advance. If the window is too wide, the analysis in time domain is not accurate enough; if it is too narrow, the accuracy will decrease in frequency domain. Obviously, the width of the window can vary to improve analysis of non-stationary signal. Unfortunately, the window is fixed in one STFT process. Also, this method suffers from poor time-frequency resolution. Thus, it is not an ideal method of signal identification. The adaptive short-time Fourier transform(ASTFT) was proposed which avoids the worst of the disadvantages of STFT. ASTFT uses an adaptive window to achieve better time-frequency representation(TFR). In addition, a post-ASTFT peak-tracking algorithm further improves the performance by following the continuous ridge in the time–frequency plane and removing the spurious deviations. The algorithm is constructed under a statistical detection and estimation framework and is an approximate maximum-likelihood sequence estimator(MLSE) of IF tracks. It can be implemented efficiently and can process real-time signals with moderately sophisticated hardware. The algorithm also lowers the detection threshold

of frequency-modulated(FM) signals over the STFT and the frequency discriminator(FD) by 4–16 dB at the expense of higher computational and storage cost [25]. In this paper, the author selected a length-N (N odd) unit-energy Hamming window as the base window, and the total cost of the whole Instantaneous frequency estimation(IFE) is

$$N\left(2N + 8\log_2 N + 40\right) + 7FLOPs/output$$

However, the additional cost can be justified in many situations.

In [52], STFT is used to obtain time-phase amplitude spectra(TPAS) which can describe a non-stationary signal. A modified STFT is firstly used to tune the amplitude, initial phase and instantaneous phase distribution of the real signal as functions of time and frequency. Then, TPAS can be calculated by these functions.

2.3.3 Gabor Transform

In order to solve the locality problem, in 1946, Dennis Gabor proposed the concept of "window Fourier transform", namely Gabor transform. Window Fourier transform or Short Time Fourier Transform can be expressed as

$$X\left(t,\omega\right) = \int_{-\infty}^{+\infty} x\left(s\right) g\left(s-t\right) e^{-j\omega s} ds$$

Define the base function as

$$\psi_{t,\omega}\left(s\right) = g\left(s-t\right) e^{j\omega s}$$

when t and ω vary, it constitutes a family which can be considered as a kind of 'basis'. $g\left(s\right)$ represents a window function, such as Hanning, Hamming and Gaussian windows. When we use Gaussian window function, for example:

$$g\left(x\right) = \pi^{-\frac{1}{4}} e^{-\frac{\pi^2}{2}}$$

The transform we get is called Gabor transform:

$$T^{gabor} f(t,\omega) = \pi^{-\frac{1}{4}} \int_{-\infty}^{+\infty} f(s)\, e^{-\frac{(s-1)^2}{2}} e^{-j\omega s} ds$$

The window of Fourier transform or Short Time Fourier Transform(STFT) is the key to achieve the local analysis. The scale of the window is a representation of the degree of locality. When the window function is a Gaussian window, it is generally called a Gabor transform. The reasons for choosing the Gaussian window are:

1) The Fourier transform of the Gaussian function is still a Gaussian function, which makes the inverse Fourier transform become localized with the window function; and at the same time it embodies the localization in the frequency domain;

2) Heisenberg uncertainty principle comes from quantum physics. It says that the position and the velocity of a particle cannot both be identified at the same time. The uncertainty in the postition times the uncertainty of velocity of a particle must be less than Planck's constant. But you can convert position and velocity to potential energy and connectic energy. So the product of uncertainty of potential energy and connectic energy will be less than Planck's constant. Potential energy and connectic energy can be converted to time and frequency respectively. According to the Heisenberg uncertainty principle, the area of the Gaussian function window has reached the lower bound of the uncertainty principle, and it is a function that minimizes the area of the time domain window, *ie*, the Gabor transform is the optimal STFT.

Note that the formula output of STFT has two independent variables, time t and frequency ω, *ie*, it is a time-frequency analysis. However, once the window function is selected, the shape of the time-frequency window remains unchanged, and the inherent relationship between the frequency and the window width is cut off. The Gabor transform essentially is a single resolution analysis. The Gabor transform can achieve the purpose of time-frequency localization: it can provide the whole information of the signal and can provide the information of the intensity of the signal's change in any local time. In short, time domain and frequency domain localized information can

be provided at the same time. Gabor transform has been used in detecting diabetic retinopathy [34], texture segmentation [16], and defect detection [21].

Gabor transform solves the problem of local analysis to a certain extent, but it is still difficult to obtain satisfactory results for catastrophic signals and non-stationary signals. That is to say, Gabor transform still has its serious defects.

1) The size and shape of Gabor transform's time-frequency window remain the same, only the position changes. In practice, it is often desired that the size and shape of the time-frequency window will change with frequency, because the frequency of the signal is inversely proportional to the period. The high frequency part needs a relatively narrow time window to improve the resolution. In the low frequency part, it is hoped that a relatively wide time window can be given to ensure the integrity of the information. In short, an adjustable time window is desired.

2) The basis functions of Gabor transform cannot be orthogonal. For fear of losing information, non-orthogonal redundant bases must be added in signal analysis or numerical calculation, which increases unnecessary calculation and storage. That is, regardless of the study of low-frequency components or high-frequency components, the width and height of the time-frequency window used by the Gabor transform does not change, which is unfavorable for studying higher frequencies or lower frequencies.

As for transient signals, multi-window discrete Gabor transform(M-DGT) is applied in [13]. Due to the balance of the width of analysis window and frequency resolution as well as time resolution, two spectra are combined by geometric average. It shows higher time–frequency resolution than that obtained when only the single analysis window is used in the traditional discrete Gabor transform.

2.3.4 Wavelet Analysis

The concept of "wavelet" was proposed by French geologist J. Morlet in the study of the distribution of subsurface rock oil reservoirs in the 1980s and was successfully applied in geologic data processing [35]; later the mathematician Meyer constructed the first smooth wavelet with certain

attenuation properties [31]. In 1987, Mallat proposed the idea of multi-scale analysis and the Mallat algorithm [29], which unifies the construction of various concrete wavelet functions proposed before this. The wavelet transform inherits and develops the localization idea of Gabor transform, and overcomes some defects of Fourier transform and Gabor transform at the same time. The most important is that the wavelet transform gives an adjustable time-frequency window. The width of the window changes with frequency. When the frequency increases, the width of the time window automatically narrows to improve the resolution. Using wavelet analysis is like using a camera with a zoom lens, and it can show us any detail.

Define signal $f(t) \in L^2(R)$, its Continuous Wavelet Transform (CWT) can be defined as

$$W_f(a,b) = \frac{1}{\sqrt{|a|}} \int_{-\infty}^{+\infty} f(t)\, \Psi\left(\frac{t-b}{a}\right) dt$$

It can be seen that the wavelet transform of the signal $f(t)$ is a binary function. It can also be seen formally that the wavelet transform of the signal $f(t)$ is essentially the weighted average of original $f(t)$ by $\psi_{a,b}(t)$ near $t = b$. This reflected the change speed of $f(t)$ with the standard of $\psi_{a,b}(t)$. In this way, the parameter b represents the time center or time point of the analysis, and the parameter a represents the size of the nearby range centered on t=b. Therefore, the parameter a is generally referred as a scale parameter (*i.e.*, ω in the Gabor transform), and the parameter b is a time center parameter (*i.e.*, t in the Cooper transformation). So, wavelet transform is also a time-frequency analysis.

The advantages of wavelet transform compared to STFT are obvious:

1) Since the wavelet mother function $\psi_{a,b}(t)$ are equivalent to window function, the window width is variable. So, the contradiction between time resolution and frequency resolution are better resolved. Its law of change makes the wavelet transform has excellent localization characteristics, and it is very effective for analyzing catastrophe signals and singular signals. It fully embodies the idea of constant relative bandwidth frequency analysis and adaptive analysis;

2) Wavelet transform can decompose various kinds of interweaved signal into signals of differ-

ent frequencies, and use different step size of the space-time domain corresponding to the size of the frequencies. Any minute details of the object can be continuously focused. This method has great significance in spectrum analysis;

3) It is not required that the wavelet transform base is orthogonal. The product of time width and frequency width is small, and the energy of the spread coefficient is more concentrated.

However, the selection of wavelet bases is a very difficult thing. In some cases, in order to achieve their desired separation effect, the researchers also need to make the appropriate wavelet base according to a particular research. More importantly, this method is very redundant, that is, the data is highly repetitive. But this transformation does not actually get rid of the limitations of Fourier transform, it is a window adjustable Fourier transform, the window of the signal must be smooth. In addition, the wavelet transform is non-adaptive. Once the mother wavelet is selected, in the entire signal analysis process can only use this one wavelet base. This method is suited for a slowly-varying signal.

2.3.5 Hilbert Huang Transform

Traditional Fourier analysis uses a series of trigonometric basis functions to orthogonaly compute signals. However, for non-stationary signals, such as signals with varying frequencies, the Fourier spectrum we obtained is only the average of the frequencies over a certain period of time and cannot accurately describe the change in frequency-time domain. The Hilbert–Huang Transform(HHT) was developed by N.E.Huang. It is a new method of self-adaptive time-frequency domain analyses, which can eliminate human factors and is suited for non-stationary and non-linear analysis. The Hilbert spectrum can be written as $x(t) = \sum_{i=1}^{n} a_i(t)\, e^{j\int \omega_i(t)dt}$. Compared with Fourier transform, it can be noticed that Hilbert transform is the extension of Fourier transform and HHT has more general meanings.

Intrinsic mode functions (IMF) are narrow band components of a signal that represent the individual oscillation modes in the signal. Since most signals are not IMF, a method called Empirical Mode Decomposition (EMD) is used to decompose signals in to IMF. Then,the meaningful instan-

taneous frequency can be obtained after doing Hilbert transform to each IMF. Since instantaneous frequency can only be applied to mono-component signals with a zero mean which does not typically exist in practical signals [19] [18], EMD has the ability to decompose a multi-component signal.

The Hilbert transform gives a mathematically precise way of defining instantaneous frequency. Unfortunately, this definition only agrees with our intuitive understanding when applied to narrow band signals. Thus, it is desired to decompose signals into a sum of narrow band signals before applying the Hilbert Transform. Also, HHT cannot separate frequencies which are closely located. This will become a severe problem in power systems because the frequencies of oscillations typically distribute in a narrow band [24]. Third, the end effects of EMD makes the results on both ends of the data set meaningless. Last but not least, HHT is based on local characteristics which makes it sensitive to not only signal dynamics but also noises [51].

To improve the end effects in EMD, mirroring approach [22], least square polynomial extending [38], constrained cubic spline [23] are proposed. By applying EMD with masking techniques, which is adding a masking signal, the mode mixing can be improved [11][10][42][26]. Using ensemble empirical mode decomposition(EEMD) can also solve this problem [49].

First of all, for steady signals or signals that contain less frequency components, HHT can help us obtain good effect of decomposition and analysis. However, for signals which are non-stationary, more random, and cross-aliasing frequency components, the analysis results are not good enough. Even if HHT is used to denoise, you still cannot tell the noise or useful signals. In conclusion, the basic theory is still immature and needs to be improved and the sieve method does not have a theoretical basis.

HHT has been applied to power quality analysis in [43] and oscillation analysis in power systems [41]. In [44], HHT is also used to analyze terahertz spectra in order to improve the state-of-art detection sensitivity of inconspicuous spectral fingerprints for materials with low concentrations and remote sensing. In [51], a preliminary result of an extended Kalman filter (EKF) method to enhance HHT to over come its shortcomings, including mode mixing and end effects.

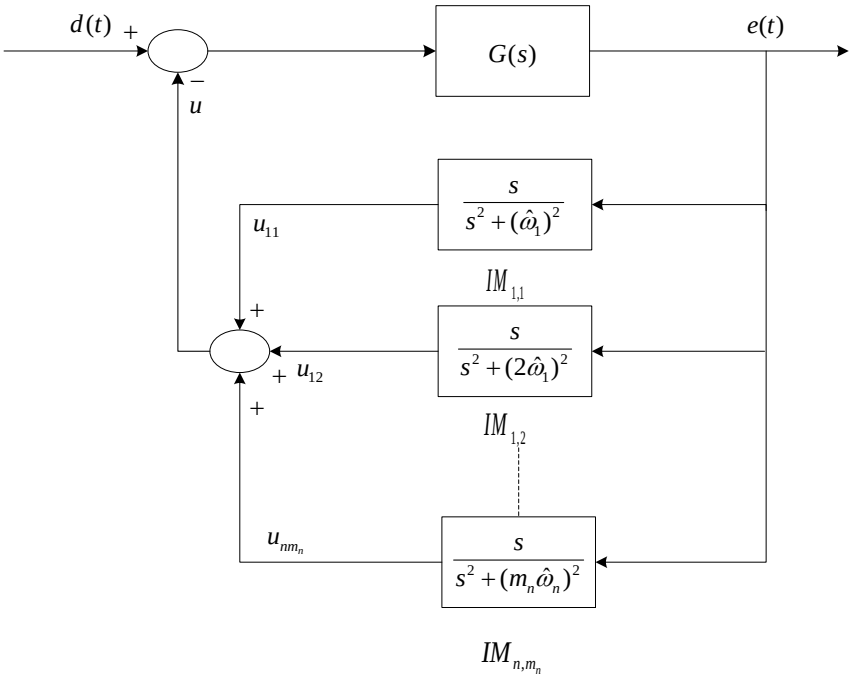

Figure 2.1: Block diagram of original algorithm

Since the IMFs are not orthogonal, the energy leakage is severe. In [20], three orthogonal techniques are used to obtain the completely orthogonal IMFs. The orthogonal IMFs can produce a more faithful representation of earthquake accelerators than the Hilbert spectrum and the Hilbert marginal spectrum.

2.4 Review of the Simple Adaptive Algorithm

The original algorithm is processed in continuous time domain. An internal model of the signal can be presented in state-space form:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & \omega \\ -\omega & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ K_f \end{bmatrix} e$$

$$u = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

The input signal is $d(t) = A\cos(\omega_c t + \phi)$, where ω_c is the true disturbance frequency and ϕ is the phase. Thus, in steady states, if $\omega \sim \omega_c$, we have:

$$e(t) = A_e \sin(\omega_c t + \varphi)$$

$$x_1(t) = A\sin(\omega_c t + \varphi)$$

$$x_2(t) = A\cos(\omega_c t + \varphi)$$

where $A = \frac{K_f A_e}{\omega^2 - \omega_c^2}$, and φ is the phase of e.

The phase diagram of $x_1$and jx_2 would show

$$|x_1(t) + jx_2(t)| = \sqrt{x_1^2(t) + x_2^2(t)} = |A|$$

$$\theta = \angle(x_1(t) + jx_2(t)) = \omega_c t + \varphi$$

The frequency ω_c can be identified by the differentiating:

$$\frac{d\theta}{dt} = \omega_c$$

In steady state, the phase plot of $x_1$and x_2 is a circle as shown in Fig. 2.2. Define $\bar{\theta} = \angle(x_1(t) + jx_2(t))$, to get ω_c, certainty equivalence principle suggests to replace the ω_c in the right hand with ω. Thus, the estimated frequency can be derived as:

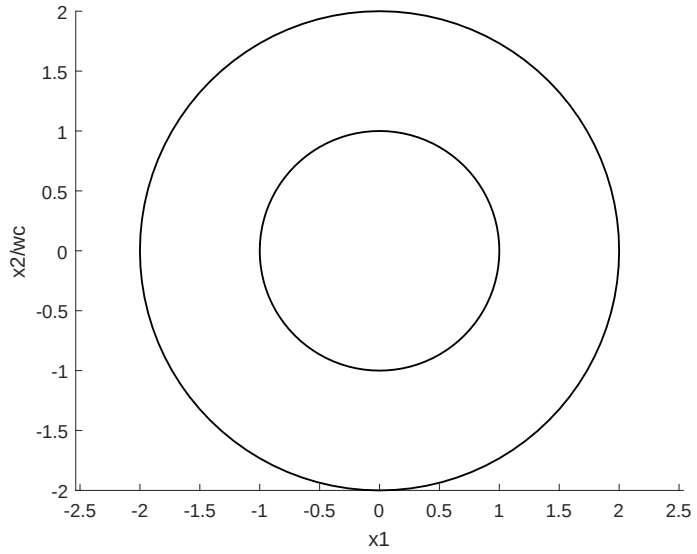

Figure 2.2: The phase diagram with two different magnitudes

$$\begin{aligned} \omega_c &= \frac{d\bar{\theta}}{dt} \\ &= \frac{\omega\left(x_2^2+\omega^2 x_1^2-K_f e x_1\right)}{x_2^2+\omega^2 x_1^2} \\ &= \omega - \frac{K_f e x_1}{x_2^2+x_1^2} \end{aligned}$$

Up to here, we showed how the rotating speed of the circle in Fig. 2.2 can be used to solve the unknown frequency.

2.5 Conclusion

Comparing wavelet transform, Gabor transform, and Fourier transform, we find that the Fourier transform has no locality; the Gabor transform has locality, but has some shortcomings (as described above); and the wavelet transform not only has locality, but also the scale parameter a can change the shape of the spectrum structure and the window, and plays the role of "zoom". So the wavelet analysis may achieve the effect of multi-resolution analysis. From the theoretical development process of the signal analysis method, it can be seen that the Fourier analysis is particularly suitable for the analysis of relatively stable signals over a long period of time. The STFT has its

own application, but its effect depends on the proper selection of the window function. The wavelet analysis is particularly suitable for analysis of mutation signals and singular signals.

Chapter 3

Previous Work

3.1 Internal Model

Brown and Zhang [6, 7] presented an algorithm for identification of periodic signals with uncertain frequencies. This approach is based on the internal model principle of control theory. An adaptive internal model of a sinusoid is incorporated in a feedback loop to achieve the goal for disturbance rejection and frequency estimation. The stability of the algorithm is proved in [7] by employing the singular perturbation theory and averaging theory. The motivation of this algorithm is from the realization that the actual frequency error can be inferred from the states of an internal model and the feedback error by using a nonlinear mapping function. This adaptation method is very similar to Hsu's [17] update law but the normalization now comes inherently instead of as a modification to Regalia's algorithm [40]. This algorithm has already been applied on several applications, such as musical pitch tracking[57], audio signal decomposition [28], power systems [56], sound and vibration control [5] and dynamic resistance measurement in spot welding [45].

In [5], although excellent disturbance rejection has been achieved, the system is stable only when the disturbance frequency varies in a small range. This will cause a problem for systems having a large phase variation (exceeding 180) over the range of the frequencies of interest, This problem can be alleviated by using control gains that are a function of the identified frequencies.

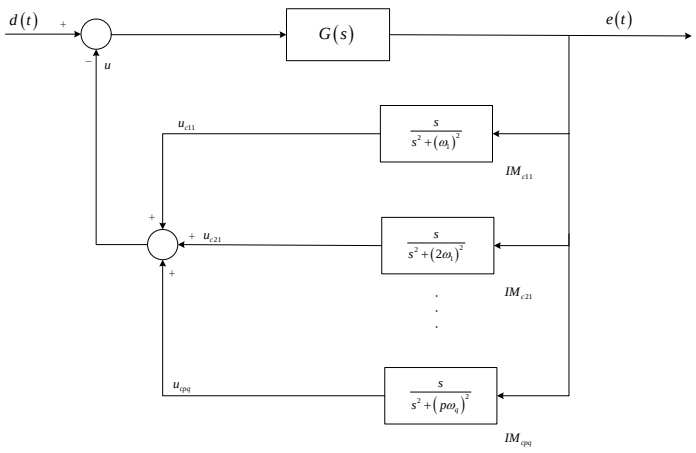

Figure 3.1: Simple tuning instantaneous Fourier decomposition block diagram

Otherwise, stability can only be achieved by inputting signals with certain frequency, which is not possible for systems that have large phase and frequency variations. Thus, it is necessary to tune the control gains so that the system remains stable.

By using instantaneous Fourier Decomposition(IFD), Y.Sun [45] has successfully implemented the internal model adaptive algorithm on two practical applications: an acoustic duct system to improve stability performance, and RSW process to estimate the dynamic resistance. This work has been extended by Y.Ma [28]. After the frequencies are known, this thesis[55] shows how the dynamic of the system can be completely specified.

This work follows the approach in [33] and find a feasible approach to a similar problem in discrete time domain.

3.1.1 Previous work in continuous time

The structure of the simply tuned instantaneous Fourier decomposition algorithm is shown in Fig. 3.1. The system will produce zero error when the model frequency is equal to the signal frequency, i.e. $q\hat{\omega}_p = q\omega_p$. Then, each u_{cpq} will be a single sinusoidal and meet the HHT definition of an intrinsic mode function.

Here, $G(s) = 1$. $\bar{A}_{cpq}$, $q\omega_p$ and φ_{pq} are the uncertain magnitudes, frequencies, and relative phase respectively. Every IM_{pq} is the internal model of a sinusoid which has a frequency of $q\omega_p$.

According to the internal model principle of control theory, to cancel a sinusoidal disturbance, a model of this disturbance needs to be created in the feedback loop. The simplest internal model of a sinusoidal disturbance has the transfer function of $K_f s/\left(s^2+\omega^2\right)$. One realization of this transfer function is:

$$\dot{X}_{cpq} = A_{cpq}X_{cpq} + B_{cpq}e \tag{3.1}$$

$$u_{cpq} = C_{cpq}X_{cpq} \tag{3.2}$$

where the state is $X_{cpq} = \begin{bmatrix} x_{1cpq} & x_{2cpq} \end{bmatrix}^T$. And A_{cpq}, B_{cpq} and C_{cpq} can be expressed as

$$A_{cpq} = \begin{bmatrix} 0 & q\omega_p \\ -q\omega_p & 0 \end{bmatrix} \quad B_{cpq} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad C_{cpq} = \begin{bmatrix} K_{1cpq} & K_{2cpq} \end{bmatrix}$$

The transfer function in continuous time is:

$$T_{cpq}(s) = \frac{K_{2cpq}s + K_{1cpq}q\omega_p}{s^2 + \left(q\omega_p\right)^2}$$

In the equation 3.1 and 3.2, if $e = 0$ and the initial condition of x_1 and x_2 are $x_1(t_0)$ and $x_2(t_0)$ respectively, the general expression of x_1 and x_2 are:

$$\begin{aligned} x_1(t) &= \cos(\omega t)\cdot x_1(t_0) + \sin(\omega t)\cdot x_2(t_0) \\ &= |x_1(t_0) + jx_2(t_0)|\sin(\omega t + \varphi) \end{aligned}$$

$$\begin{aligned} x_2(t) &= -\sin(\omega t)\cdot x_1(t_0) + \cos(\omega t)\cdot x_2(t_0) \\ &= |x_1(t_0) + jx_2(t_0)|\cos(\omega t + \varphi) \end{aligned}$$

where $\varphi = \arctan\frac{x_1(t_0)}{x_2(t_0)}$.

We have $x_1^2(t) + x_2^2(t) = x_1^2(t_0) + x_2^2(t_0) = constant$. So, the phase diagram for $x_1(t)$ and $x_2(t)$ is a circle with the radius of $\sqrt{x_1^2(t_0) + x_2^2(t_0)}$. Also, because $\sin^2(\omega) + \cos^2(\omega) = 1$, we have

$$|x_1(t_0) + jx_2(t_0)| = |x_1(t) + jx_2(t)| = constant$$

So,

$$\theta = \arctan \frac{x_1(t)}{x_2(t)} = \arctan \frac{\sin(\omega t + \varphi)}{\cos(\omega t + \varphi)} = \omega t + \varphi$$

and

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

Therefore, we can use the states x_1 and x_2 to obtain the frequency ω.

Now, in the general case where $q\hat{\omega}_p$ and e are not zero, the responses at $x_{1cpq}(t)$ and $x_{2cpq}(t)$ in steady state are:

$$x_{1cpq}(t) = \frac{\omega}{\omega_c}\bar{A}_{cpq}\sin(q\omega_p t + \phi_{cpq}) \tag{3.3}$$

$$x_{2cpq}(t) = \bar{A}_{cpq}\cos(q\omega_p t + \phi_{cpq}) \tag{3.4}$$

Here, we have:

$$\begin{aligned} \bar{A}_{cpq} &= \sqrt{K_{1cpq}^2 x_{1cpq}^2(t) + K_{2cpq}^2 x_{2cpq}^2(t)} \\ \phi_{cpq} &= \angle\left(x_{1cpq}(0) + jx_{2cpq}(0)\right) \end{aligned} \tag{3.5}$$

let $q = 1$,

$$q\omega_p \sim \frac{d}{dt}\theta(t) = \frac{\dot{x}_{1cpq}x_{2cpq} - x_{1cpq}\dot{x}_{2cpq}}{x_{1cpq}^2 + x_{2cpq}^2}$$

$$= \frac{\left(q\hat{\omega}_p x_{2cpq} + 0\right)x_{2cpq} - x_{1cpq}\left(-q\hat{\omega}_p x_{1cpq} + e\right)}{x_{1cpq}^2 + x_{2cpq}^2}$$

$$q\omega_p - q\hat{\omega}_p = \frac{-e \cdot x_{1cpq}}{x_{1cpq}^2 + x_{2cpq}^2} \tag{3.6}$$

Thus, the difference $\sim q\omega_p$ between fundamental and estimated frequency, $q\omega_p$ and $q\hat{\omega}_{cp}$ can be presented as

$$\triangle q\omega_p = q\omega_p - q\hat{\omega}_p \approx \frac{ex_{1cpq}}{x_{1cpq}^2 + x_{2cpq}^2} \tag{3.7}$$

Using simple integrator controller with gain K_{ca} to update the frequency estimate we have

$$\frac{d\hat{\omega}}{dt} = K_{ca}\triangle q\omega_p \approx K_{ca}\frac{ex_{1cpq}}{x_{1cpq}^2 + x_{2cpq}^2} \tag{3.8}$$

With $G(s) = 1$, this structure has the benefit that the closed loop system is stable whatever n, m_n and $q\hat{\omega}_p$ are; however, the speed of the closed loop dynamics can vary greatly with the $q\hat{\omega}_p$ and the dynamics of the system is uncontrollable and uncertain which can increase amplification of measurement noise.

3.1.2 Simple model in discrete time

The zero order hold equivalent discrete time realization of the previous model is

$$X_{dpq}(k+1) = A_{dpq}X(k) + B_{dpq}e(k)$$

$$u_{dpq}(k) = C_{dpq}X_{dpq}(k)$$

where for simplicity sake we will drop the time dependecy k from this point on, we have $X_{dpq}(k) = \begin{bmatrix} x_{1dpq}(k) & x_{2dpq}(k) \end{bmatrix}^T$. And A_{dpq}, B_{dpq} and C_{dpq} can be expressed as

$$A_{dpq} = \begin{bmatrix} \cos q\omega_p & \sin q\omega_p \\ -\sin q\omega_p & \cos q\omega_p \end{bmatrix} \quad B_{dpq} = \begin{bmatrix} \frac{1-\cos q\omega_p}{q\omega_p} \\ \frac{\sin q\omega_p}{q\omega_p} \end{bmatrix} \quad C_{dpq} = \begin{bmatrix} K_{1dpq} & K_{2dpq} \end{bmatrix}$$

The point wise in time transfer function in discrete time is:

$$T_d(z) = \frac{\left(K_{1dpq} - K_{1dpq}\cos q\omega_p + K_{2dpq}\sin q\omega_p\right)z + K_{1dpq} - K_{1dpq}\cos q\omega_p - K_{2dpq}\sin q\omega_p}{q\omega_p\left(z^2 - 2z\cos q\omega_p + 1\right)}$$

where the sample period is $T = 1$.

3.1.3 Alternative model in continuous time

The previous models used in the original work are not the best representations if one wishes to make the gains of the system time varying. Instantaneous changes in the Ks result in potentially large instantaneous changes in u. It would be better if the system behaved like a "bumpless" time varying system, *i.e.* step changes in the gains result in continuous values of u(t). This can be achieved by placing the gains in the input matrix B. Mohsen reparameterized the controllers as follows[32]. The state-space in previous work in continuous time is:

$$\dot{X}_{ccpq} = A_{ccpq}X_{ccpq} + B_{ccpq}e$$

$$u_{ccpq} = C_{ccpq}X_{ccpq}$$

where the state is $X_{ccpq} = \begin{bmatrix} x_{1ccpq} & x_{2ccpq} \end{bmatrix}^T$. And A_{ccpq}, B_{ccpq} and C_{ccpq} can be expressed as

$$A_{ccpq} = \begin{bmatrix} 0 & q\omega_p \\ -q\omega_p & 0 \end{bmatrix} \quad B_{ccpq} = \begin{bmatrix} K_{1ccpq} \\ K_{2ccpq} \end{bmatrix} \quad C_{ccpq} = \begin{bmatrix} 0 & 1 \end{bmatrix}$$

The transfer function in continuous time is:

$$T_{ccpq}(s) = \frac{K_{2ccpq}s + K_{1ccpq}q\omega_p}{s^2 + \left(q\omega_p\right)^2}$$

Here, we have the formula:

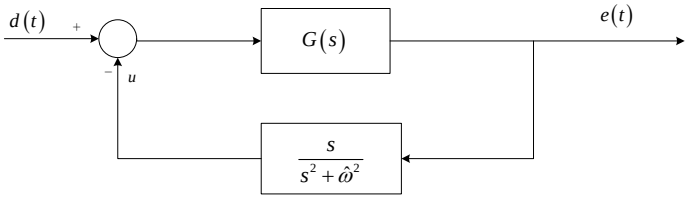

Figure 3.2: Block diagram of continuous system with one internal model

$$\theta(t) = \angle\left(x_{1ccpq} + jx_{2ccpq}\right)$$

$$q\hat{\omega}_p = \frac{d}{dt}\theta(t) = \frac{\dot{x}_{1ccpq}x_{2ccpq} - x_{1ccpq}\dot{x}_{2ccpq}}{x_{1ccpq}^2 + x_{2ccpq}^2}$$

$$= \frac{\left(q\omega_p x_{2ccpq} + K_{1ccpq}e\right)x_{2ccpq} - x_{1ccpq}\left(-q\omega_p x_{1ccpq} + K_{2ccpq}e\right)}{x_{1ccpq}^2 + x_{2ccpq}^2}$$

$$q\hat{\omega}_p = q\omega_p + \frac{K_{1ccpq}x_{2ccpq} - K_{2ccpq}x_{1ccpq}}{x_{1ccpq}^2 + x_{2ccpq}^2}e \tag{3.9}$$

$$q\tilde{\omega}_p = \frac{K_{1ccpq}x_{2ccpq} - K_{2ccpq}x_{1ccpq}}{x_{1ccpq}^2 + x_{2ccpq}^2}$$

3.2 System with One Internal Model (Off-Line Tuning)

3.2.1 Continuous time

In [55], we know that by designing the **closed loop system** to be equal to band-pass filter with notches, we can achieve required stability in our system. The transfer function of a 2nd order band-pass filter is

$$T_{bp}(s) = \frac{B \cdot s^2}{s^4 + C_1 s^3 + C_2 s^2 + C_3 s + C_4} \tag{3.10}$$

We add one notch in this case.

Coefficients of band-pass filter

C_1	C_2	C_3	C_4	B
200.0208	$3.8974 * 10^4$	$2.3690 * 10^5$	$1.4027 * 10^6$	$3.2624 * 10^4$

Table 3.1: Coefficients of band-pass filter (continuous time)

Denominator coefficients of band-pass filter and model

	T_{de}	T_{bpn}
s^0	$a_4\omega^2$	$C_4\omega^2$
s^1	$a_3\omega^2$	$C_3\omega^2 + 2\varepsilon\omega C_4$
s^2	$a_4 + a_2\omega^2 + K_1 b$	$C_2\omega^2 + 2\varepsilon\omega C_3 + C_4$
s^3	$a_3 + a_1\omega^2 + K_2 b$	$C_1\omega^2 + 2\varepsilon\omega C_2 + C_3$
s^4	$a_2 + \omega^2$	$\omega^2 + C_1 \cdot 2\varepsilon\omega + C_2$
s^5	a_1	$2\varepsilon\omega + C_1$
s^6	1	1

Table 3.2: Denominator coefficients of band-pass filter and model (continuous time)

$$T_n(s) = \frac{s^2 + \omega^2}{s^2 + 2\varepsilon\omega s + \omega^2} \tag{3.11}$$

The overall transfer function of Fig. 3.2 is

$$T_{de}(s) = \frac{G(s)}{1 + G(s) \cdot \left(\frac{K_2 s + K_1}{s^2 + \omega^2}\right)} \tag{3.12}$$

where

$$G(s) = \frac{b \cdot s^2}{s^4 + a_1 s^3 + a_2 s^2 + a_3 s + a_4} \tag{3.13}$$

After expanding the denominators of T_{de} and $G(s)$, we can get every coefficients of the polynomial as the Table. 3.2 below.

We are designing a fourth order Chebyshev Type I band-pass filter which passes frequencies between 2π and 60π with 1 dB of ripple in the passband. The coefficients of the denominator of band-pass filter is shown as Table. 3.1.

Here, the signal we use has the frequency of $4.2 * 2\pi$ rad/s. By matching the coefficients, the coefficients of T_{de} is shown as in Table. 3.3.

Coefficients of model

a_1	$2\varepsilon\omega + C_1$	205.2987
a_2	$C_1 \cdot 2\varepsilon\omega + C_2$	$4.0029 * 10^4$
a_3	$\left(C_3\omega^2 + 2\varepsilon\omega C_4\right)/\omega^2$	$2.4753 * 10^5$
a_4	C_4	$1.4027 * 10^6$
K_1	$\left(C_2\omega^2 + 2\varepsilon\omega C_3 + C_4 - a_4 - a_2\omega^2\right)/b$	15.7897
K_2	$\left(C_1\omega^2 + 2\varepsilon\omega C_2 + C_3 - a_3 - a_1\omega^2\right)/b$	5.8666
b	B	$3.2624 * 10^4$

Table 3.3: Coefficients of model (continuous time)

3.3 System with More than One Internal Model (Online tuning)

3.3.1 Continuous time

This section is also written in [32]. Here, a 2nd order band-pass filter is given in Eq. 3.10. And the notches are expressed as

$$T_n = \prod \frac{s^2 + (q\hat{\omega}_p)^2}{s^2 + 2\epsilon_{pq} q\hat{\omega}_p s + (q\hat{\omega}_p)^2} \tag{3.14}$$

The diagram of the model is shown in Fig. 3.3. The block $f(X_{11})$ is the adaptation function shown in Eq. 3.8.

Thus, the desired closed loop has transfer function is

$$T_{bpn} = \frac{d_1 s^2}{s^4 + c_1 s^3 + c_2 s^2 + c_3 s + c_4} \times \prod \frac{s^2 + (q\hat{\omega}_p)^2}{s^2 + 2\epsilon_{pq} q\hat{\omega}_p s + (q\hat{\omega}_p)^2} \tag{3.15}$$

The transfer function of the model in Fig. 3.3 is

$$\begin{aligned} T_{de=} & \frac{G(s)}{1 + G(s)\sum_{p=1}^{n}\sum_{q=1}^{m_i}\left(\frac{K_{2pq}s + K_{1pq}q\hat{\omega}_p}{s^2 + (q\hat{\omega}_p)^2}\right)} \\ = & \frac{b_1 s^2 \prod (s^2 + (q\hat{\omega}_p)^2)}{a(s)\prod (s^2 + (q\hat{\omega}_p)^2) + b_1 s^2 \sum (K_{2kl}s + K_{1kl}l\hat{\omega}_k)\gamma_{kl}(s)} \end{aligned} \tag{3.16}$$

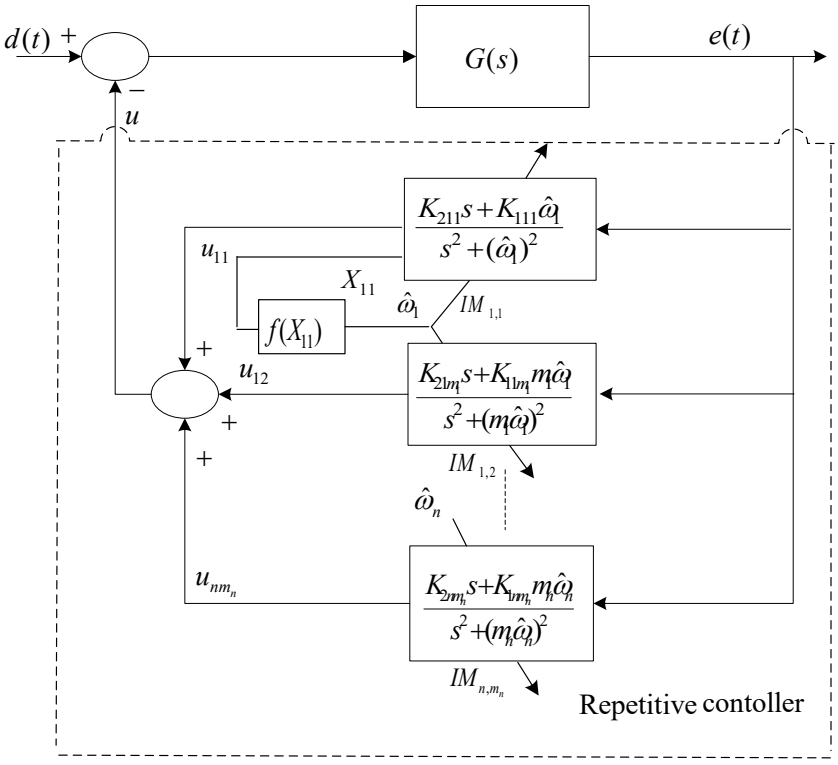

Figure 3.3: Block diagram of continuous system with more than one internal model

where

$$\gamma_{kl} = \prod_{p=1}^{n_i} \prod_{\substack{q=1 \\ \{p \neq l \text{ if } q = k\}}}^{m_i} (s^2 + (q\hat{\omega}_p)^2) \tag{3.17}$$

Note $\prod$ in all equations represents $\prod_{p=1}^{n} \prod_{q=1}^{m_i}$ and $\sum$ represents $\sum_{k=1}^{n} \sum_{l=1}^{m_k}$. The terms γ_{kl} are the product of all the terms $s^2 + (q\hat{\omega}_p)^2$ except the $q = k,\ p = l$ term.

By matching the numerators, b_1 can be easily calculated, $b_1 = d_1$. Matching the denominators generates $2n_t + 4$ coupled equations. This generates $2n_t + 4$ unknowns where $n_t = \sum_{i=1}^{n} m_i$. which is too complex for a computer to solve efficiently. Now, we will show a less computationally intensive algorithm to calculate these unknowns.

The a is can be calculated by substituting 4 other arbitrary frequencies into the denominator

setting up 4 equations with 4 unknowns. One characteristic of T_{de} is that the first term of the denominator will become zero when $s = \pm jl\hat{\omega}_k$, and the many terms in sum notation will be reduced to only one term. Utilizing this, when $s = \pm jl\hat{\omega}_k$, we can equate the denominators of the transfer function as:

$$\begin{aligned} &b_1 s^2(K_{2kl}s + K_{1kl}l\hat{\omega}_k)\gamma_{kl}(s) \\ &= (s^4 + c_1 s^3 + c_2 s^2 + c_3 s + c_4)\prod\prod(s^2 + 2\epsilon_{pq}q\hat{\omega}_p s + (q\hat{\omega}_p)^2) \end{aligned} \tag{3.18}$$

For every pair of l and k , 3.18 generates two equations containing two unknowns K_{2kl} and K_{1kl}.

3.3.2 Linear dependence

When the $l\omega_k \neq q\omega_p$,for $p \neq k$, the gains of internal models can be solved as above. However, it cannot be solved if $l\omega_k = q\omega_p$ or $l\omega_k$ is pretty close to $q\omega_p$ because the denominator in T_{de} will be zero. To deal with this situation, we need to drop the redundant internal model. After calculating the gains, i.e. K_{1pq}, K_{2pq}, the original gains for these two repeated internal models will be $K_{1pq} = K_{1lq} = 0.5K_{1pq}$ and $K_{2pq} = K_{2lq} = 0.5K_{2pq}$.

3.4 Summary

This chapter shows the basics of this thesis which are done by prior researchers. At the beginning of this chapter, we showed how the internal models can be used to identify the frequencies. Mathematical derivations are illustrated both in continuous time domain and discrete time domain. Then, an alternative state-space representation in continuous time is shown.

Chapter 4

Algorithm Development

4.1 Alternative Model in Discrete Time

With the sample period of $T = 1$, the previous state space of each internal model IM_{pq} can be written as the time varying discrete state space model:

$$\begin{aligned} X_{pq}(k+1) &= A_{pq}X_{pq}(k) + B_{pq}e(k) \\ u_{pq}(k) &= C_{pq}X_{pq}(k) \end{aligned} \tag{4.1}$$

where the state is $X_{pq}(k) = \begin{bmatrix} x_{1pq}(k) & x_{2pq}(k) \end{bmatrix}^T$. And A_{pq}, B_{pq} and C_{pq} can be expressed as

$$\begin{aligned} A_{pq} &= \begin{bmatrix} \cos q\omega_p & -\sin q\omega_p \\ \sin q\omega_p & \cos q\omega_p \end{bmatrix} \\ B_{pq} &= \begin{bmatrix} \frac{K_{1ccpq}\sin q\omega_p}{\omega} - \frac{K_{2ccpq}(1-\cos q\omega_p)}{\omega} \\ \frac{K_{2ccpq}\sin q\omega_p}{q\omega_p} + \frac{K_{1ccpq}(1-\cos q\omega_p)}{q\omega_p} \end{bmatrix} = \begin{bmatrix} \bar{K}_{1pq} \\ \bar{K}_{2pq} \end{bmatrix} \\ C_{pq} &= \begin{bmatrix} 0 & 1 \end{bmatrix} \end{aligned} \tag{4.2}$$

The transfer function of this internal model , where we have dropped the time dependencies for the sack of simplicity, is:

$$T_{IM} = \frac{\left(K_{1ccpq} - K_{1ccpq}\cos q\omega_p + K_{2ccpq}\sin q\omega_p\right)z + K_{1ccpq} - K_{1ccpq}\cos q\omega_p - K_{2ccpq}\sin q\omega_p}{q\omega_p\left(z^2 - 2z\cos q\omega_p + 1\right)} \tag{4.3}$$

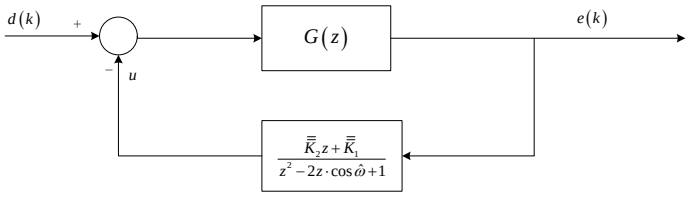

Figure 4.1: Block diagram of discrete system with one internal model

We assign $\bar{\bar{K}}$ as in Fig. 4.1 and deduce K backward as:

$$\begin{aligned}
\bar{\bar{K}}_{1pq} &= \frac{K_{1ccpq} - K_{1ccpq}\cos q\omega_p - K_{2ccpq}\sin q\omega_p}{q\omega_p} \\
\bar{\bar{K}}_{2pq} &= \frac{K_{1ccpq} - K_{1ccpq}\cos q\omega_p + K_{2ccpq}\sin q\omega_p}{q\omega_p} \\
K_{1ccpq} &= \frac{\left(\bar{\bar{K}}_{1pq} + \bar{\bar{K}}_{2pq}\right)q\omega_p}{2\left(1-\cos q\omega_p\right)} \\
K_{2ccpq} &= \frac{\left(\bar{\bar{K}}_{2pq} - \bar{\bar{K}}_{1pq}\right)q\omega_p}{2\sin q\omega_p}
\end{aligned} \tag{4.4}$$

Since it is the discrete time implementation of adaptive model in continuous time, the states of these two models are same when the input only changes at the sample times. Here, we use the same adaptation law as previous one:

$$\triangle\omega_p = \frac{K_{1ccp1}x_{2p1} - K_{2ccp1}x_{1p1}}{x_{1p1}^2 + x_{2p1}^2} \tag{4.5}$$

4.2 System with One Internal Model (Off-Line Tuning)

We have shown the off-line tuning in continuous time in Chapter 3. The tuning in discrete time is shown below.

4.2.1 Discrete time

The simple model structure in discrete time is shown in Fig. 4.1.

The transfer function of band-pass filter is

$$T_{bp}(z) = \frac{B\left(z^4 - 2z + 1\right)}{z^4 + C_1 z^3 + C_2 z^2 + C_3 z + C_4}. \tag{4.6}$$

The notch has the transfer function as

$$\begin{aligned} T_n(z) &= \frac{\left(z-e^{j\omega}\right)\left(z-e^{-j\omega}\right)}{\left(z-e^{-\varepsilon\omega+j\omega}\right)*\left(z-e^{-\varepsilon\omega-j\omega}\right)} \\ &= \frac{z^2-2z\cos\omega+1}{z^2-2z\rho\cos\omega+\rho^2} \end{aligned} \tag{4.7}$$

Here, the small number ρ is assigned as

$$\rho = e^{-\varepsilon\omega}$$

The point wise in time transfer function of the model is

$$T_{de}(z) = \frac{G(z)}{1 + G(z) \cdot \left(\frac{\bar{\bar{K}}_2 z + \bar{\bar{K}}_1}{\left(z-e^{j\omega}\right)\left(z-e^{-j\omega}\right)}\right)} \tag{4.8}$$

where

$$G(z) = \frac{b\left(z^4 - 2z^2 + 1\right)}{z^4 + a_1 z^3 + a_2 z^2 + a_3 z + a_4} \tag{4.9}$$

So, T_{de} can be expanded as

$$T_{de}(z) = \frac{b\left(z^4 - 2z^2 + 1\right) \cdot \left(z^2 - 2z\cos\omega + 1\right)}{\left(z^4 + a_1 z^3 + a_2 z^2 + a_3 z + a_4\right) \cdot \left(z^2 - 2z\cos\omega + 1\right) + b \cdot \left(z^4 - 2z^2 + 1\right) \cdot \left(\bar{\bar{K}}_2 z + \bar{\bar{K}}_1\right)} \tag{4.10}$$

After expanding the denominators of T_{de} and $G(z)$, we can get every coefficients of the polynomial as the Table. 4.1 below.

Here, we use a 4th-order Chebyshev Type I band-pass filter with a lower passband frequency of $1 \cdot 2\pi/400$ Hz and a higher passband frequency of $30 \cdot 2\pi/400$ Hz. The coefficients of the denominator of band-pass filter is shown as Table. 4.2.

Here, the signal we use has the frequency of $4.2 * 2\pi/400$ rad/s. By matching the coefficients, the coefficients of T_{de} is shown as in Table. 4.3.

Denominator coefficients of band-pass filter and model

	T_{de}	T_{bpn}
s^0	$a_4 + b\bar{\bar{K}}_1$	$T_0 = \rho^2 C_4$
s^1	$a_3 - 2b\bar{\bar{K}}_1 + b\bar{\bar{K}}_2 - 2a_4 \cos\omega$	$T_1 = C_3\rho^2 - 2C_4\rho\cos\omega$
s^2	$a_2 + a_4 - 2b\bar{\bar{K}}_2 - 2a_3 \cos\omega$	$T_2 = C_2\rho^2 - 2C_3\rho\cos\omega + C_4$
s^3	$a_1 + a_3 - 2a_2 \cos\omega$	$T_3 = C_1\rho^2 - 2C_2\rho\cos\omega + C_3$
s^4	$a_2 + b\bar{\bar{K}}_1 - 2a_1 \cos\omega + 1$	$T_4 = \rho^2 - 2C_1\rho\cos\omega + C_2$
s^5	$a_1 - 2\cos\omega + b\bar{\bar{K}}_2$	$T_5 = C_1 - 2\rho\cos\omega$
s^6	1	1

Table 4.1: Denominator coefficients of band-pass filter and model (discrete time)

Coefficients of band-pass filter

C_1	C_2	C_3	C_4	B	ρ_{pq}
−2.0264	1.4312	−0.7174	0.3153	0.2664	0.9934

Table 4.2: Coefficients of band-pass filter (discrete time)

Coefficients of model

a_1	$T_5 + 2\cos\omega - b\bar{\bar{K}}_2$	−2.0136
a_2	$T_4 - b\bar{\bar{K}}_1 + 2a_1\cos\omega - 1$	1.4173
a_3	$T_3 - a_1 + 2a_2\cos\omega$	−0.7125
a_4	$T_0 - b\bar{\bar{K}}_1$	0.3116
K_1	/	−0.0015
K_2	/	0.0015
b	B	0.2664

Table 4.3: Coefficients of model (discrete time)

Calculation of $\bar{\bar{K}}$

	T_{de} $= b \cdot \left(z^4 - 2z + 1\right) \cdot \left(\bar{\bar{K}}_2 z + \bar{\bar{K}}_1\right)$ $= zz_i \cdot \left(\bar{\bar{K}}_2 z_i + \bar{\bar{K}}_1\right)$	T_{bpn}
$z_1 = e^{j\omega}$	$zz_1 \cdot \bar{\bar{K}}_2 \cdot z_1 + zz_1 \cdot \bar{\bar{K}}_1$	$-2.8972 * 10^{-6} - 2.7712 * 10^{-6} j$
$z_2 = e^{-j\omega}$	$zz_2 \cdot \bar{\bar{K}}_2 \cdot z_2 + zz_2 \cdot \bar{\bar{K}}_1$	$-2.8972 * 10^{-6} + 2.7712 * 10^{-6} j$

Table 4.4: Calculation of $\bar{\bar{K}}$

Table. 4.4 shows a pair of equations with two unknowns, $\bar{\bar{K}}_1$ and $\bar{\bar{K}}_2$. This work is the foundation of later sections and makes our main algorithm easy to understand.

4.3 System with More than One Internal Model (Online tuning)

4.3.1 Continuous time

In chapter 3 state that the 4 parameters a_i can be solved by equating the denominators of the transfer function at any 4 imaginary random values of s. Thiswork is an original contribution of this thesis and was originally presented in [32]. We have that

$$\begin{aligned}\prod_{i=0}^{n_t}(s+r_i) &= s^{n_t} + \sum r_i s^{n_t-1} + \sum_i \sum_{j>i} r_i r_j s^{n_t-2} \\ &\quad + \cdots + s\sum_i \prod_{j\neq i} r_j + \prod r_i\end{aligned} \tag{4.11}$$

also,

$$\begin{aligned}\prod_{i=1}^{n_t}(s^2+2\epsilon_i\omega_i+\omega_i^2) &= s^{2n_t} + \sum 2\epsilon_i\omega_i s^{2n_t-1} \\ &\quad + \left(\sum \omega_i^2 + \sum_i \sum_{j>i} 4\epsilon_j\omega_j\epsilon_{\mathrm{i}}\omega_i\right) s^{2n_t-2} \\ &\quad + \cdots + s\sum_i 2\epsilon_i\omega_i \prod_{j\neq i}\omega_j^2 + \prod \omega_i^2\end{aligned} \tag{4.12}$$

So, a_i $(i = 1, 2, 3, 4)$ can be solved by matching the coefficients of the degree $0, 1, 2n_t + 2, 2n_t + 3$ terms of the denominators of T_{de} and T_{bpn}. Setting $\{\omega_p\} = \{q\omega_p\}$ and $\{\epsilon_p\} = \{\epsilon_{pq}\}$, we get

$$\begin{aligned}a_1 &= c_1 + \sum 2\epsilon_p\omega_p \\ a_2 &= \sum_{p=1}^{n_t}\sum_{q>i} 4\epsilon_q\omega_q\epsilon_{\mathrm{p}}\omega_{\mathrm{p}} + c_1\sum_{p=1}^{n_t} 2\epsilon_p\omega_p + c_2 \\ a_3 &= c_3 + c_4\sum_{p=1}^{n_t} 2\epsilon_p/\omega_p \\ a_4 &= c_4\end{aligned} \tag{4.13}$$

4.3.2 Discrete time

The signal we use in discrete time is of the following form:

$$d(k) = \sum_{p=1}^{n}\sum_{q=1}^{m_p} \overline{A}_{pq}(k)\cos\phi(k) + n(k) \tag{4.14}$$

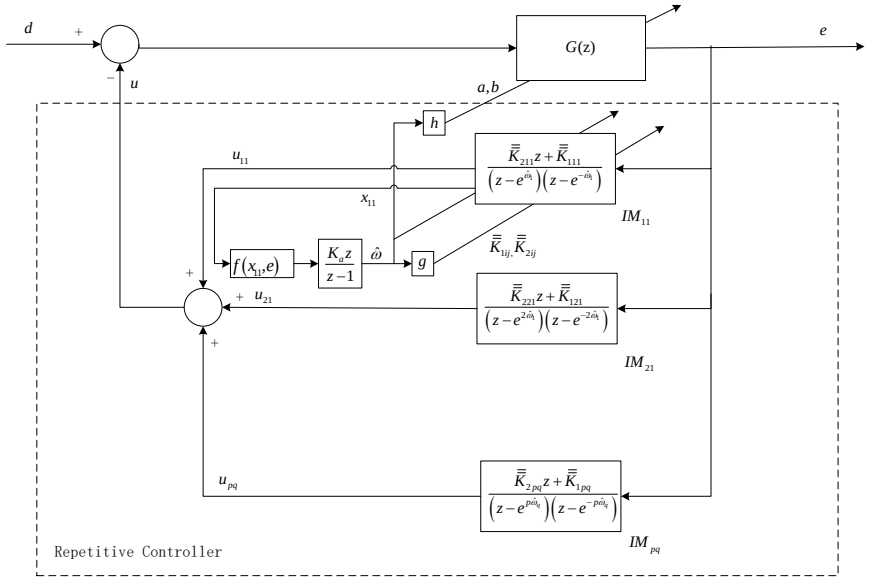

Figure 4.2: Structure of signal identification based on the internal model

where the phase is given by

$$\phi_{pq}(k) = \sum_{i=1}^{k} q\omega_p(k) + \phi_{pq}(0) \tag{4.15}$$

Here, n is noise and the signals are the sum of n time-varying sinusoidal waves with m_p harmonics each. It is assumed that the frequencies and amplitudes vary slowly in time. $\overline{A}_{pq}$,ω_p and ϕ_{pq} are unknown and uncertain magnitude, frequency, and relative phases respectively.

Fig. 4.2 shows structure of the instantaneous Fourier decomposition algorithm. $G(z)$ is a tuning function paralleled with internal models. Functions of f, g and h is shown in Eq. 4.5, Eq. 4.4 and Eq. 4.25 to 4.28 respectively. Each transfer function IM_{pq} represents a corresponding sinusoidal with the frequency of $q\omega_p$. The transfer function of each internal model has poles at $e^{\pm q\hat{\omega}_p}$. Note that the n fundamental frequencies are calculated only using the states of the fundamental harmonic model. This is based on the assumption that the fundamental typically has more energy than the harmonics. This is not necessary, any model can be used in the case where the fundamental is absent. Alternative choices include estimating the frequency based on the model with the greatest energy, or a weighted average of the models.

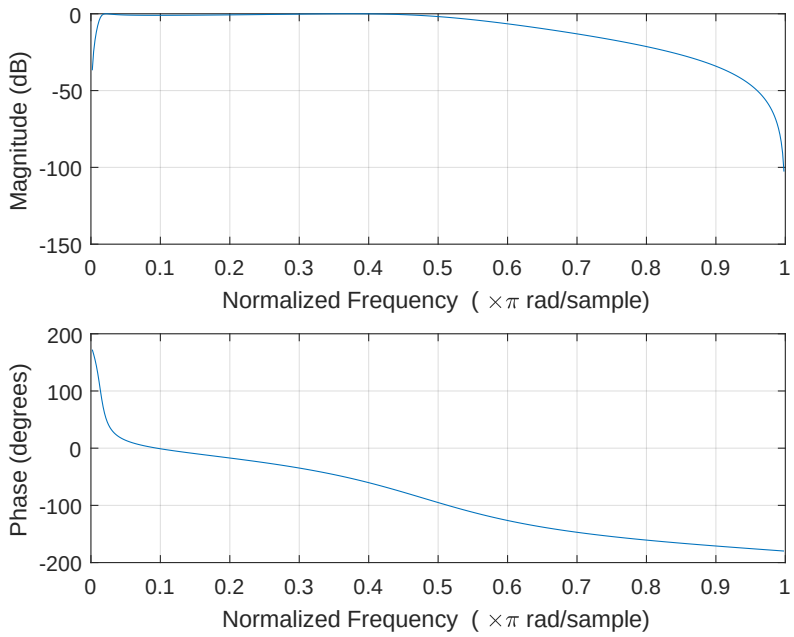

Figure 4.3: Bode diagram of band-pass filter

To identify all frequencies in the signal in equation 4.14, a suitable gain needs to be chosen to so that this remains a stable feedback loop. This situation has already been successfully implemented to reject unknown periodic disturbances in magnetic hard disk drives[36]. However, this brings a lot of restrictions to our algorithm. As we discussed in Chapter 2, there is no control over the dynamic system in our case, thus, we need to tune the gain adaptively.

4.3.3 Band-pass filter

Fig. 4.3 and Fig. 4.4 show the theoretical Bode plot of desired band-pass filter and band-pass filter with ten notches respectively. Here, we use a 4th-order Chebyshev Type I band-pass filter with a lower passband frequency of $1 \cdot 2\pi/400$ Hz and a higher passband frequency of $30 \cdot 2\pi/400$ Hz.

4.3.4 Parameter calculation

The transfer function of band-pass filter with notches is

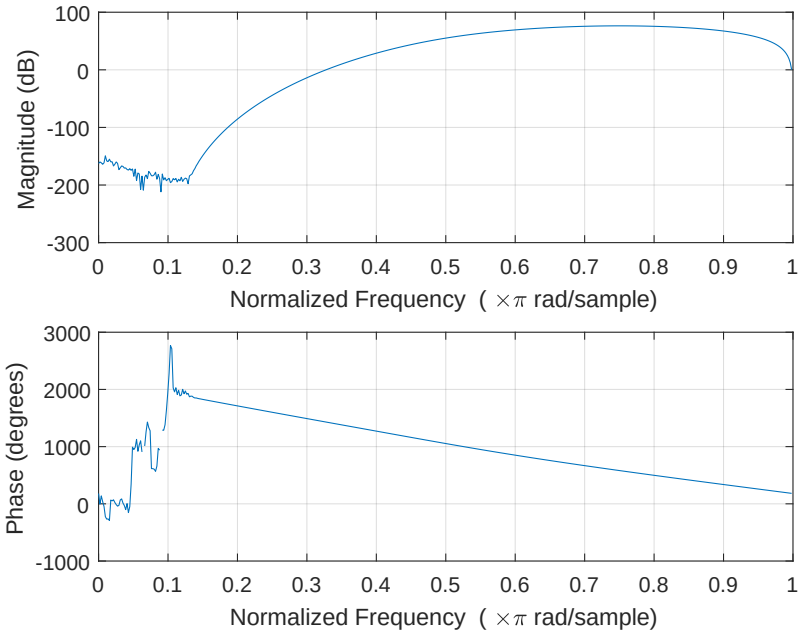

Figure 4.4: Bode diagram of band-pass filter with notches

$$\begin{aligned} T_{bpn}(z) &= \frac{B_1\left(z^4-2z+1\right)}{z^4+C_1z^3+C_2z^2+C_3z+C_4} * \prod \frac{\left(z-e^{jq\omega_p}\right)\left(z-e^{-jq\omega_p}\right)}{\left(z-\rho_{pq}e^{jq\omega_p}\right)*\left(z-\rho_{pq}e^{-jq\omega_p}\right)} \\ &= \frac{B_1\left(z^4-2z+1\right)}{z^4+C_1z^3+C_2z^2+C_3z+C_4} * \prod \frac{z^2-2z\cos q\omega_p+1}{z^2-2z\rho_{pq}\cos q\omega_p+\rho_{pq}^2} \end{aligned} \tag{4.16}$$

Here, the small number ρ_{pq} is calculated by

$$\rho_{pq} = e^{-\varepsilon_{pq}q\omega_p}$$

Here, ε_{pq} are small real numbers, and $q\omega_p$ are the notches frequency. The presence of zeros (roots) at $e^{jq\omega_p}$ in the numerator is a fundamental consequence of the internal model principle.

Therefore, the algorithm has a better ability to improve noise rejection. By transferring the system from the s-domain, the point wise in time transfer function from d to e in z-domain is

$$T_{de}(z) = \frac{G(z)}{1+G(z)\sum_{p=1}^{n}\sum_{q=1}^{m_p}\left(\frac{\bar{\bar{K}}_{2pq}z+\bar{\bar{K}}_{1pq}}{\left(z-e^{jq\omega_p}\right)\left(z-e^{-jq\omega_p}\right)}\right)} \tag{4.17}$$

The transfer function of the plant is

$$G(z) = \frac{b_1\left(z^4 - 2z + 1\right)}{z^4 + a_1 z^3 + a_2 z^2 + a_3 z + a_4} = \frac{b(z)}{a(z)} \tag{4.18}$$

The numerator of T_{de} is

$$T_{de_{num}}(z) = b_1\left(z^4 - 2z + 1\right)\prod\left(z^2 - 2z\cos q\omega_p + 1\right) \tag{4.19}$$

and the denominator of T_{de} is

$$\begin{array}{rc} T_{de_{den}}(z) & = a(z)\prod\left(z^2 - 2z\cos q\omega_p + 1\right) \\ & +b_1\left(z^4 - 2z + 1\right)\sum_{p=1}^{n}\sum_{q=1}^{m_p}\left[\left(\bar{\bar{K}}_{2pq}z + \bar{\bar{K}}_{1pq}\right)\Upsilon_{pq}(z)\right] \end{array} \tag{4.20}$$

where

$$\Upsilon_{pq}(z) = \prod_{k=1}^{n}\quad\prod_{\substack{l=1 \\ l\neq p\ if\ p=k}}^{m_p}\left(z^2 - 2z\cos l\omega_k + 1\right) \tag{4.21}$$

Apparently, by matching the numerator of T_{de} and T_{bpn}, we can get that

$$b_1 = B_1 \tag{4.22}$$

As what we showed in Chapter 3, if we look at the equations carefully, we can find that when $z = e^{\pm jq\omega_p}$, we have

$$b_1\left(z^4 - 2z + 1\right)\sum_{p=1}^{n}\sum_{q=1}^{m_p}\left(\bar{\bar{K}}_{2pq}z + \bar{\bar{K}}_{1pq}\right)\Upsilon_{kl}(z) = C(z) * \prod\left(z^2 - 2z\rho_{pq}\cos q\omega_p + \rho_{pq}^2\right) \tag{4.23}$$

Thus, every substitution of $e^{\pm jq\omega_p}$ will generate 2 equations with two unknowns of $\bar{\bar{K}}_{2pq}$ and $\bar{\bar{K}}_{1pq}$.

We can find explicit solution for a_i $(i = 1, 2, 3, 4)$by equating the coefficients of z^0, z^1,$z^{2m_p+2}$and

z^{2m_p+3}. As what we have in continuous time, we have the formulas below in discrete time.

$$\begin{aligned}\prod_{i=1}^{n}\left(z^2-2\rho\cos p\omega_q z+\rho^2\right) &= z^{2n}+\left[\sum_{i=1}^{n}\left(-2\rho\cos p\omega_q\right)\right]z^{2n-1}\\ &+\left[\sum_{i=1}^{n}\left(\rho^2\right)+\sum_{i=1}^{n}\sum_{\substack{j=1\\ j\succ i}}^{m}\left(-2\rho\cos p\omega_q\right)\right]z^{2n-2}\\ &+\cdots+\left[\sum_{i=1}^{n}\left(-2\rho\cos\omega_{ij}\right)\sum_{\substack{j=1\\ j\succ i}}^{m}\left(\rho^2\right)\right]z+\prod_{i,j=1}^{n}\left(\rho^2\right)\end{aligned}$$

and

$$\begin{aligned}\prod_{i=1}^{n}\left(z^2-2\cos\omega_{ij}z+1\right) &= z^{2n}+\left[\sum_{i=1}^{n}\left(-2\cos\omega_{ij}\right)\right]z^{2n-1}\\ &+\left[\sum_{i=1}^{n}(1)+\sum_{i=1}^{n}\sum_{\substack{j=1\\ j\succ i}}^{m}\left(-4\cos\omega_i\cos\omega_j\right)\right]z^{2n-2}\\ &+\cdots+\left[\sum_{i=1}^{n}\left(-2\cos\omega_{ij}\right)\sum_{\substack{j=1\\ j\succ i}}^{m}(1)\right]z+\prod_{i,j=1}^{n}\end{aligned} \tag{4.24}$$

We can calculate a as:

$$\begin{aligned}a_1 &= C_1+C_0\sum_{p=1}^{n}\sum_{q=1}^{m_p}\left(-2\rho_{pq}\cos q\omega_p\right)\\ &-\sum_{p=1}^{n}\sum_{q=1}^{m_p}\left(-2\cos q\omega_p\right)-b_1*\sum_{p=1}^{n}\sum_{q=1}^{m_p}\left(\bar{\bar{K}}_{2pq}*1^9\right)\end{aligned} \tag{4.25}$$

$$
\begin{aligned}
a_2 \;&= C_0 \textstyle\sum_{p=1}^{n} \sum_{q=1}^{m_p} \left(\rho_{pq}^2\right) - a_1 * \sum_{p=1}^{n} \sum_{q=1}^{m_p} \left(-2\cos q\omega_p\right) \\
&\quad + C_0 * \sum_{p=1}^{n} \sum_{q=1}^{m_p} \sum_{k=1}^{n} \sum^{m_p}_{\substack{l=1 \\ l \neq q if p=k}} \left(-2\rho_{pq}\cos q\omega_p\right)\left(-2\rho_{kl}\cos l\omega_k\right) \\
&\quad - b_1 * \sum_{p=1}^{n} \sum_{q=1}^{m_p} \left[\bar{\bar{K}}_{2pq} * \sum_{k=1}^{n} \sum^{m_p}_{\substack{l=1 \\ l \neq q if p=k}} \left(-2\cos l\omega_k\right) + \bar{\bar{K}}_{1pq} \right] \\
&\quad - \sum_{p=1}^{n} \sum_{q=1}^{m_p} \sum_{k=1}^{n} \sum^{m_p}_{\substack{l=1 \\ l \neq q if p=k}} \left(-2\cos q\omega_p\right)\left(-2\cos l\omega_k\right) \\
&\quad + C_1 \textstyle\sum_{p=1}^{n} \sum_{q=1}^{m_p} \left(-2\rho_{pq}\cos q\omega_p\right) - \sum_{p=1}^{n} \sum_{q=1}^{m_p} (1)
\end{aligned}
\tag{4.26}
$$

$$
\begin{aligned}
a_3 \;&= C_3 \textstyle\prod_{p=1}^{n} \prod_{q=1}^{m_p} \left(\rho_{pq}^2\right) - a_4 * \sum_{p=1}^{n} \sum_{q=1}^{m_p} \left(-2\cos q\omega_p\right) \\
&\quad + C_4 \sum_{p=1}^{n} \sum_{q=1}^{m_p} \left[\left(-2\rho_{pq}\cos q\omega_p\right) \prod_{k=1}^{n} \prod^{m_p}_{\substack{l=1 \\ l \neq q if p=k}} \left(\rho_{kl}^2\right) \right] \\
&\quad - b_1 * \sum_{p=1}^{n} \sum_{q=1}^{m_p} \left[\bar{\bar{K}}_{2pq} + \bar{\bar{K}}_{1pq} * \sum_{k=1}^{n} \sum^{m_p}_{\substack{l=1 \\ l \neq q if p=k}} \left(-2\cos l\omega_k\right) \right]
\end{aligned}
\tag{4.27}
$$

$$
a_4 = C_4 \prod_{p=1}^{n} \prod_{q=1}^{m_p} \left(\rho_{pq}^2\right) - b_1 \sum_{p=1}^{n} \sum_{q=1}^{m_p} \left[\bar{\bar{K}}_{1pq} * 1^9 \right] \tag{4.28}
$$

4.3.5 Linear dependence

When the $l\omega_k \neq q\omega_p$,for $p \neq k$, the gains of internal models can be solved as above. However, it cannot be solved if $l\omega_k = q\omega_p$ or $l\omega_k$ is pretty close to $q\omega_p$ because the denominator in T_{de} will be zero. To deal with this situation, we need to drop the redundant internal model. After calculating the gains, i.e. $\tilde{K}_{1pq}$, $\tilde{K}_{2pq}$, the original gains for these two repeated internal models will be $\bar{\bar{K}}_{1pq} = \bar{\bar{K}}_{1lq} = 0.5\tilde{K}_{1pq}$ and $\bar{\bar{K}}_{2pq} = \bar{\bar{K}}_{2lq} = 0.5\tilde{K}_{2pq}$.

4.4 Improved Approach

4.4.1 Method

The primary structure of the closed loops system was chosen to be band-pass in nature as the frequency identification algorithm will not work if the error signal contains a DC component, thus the necessity of eliminating low frequency components. The speed with which the effects of initial condition error is governed by the slowest closed loop poles. These are the poles for at the low frequency cutoff of the band pass filter. Alternatively, a integral controller, the simplest internal model controller can be added to the other internal model controllers and the band-pass filter can be replace by a low pass filter. The slowest closed loop poles of these system are the high frequency cutoff poles, resulting in a system with significantly improved transient characteristics. This means the order of the filter decreased from 4 to 2.

4.4.2 Low-pass filter

The transfer function of low-pass filter is designed as

$$T_{lp}(z) = \frac{B_1 (z+1)^2}{z^2 + C_1 z + C_2} \tag{4.29}$$

The point wise in time transfer function of low-pass filter with notches is

$$\begin{aligned} T_{lpn}(z) &= \frac{B_1(z+1)^2}{z^2+C_1 z+C_2} \cdot \prod \frac{\left(z-e^{jq\omega_p}\right)\left(z-e^{-jq\omega_p}\right)}{\left(z-\rho_{pq}e^{jq\omega_p}\right)\cdot\left(z-\rho_{pq}e^{-jq\omega_p}\right)} \cdot \frac{z-1}{z-\rho} \\ &= \frac{B_1(z+1)^2}{z^2+C_1 z+C_2} \cdot \prod \frac{z^2-2z\cos q\omega_p+1}{z^2-2z\rho_{pq}\cos q\omega_p+\rho_{pq}^2} \cdot \frac{z-1}{z-\rho} \end{aligned} \tag{4.30}$$

Here, the small number ρ_{pq} is calculated by:

$$\rho_{pq} = e^{-\varepsilon_{pq} q\omega_p}$$

Fig. 4.5 show the theoretical bode plot of desired low-pass filter. Here, we use a 2nd-order Chebyshev Type I low-pass filter with a lower passband frequency of $1 \cdot 2\pi/400$ Hz and a higher

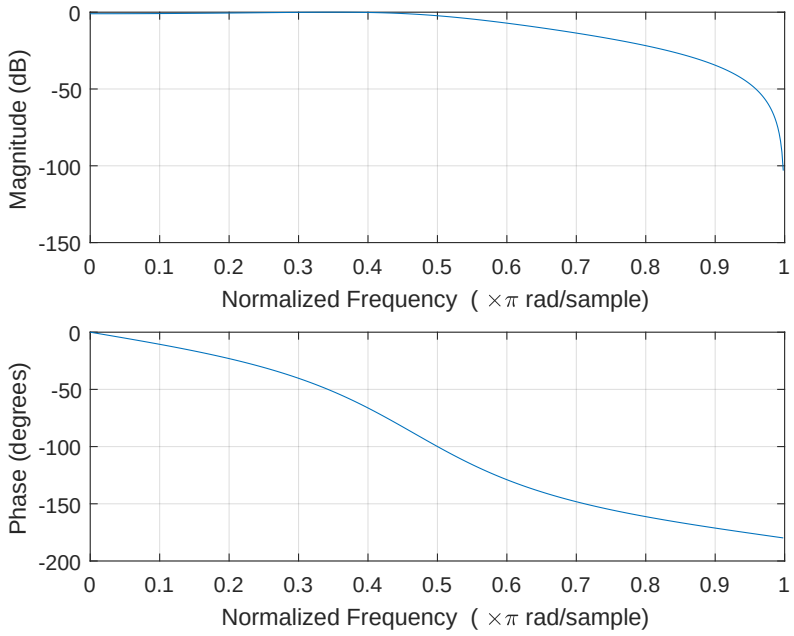

Figure 4.5: Bode diagram of low-pass filter

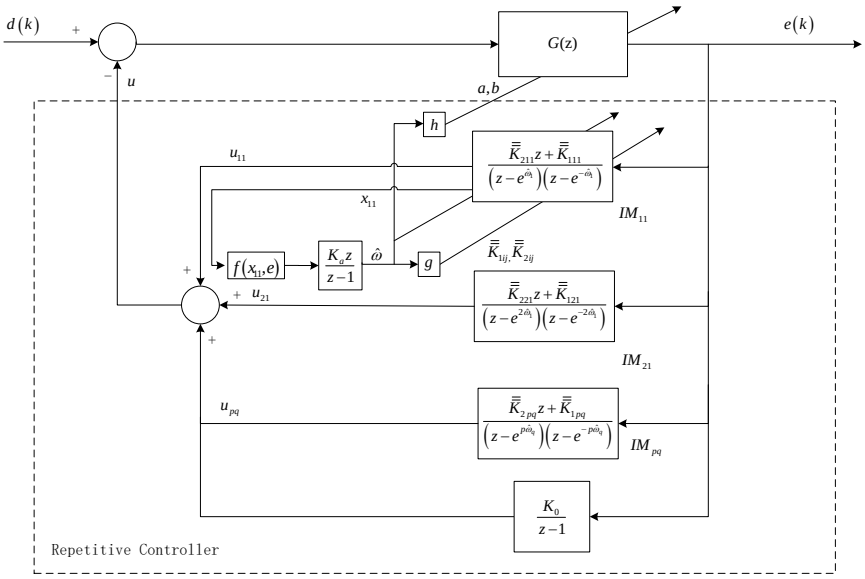

Figure 4.6: Improved structure of signal identification with model of DC content

passband frequency of $30 \cdot 2\pi/400$ Hz.

4.4.3 Improved system

Functions of f and g is shown in Eq. 4.5 and Eq. 4.4. h can be calculated by matching Table. 4.5 and Table. 4.6 respectively. The point wise in time transfer function from d to e in z-domain is

$$T_{de}(z) = \frac{G(z)}{1 + G(z)\sum_{p=1}^{n}\sum_{q=1}^{m_p}\left(\frac{\bar{\bar{K}}_{2pq}z+\bar{\bar{K}}_{1pq}}{(z-e^{jq\omega_p})(z-e^{-jq\omega_p})} + \frac{K_0}{z-1}\right)} \tag{4.31}$$

The point wise in time transfer function of the plant is

$$G(z) = \frac{b_1(z+1)^2}{z^2 + a_1 z + a_2} = \frac{b(z)}{a(z)} \tag{4.32}$$

The numerator of T_{de} is

$$T_{de_{num}} = b_1(z+1)^2(z-1)\prod\left(z^2 - 2z\cos q\omega_p + 1\right) \tag{4.33}$$

and the denominator of T_{de} is

$$\begin{aligned} T_{de_{den}} \quad = \quad & \left(z^2 + a_1 z + a_2\right)(z-1)\prod\left(z^2 - 2z\cos q\omega_p + 1\right) \\ & + b_1(z+1)^2(z-1)\sum_{p=1}^{n}\sum_{q=1}^{m_p}\left[\left(\bar{\bar{K}}_{2pq}z + \bar{\bar{K}}_{1pq}\right)\Upsilon_{pq}(z)\right] \\ & + b_1(z+1)^2 K_0\prod\left(z^2 - 2z\cos q\omega_p + 1\right) \end{aligned} \tag{4.34}$$

where

$$\Upsilon_{pq}(z) = \prod_{k=1}^{n}\prod_{\substack{l=1 \\ l \neq p\ if\ p=k}}^{m_p}\left(z^2 - 2z\cos l\omega_k + 1\right) \tag{4.35}$$

4.4.4 Parameter calculation

To simplify the calculation, when $z = 1$, we have

$$\begin{aligned} T_{lp_{den}} &= (1 + C_1 + C_2)\cdot\prod\left(1 - 2\rho_{pq}\cos q\omega_p + \rho_{pq}^2\right)\cdot(1-\rho) \\ T_{de_{den}} &= 4\cdot b_1 K_0\prod\left(1 - 2\cos q\omega_p + 1\right) \end{aligned} \tag{4.36}$$

Then, we can calculate K_0 as

Coefficients of $T_{lp_{den}}$

	$T_{lp_{den}}$
z^0	$C_2 \cdot \prod \rho_{pq}^2 \cdot (-\rho)$
z^1	$C_1 \cdot \prod \rho_{pq}^2 \cdot (-\rho)$ $+C_2 \sum_{k=1}^{n} \left(-2 \cdot \rho_{pq} \cdot \cos q\omega_p \cdot \prod^{m_p}_{\substack{l=1 \\ l \neq p \; if \; p=k}} \left(\rho_{lk}^2\right) \right) \cdot (-\rho)$ $+C_2 \cdot \prod \rho_{pq}^2 \cdot 1$

Table 4.5: Coefficients of $T_{lp_{den}}$

Coefficients of $T_{de_{den}}$

	$T_{de_{den}}$
z^0	$a_2 \cdot (-1) + b_1 \cdot (-1) \cdot \sum \bar{\bar{K}}_{1pq} + b_1 \cdot K_0$
z^1	$a_1 \cdot (-1) \cdot 1 + a_2 \cdot 1 \cdot 1 + a_2 \cdot (-1) \cdot \sum \left(-2 * \cos q\omega_p\right)$ $+b_1 \cdot 2 \cdot K_0 \cdot 1 + b_1 \cdot 1 \cdot K_0 \cdot \sum \left(-2 * \cos q\omega_p\right) + b_1 \cdot$ $\{2 \cdot (-1) \cdot \sum \bar{\bar{K}}_{1pq} + 1 \cdot 1 \cdot \sum \bar{\bar{K}}_{1pq} +$ $1 \cdot (-1) \cdot \left[\sum \bar{\bar{K}}_{2pq} + \sum \left(\bar{\bar{K}}_{1pq} \cdot \sum^{m_p}_{\substack{l=1 \\ l \neq p \; if \; p=k}} (-2 \cdot \cos l\omega_k) \right) \right] \}$

Table 4.6: Coefficients of $T_{de_{den}}$

$$K_0 = \frac{(1 + C_1 + C_2) \cdot \prod \left(1 - 2\rho_{pq} \cos q\omega_p + \rho_{pq}^2\right)(1 - \rho)}{4 \cdot b_1 \prod \left(1 - 2\cos q\omega_p + 1\right)} \tag{4.37}$$

By substituting $z = e^{\pm jq\omega_p}$, we have

$$\begin{aligned} &b_1 (z+1)^2 (z-1) \sum_{p=1}^{n} \sum_{q=1}^{m_p} \left[\left(\bar{\bar{K}}_{2pq} z + \bar{\bar{K}}_{1pq} \right) \Upsilon_{pq}(z) \right] \\ &= \left(z^2 + C_1 z + C_2\right) \cdot \prod \left(z^2 - 2z\rho_{pq} \cos q\omega_p + \rho_{pq}^2\right)(z - \rho) \end{aligned} \tag{4.38}$$

Thus, every substitution of $e^{\pm jq\omega_p}$ will generate 2 equations with two unknowns of $\bar{\bar{K}}_{2pq}$ and $\bar{\bar{K}}_{1pq}$.

By equating the coefficient of z^0 and z^1 in both $T_{lp_{den}}$ and $T_{de_{den}}$, we can calculate a. The coefficients are showed in Table. 4.5 and Table. 4.6 respectively.

4.5 Summary

This chapter shows the main method of our algorithm which is based on the internal model principle and repetitive controller. At the beginning of this chapter, we showed the internal models in continuous time domain and discrete time domain. Then, we implemented one internal model and calculated the parameters off-line. After analyzing the drawbacks of it, we used adaptive parameters and implemented more internal models in continuous time. As the main contribution in this chapter, we implemented this adaptive model in discrete time in sec. 4.3. To improve the system, a low-pass filter is used and the new system is successfully implemented in sec. 4.4. The simulation results of these approaches above will be shown in the next chapter.

Chapter 5

Simulation Results and Comparison

5.1 Simulation and Comparison

In this chapter, the simulation results are shown and analyzed. Simulations are carried out both for the band-pass and low-pass filter based algorithms. Not only the algorithm shown in previous chapter, but also the comparison with other algorithms will be shown in this chapter. Simulations are carried out under MAT-LAB/SIMULINK R2016b environment where Solver is discrete (no continuous states) and fixed step size is 1 sample per sample. All random noises are zero mean.

5.1.1 Signal to be identified

Our input is the sum of two signals which are the outputs of two same models shown in Fig. 5.1. The reciprocal block $1/u$ is to convert a period to the frequency. The initial conditions for both fundamental frequencies are 0.0105 and 0.0125 rad/sample respectively. The feedback loop contains a pure delay which can provide any periodic disturbance with period given by the delay. The low-pass filter restricts the energy to frequencies below 0.05 rad/sample and variance of 0.5. The disturbance input keeps the amplitudes and relative phases of this disturbance vary slowly. The band-limit and variance are 0.25 π rad/sample and 0.1 respectively. The initial condition for these two fundamental frequencies are 0.021π and 0.025π rad/sample. The reciprocal unit is to

convert the model frequencies to the period required in the feedback loop. The disturbances input to the repetitive controller guarantee that the amplitude and phase vary slowly.

Additional noise added to the first signal has the cutoff frequency of 0.25π rad/sample and variance of 0.001. The signals to be identified are shown in Fig. 5.2. We can see that the first signal has a period around 95 samples, while the other one has 85 samples as its period.

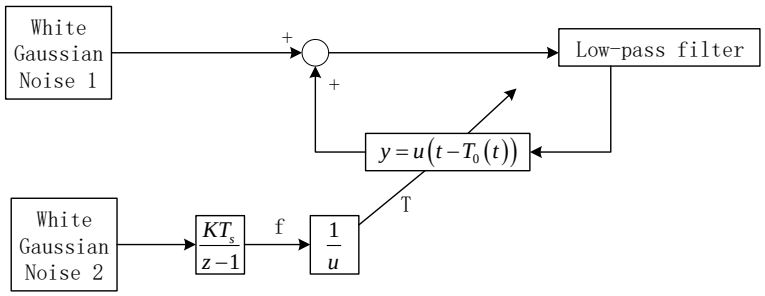

Figure 5.1: Block diagram of periodic signals generator

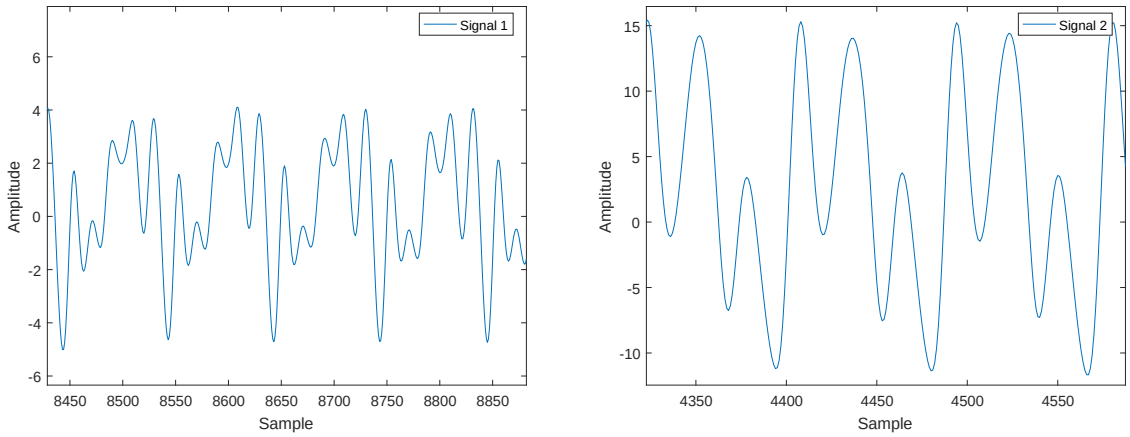

Figure 5.2: Periodic signals

5.1.2 Harmonic magnitude

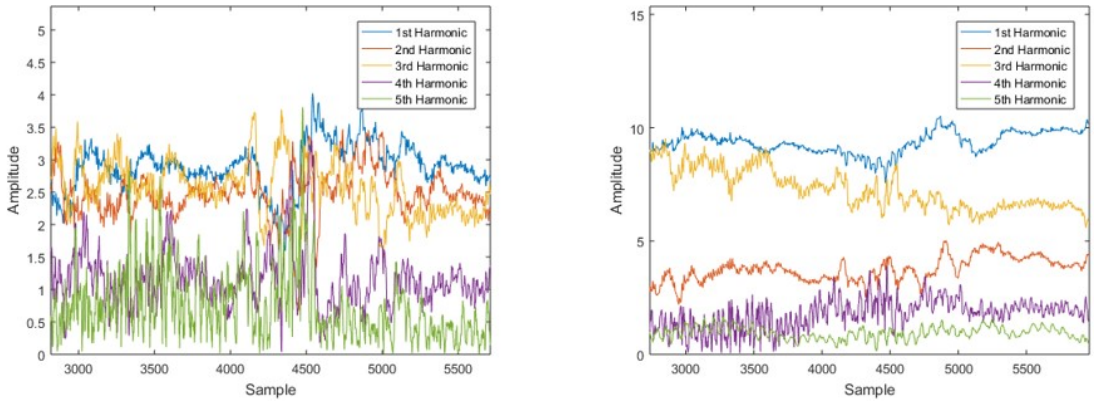

Figure 5.3: Amplitude of first and second set of harmonics

A harmonic of such a wave is a wave with a frequency that is a positive integer multiple of the frequency of the original wave, known as the fundamental frequency. As we mentioned in Chapter one, our signal contains n fundemental frequencies with m_p harmonics each. Here, in the simulation, we chose $n = 2$ and $m_p = 5$. So, we have 2 sets of harmonics, each of which contains 5 harmonics. Fig. 5.3 shows the number of harmonics and identification of each amplitude. The 1st harmonic is our fundamental frequency, while 2nd, 3rd and 4th harmonics are integer multiples of this fundamental frequency. As we can see, the amplitudes of harmonics can cross each other. Unfortunately, with the signal generated by the repetitive controller model, we do not know the true harmonic component of the signal. Since the harmonics are driven by a white noise source, it can be kind of noisy as in these figures.

5.1.3 Band-pass filter

The adaptive gains for both frequencies are chose as $K_a = 0.0035$. The overall transfer function is designed as a 4th order band-pass digital Chebyshev filter with 1db peak-to-peak ripple in the pass-band, and the pass band-edge frequencies are 0.15π and 0.005π rad/sample. The transfer function of this filter is

$$T_{bp}(z) = \frac{0.2664z^4 - 0.5327z^2 + 0.2664}{z^4 - 2.026z^3 + 1.431z^2 - 0.7174z + 0.3153}$$

The initial tuning function $G(z)$ is

$$G(z) = \frac{0.2664z^4 - 0.5327z^2 + 0.2664}{z^4 - 1.7471z^3 + 0.8982z^2 - 0.4651z + 0.3164}$$

5.1.3.1 Parameter calculation

The number of internal models is same as the number of notches. Thus, all frequencies with different harmonics can be identified. As we discussed in Chapter 4, the initial gains of 10 internal models can be calculated offline as shown in Table. 5.1.

Values Of $\bar{\bar{K}}_{pq}$

p	q	$\bar{\bar{K}}_{1pq}$	$\bar{\bar{K}}_{2pq}$
1	1	−0.01404	0.01397
1	2	−0.02613	0.02351
1	3	−0.05138	0.04681
1	4	−0.08282	0.09024
1	5	0.05156	0.006763
2	1	−0.01344	0.01415
2	2	−0.04018	0.03903
2	3	−0.0769	0.06931
2	4	−0.1793	0.1672
2	5	0.01195	0.02316

Table 5.1: Values of $\bar{\bar{K}}$ in algorithm based on band-pass filter

5.1.3.2 Frequency identification

Under these conditions, a 20000-samples Mat-lab simulation could be performed in under 5s. The identified frequencies are shown in Fig. 5.4 from which we can see that the identified signal tracks the fundamental one in a good way.

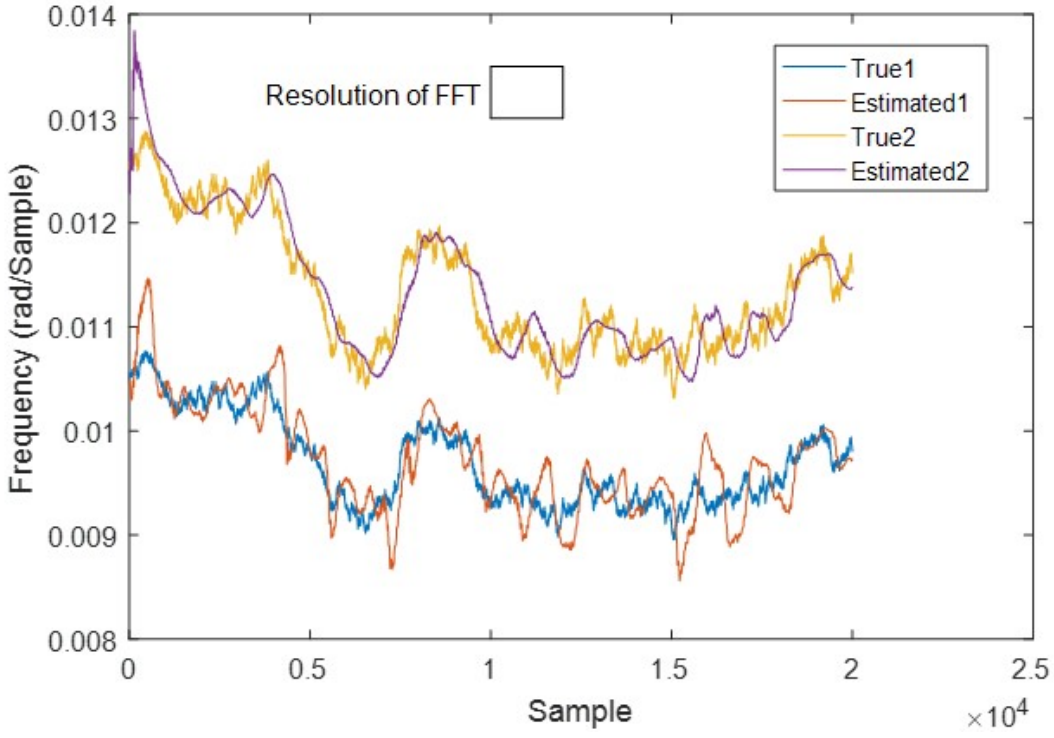

Figure 5.4: Frequency identification for first fundamental component of proposed tuning function

In Fig. 5.5, the estimated signal converges at around 400 samples. This is half the result in [32] as shown in Fig. 5.6. Note that theory for this algorithm shows that his algorithm will be approximately linear with a time constant given by $1/K_a$. This means that the implementation of this algorithm in discrete time not only has good tracking but also saves a lot of time. The area of the small box at the top of this figure represents the time/frequency resolution of FFT or similar algorithms such as wavelet analysis. The height of the window represents the frequency resolution which equals $1/T$, where $T = N/f_s$,T is window width and represents the time resolution (f_s is sampling frequency, N is the number of samples). The area of this box is fixed. If its height is halved the width is doubled. In this particular case, the height of the box is 0.0005 cycles/sample.

To achieve this frequency resolution requires a window of 20000 samples or the width of the box. The signal we are trying to analyze cannot be adequately represented by this box, because the box is bigger than the fluctuations in the signal that we are trying to capture. This box also approximately applies to wavelet, Gabor or anything this is similar in nature with FFT which are governed by time-frequency resolution trade-off. The difference in the various approach would be represented by different shapes of the box, for example making it more triangular. Obviously, other shapes with the same height and width would have lesser areas. Since the FFT averages over the width of the window, it is clear that one cannot get a time frequency resolution with the FFT appropriate for this signal.

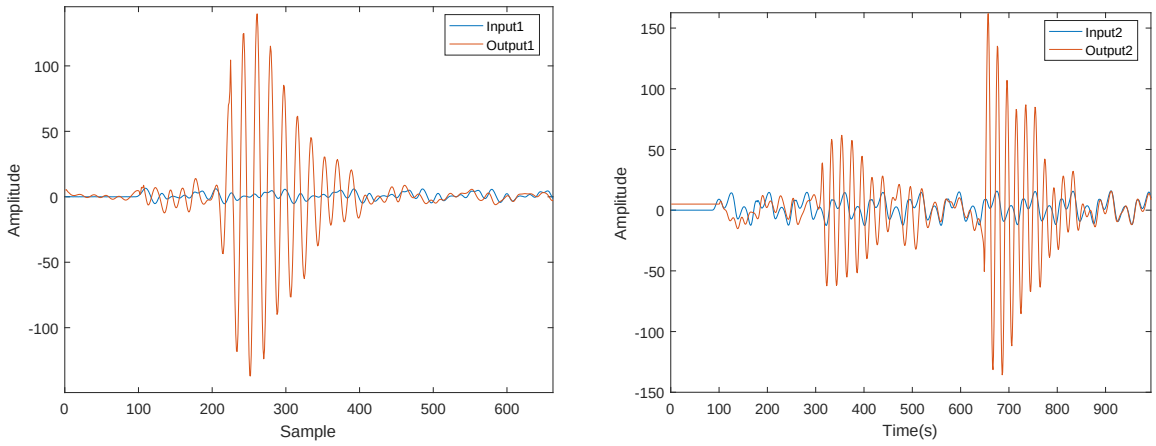

Figure 5.5: Convergence of the first and second component of band-pass tuning function

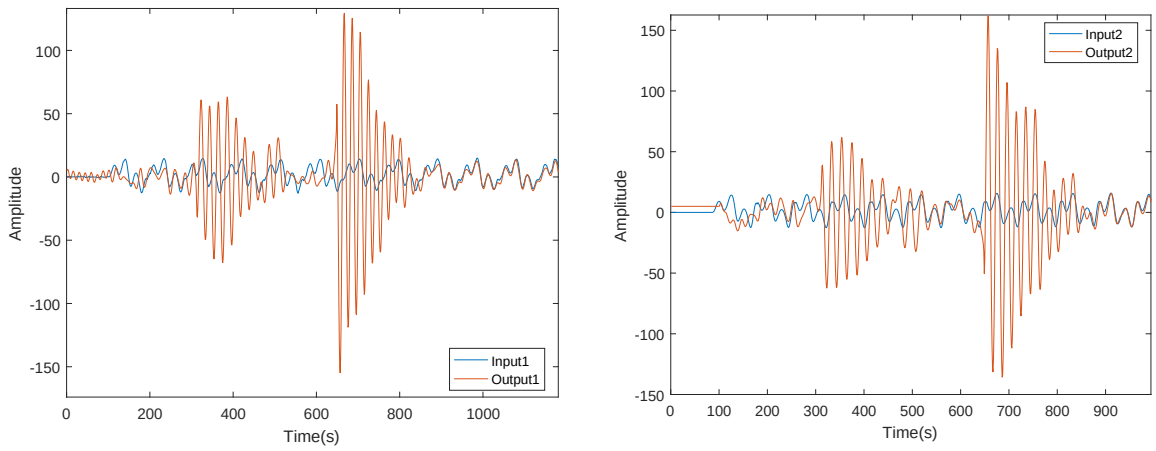

Figure 5.6: Convergence of the first and second component of continuous implementation

As shown in Fig. 5.6, after 400 samples, the first identified signal tracks the output of first signal generator both in amplitude and phase, so does the second identified signal as shown. Because of difficulties in setting initial conditions for the signal generator, the signal does not commence till a full period has passed, ie. around the 80th sample. Since the algorithm requires a signal to be presented, frequency tuning does not commence till sample 80.

In addition, the sum of input $d(t)$ and output $y(t)$ signals are compared by the given figures of Fig. 5.7. Although it can be identified in a good way, we can still see that there exists a significant DC component between the actual and identified amplitude. This is reasonable because we did not deal with the DC component yet.

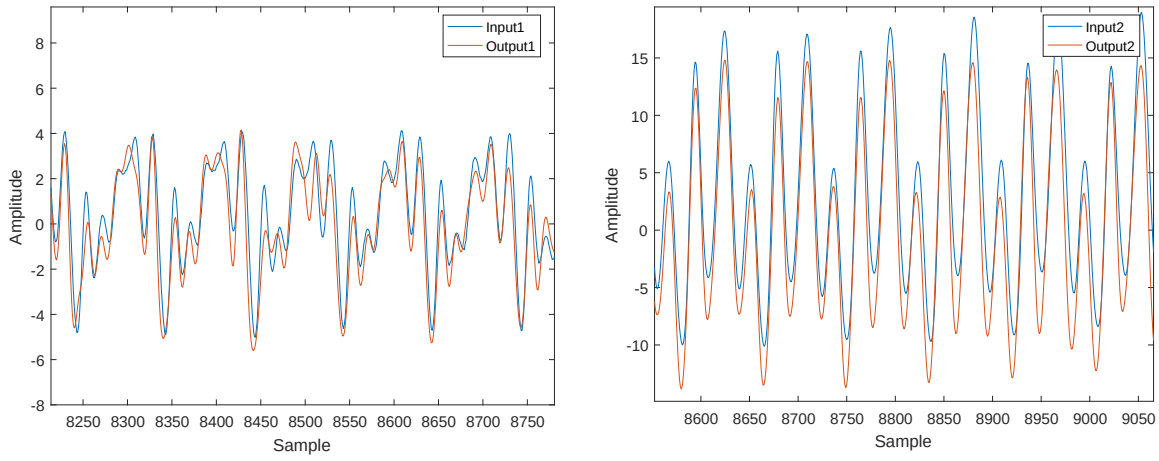

Figure 5.7: Frequency identification for first and second component of proposed algorithm

By doing the Fourier transform of the identified signal and the error, we can get a better view of the algorithm. We can see from Fig. 5.8 that every harmonic can be identified. Since we did not deal with the DC component, there is a huge DC component error. The DC component of the error is reletively small in Fig. 5.9.

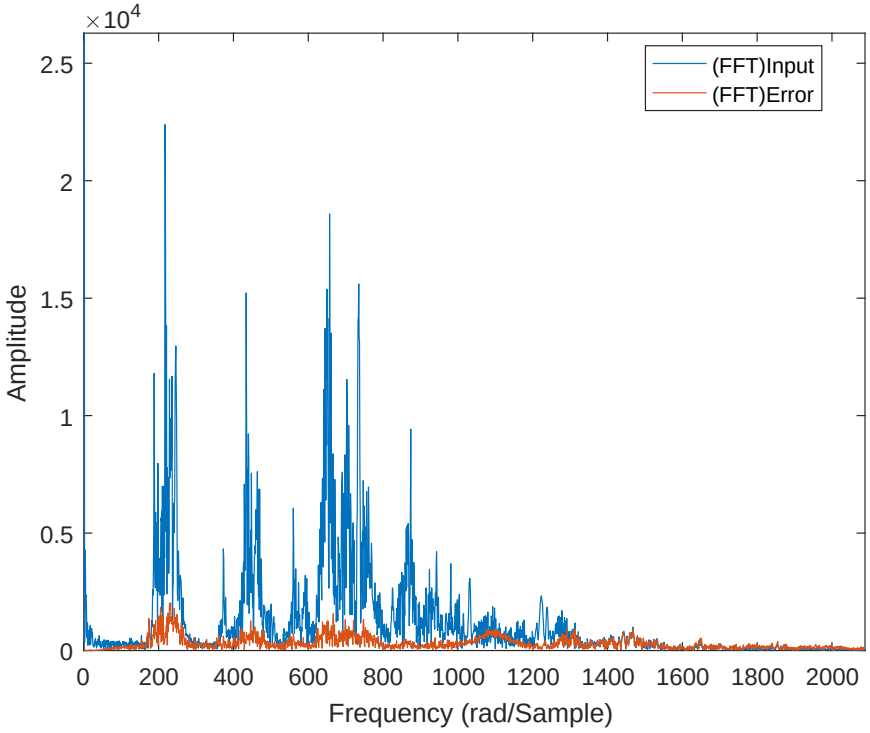

Figure 5.8: Fast Fourier transform of input signal and error in proposed algorithm

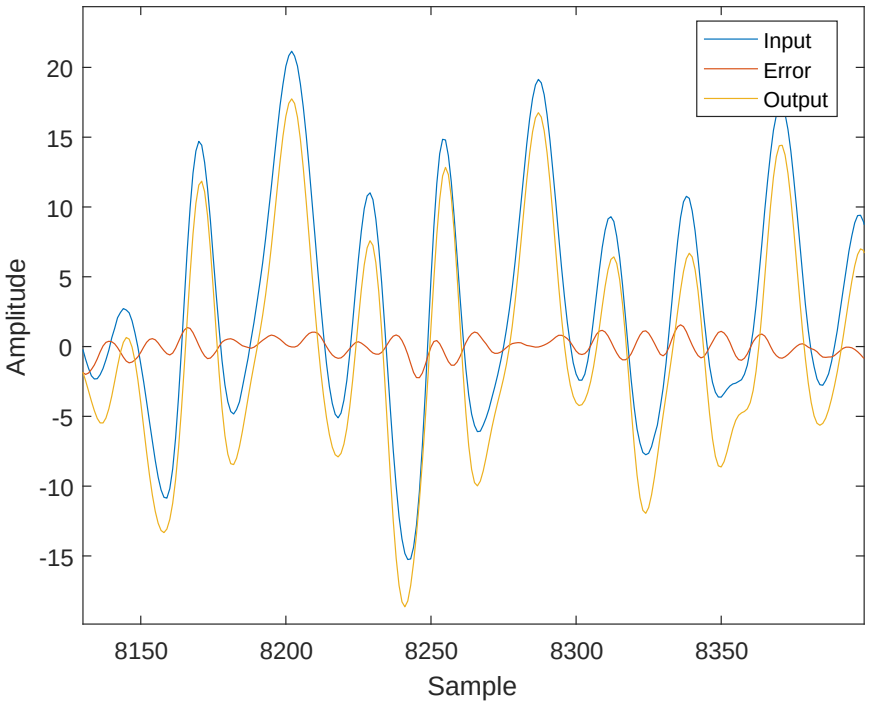

Figure 5.9: Input, output and error in proposed algorithm

5.1.4 Low-pass filter

Here, we use a 2nd-order Chebyshev Type I low-pass filter with a passband frequency of $30 \cdot 2\pi/400$ Hz. The transfer function of this filter is

$$T_{lp}(z) = \frac{0.2664 \cdot (z+1)^2}{z^2 - 0.1198z + 0.3153}$$

The initial tuning function $G(z)$ is

$$G(z) = \frac{0.2664 \cdot (z+1)^2}{z^2 + 0.7915z + 0.3706}$$

5.1.4.1 Parameter calculation

By substituting the band-pass filter with a low-pass filter, the gains can be calculated offline as in Table. 5.2.

Values Of $\bar{\bar{K}}_{pq}$

p	q	$\bar{\bar{K}}_{1pq}$	$\bar{\bar{K}}_{2pq}$
1	1	−0.2086	0.2157
1	2	−0.2197	0.2334
1	3	−0.1778	0.2159
1	4	0.0401	0.0507
1	5	0.5368	−0.5021
2	1	−0.0449	0.0627
2	2	−0.1287	0.1628
2	3	−0.2007	0.2592
2	4	−0.2122	0.3639
2	5	0.2224	−0.1880

Table 5.2: Values of $\bar{\bar{K}}$ in algorithm based on low-pass filter

5.1.4.2 Frequency identification

As we mentioned previously, we modified our proposed algorithm signals tracking by identifying the DC component, and increased the speed of the dominant poles of the nominal closed loop system. The identified frequencies are shown in Fig. 5.10. Compared with 5.4, we can see that this low-pass approach results in a better result. A quantatative comparison is given in the next section.

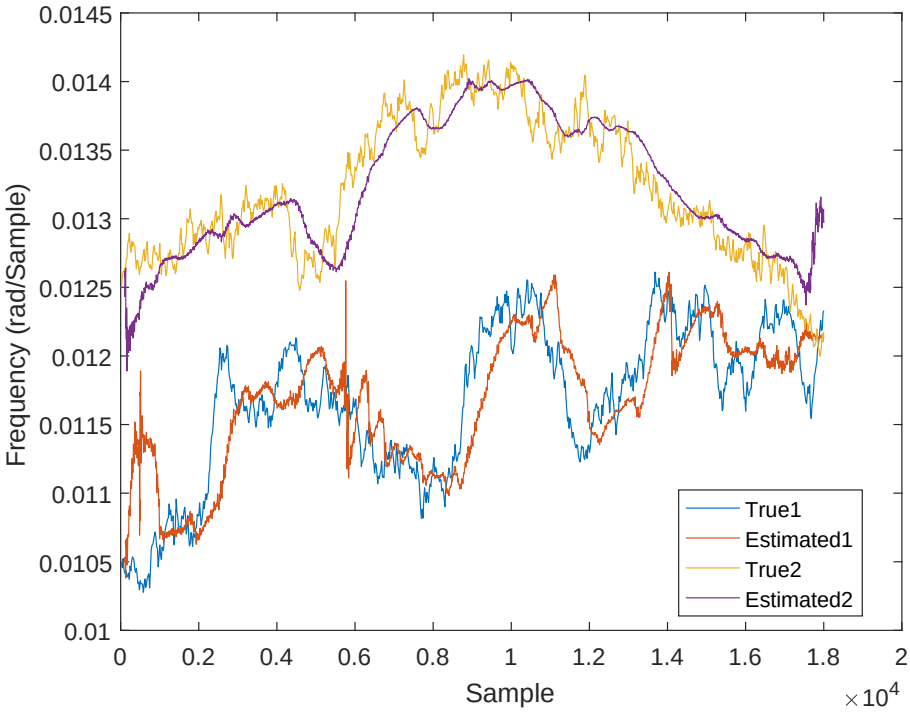

Figure 5.10: Frequency identification for first fundamental component of alternative algorithm

The FFT of the estimated signal and the error is shown in Fig. 5.11 and Fig. 5.12. Since the DC component is identified, the error is much smaller with the proposed tuning algorithm. Also, the performance of the controller system has been enhanced with alternative algorithm.

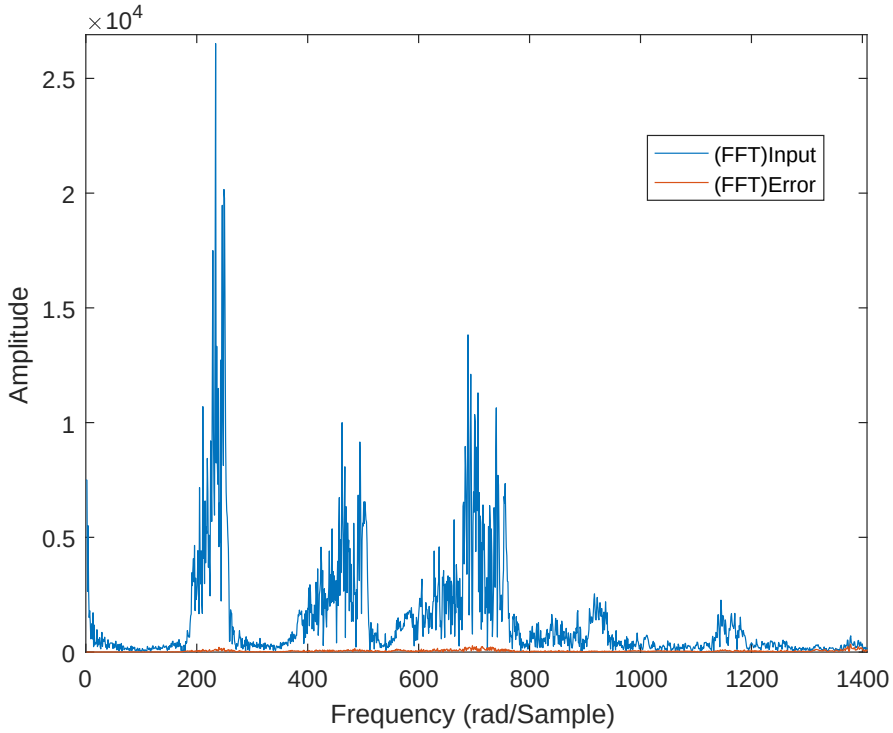

Figure 5.11: Fast Fourier transform of input signal and error in alternative algorithm

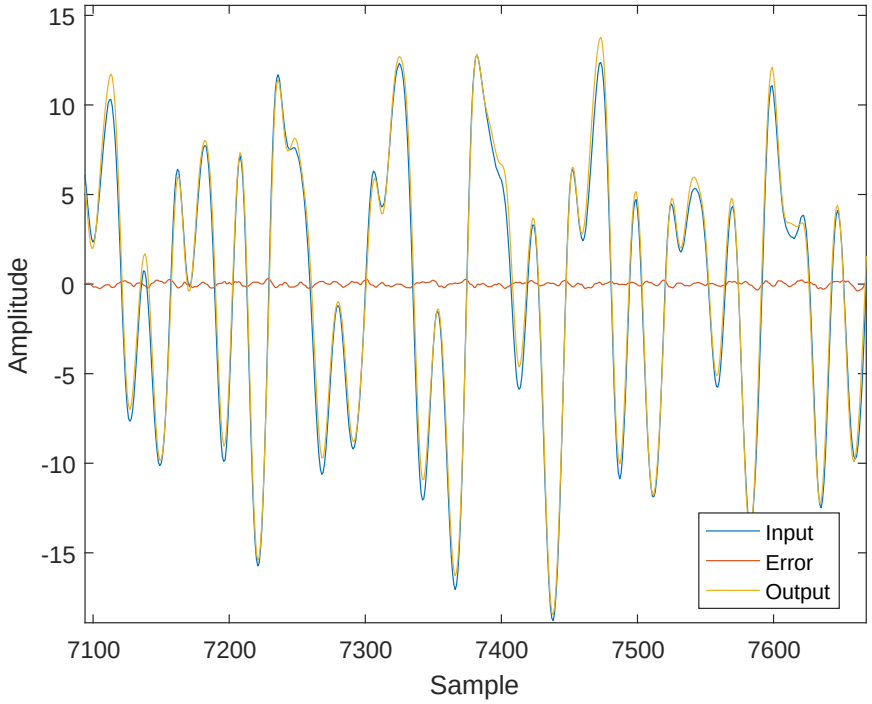

Figure 5.12: Input, output and error in alternative algorithm

Average frequency error in real-time Simulink implementation (rad/sample)

Continuous Time (ode4)	Continuous Time (ode8)	Discrete Time	Alternative Discrete Time
-9.1268×10^{-4}	-9.1174×10^{-4}	2.2122×10^{-4}	-2.1434×10^{-4}

Table 5.3: Comparison of average frequency error among four algorithms

5.1.5 Accuracy comparison

We compare our algorithms with discretized continuous algorithms processing under ode4 and ode8. Fig. 5.13 shows the comparison qualitatively while the Table. 5.3 shows it qualitatively.

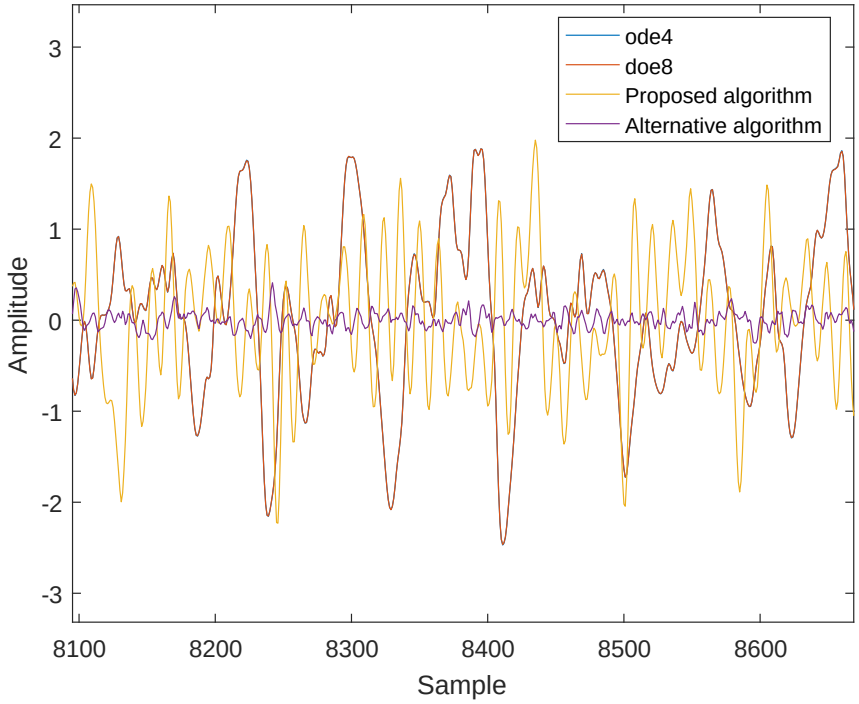

Figure 5.13: Comparison of average frequency error

Ode4 uses Fourth-Order Runge-Kutta (RK4) Formula as its integration technique, which has fourth order of accuracy; while ode8 uses Dormand-Prince RK8(7) Formula which has the highest accuracy of eight in Matlab. Comparing two algorithms in continuous time, although we improve the precision by using ode8, the error only decreased by 0.0094×10^{-4} in our continuous time algorithm. However, the error in discrete time implementation is much smaller than continuous time algorithm even if we compare it with the algorithm using ode8. So, our algorithm implemented in discrete time has better accuracy than previous algorithm.

Maximum sample rate for real-time Simulink implementation (samples/second)

Continuous Time (ode4)	Continuous Time (ode8)	Discrete Time	Alternative Discrete Time
4000	1428	8695	9523

Table 5.4: Comparison of maximum sample rate among four algorithms

5.1.6 Effciency comparison

Table.5.4 shows the comparison among real-time algorithm(using ode8 and ode4), proposed algorithm and alternative algorithm in terms of maximum sample rate. We can compare this table with Table. 5.3, we can see that the accuracy does not improved a lot when we use ode8 instead of ode4, moreover, the speed of the system will tremendously decrease by 65%. Comparing our alternative algorithm with proposed one, the system is accelerated. Most importantly, the impementation in discrete time takes half of the time of continous time algorithm. That is, the implementation in discrete time owes a huge advantage in efficiency compared with implementation in continuous time.

Normally we would be confident that a discrete time algorithm would run much faster than a discretized implementation of a continuous time algorithm. However, in this case, the discrete time algorithm requires evaluation of a significant number of sinusoidal functions each sample time while the continuous time algorithm only involves additions and multiplications. Thus we were in fact uncertainty as to whether the discrete implementation would in fact be faster than the continuous implementation. This may be because of using transcendental function and one more term appearing in the numerator of band-pass filter in discrete time. Computers need more time to process transcendental function comparing the simple functions used in continuous time. Another issue is that the discrete time algorithm is more sensitive to original value than continuous time algorithm. Because the numerical stability of change parameters in a polynomial by a 0.001% causes 10% motion in pole location. In z-domain, 10% to a pole near unit circle is a huge change, but is reletively insignificant in continuous time. So, discrete is more sensitive than continuous system.

5.2 Comparison to Other Algorithms

This algorithm has already been compared to other methods[48, 2, 40, 25] in [53, 32, 54] and found to be superior. As what we discussed in Chapter 2, repetitive control is used in [48] to reject the disturbance of periodic signals with an unknown period. This algorithm cannot cancel the disturbances completely and there exists a big oscillation. The output of the system also contains more disturbances compared to ours. Another comparison is made with AFC. Although it shows a good identification and cancellation, the transfer function of the plant need to be known specifically in advance which is not easy to obtain in practice. And the more complicated the plant is, the more complicated the algorithm will be. When we do the comparison with HHT[32], the HHT completely failed to identify the component frequencies.

5.3 Summary

In this chapter, we have shown that the identification and cancellation are implemented with our algorithm successfully. The overlapping frequencies can be identified by dropping the redundant internal model. By substituting the band-pass filter with low-pass filter, the result becomes better and faster. At last, we made a comparison with real-time implemented algorithm which shows a big benefit of our algorithm over others.

Chapter 6

Conclusions and Future Directions

The implementation of signal identification based on internal model in discrete time is shown in this thesis. First, the background and motivation of our research are introduced. Then, some relative methods to our problem are reviewed and the pros and cons of them are analyzed. These methods are divided to two classics: time-frequency representation-based method and filter-based method. Since our algorithm is based on internal model principle, to illustrate how this principle works, we showed how to use the states of internal model to identify the frequency by using a simple internal model. Then, we introduced previous work which are done by other researchers including the team of Dr.Brown in Chapter 3. These works includes simple algorithm and an alternative algorithm in continuous time. From one internal model to more than one internal models, from off-line tuning to online tuning, we showed the derivation of my algorithm clearly. The implementation of previous work in discrete time is shown in the next chapter. First, we convert the model in real-time to discrete time. Then, we illustrated a better algorithm to calculate parameter a. Although it has the same idea as in continuous time, it becomes much more complex in discrete time. However, since this algorithm can reduce $2n+4$ redundant equations to 4 equations, it is still meaningful to use this algorithm. To get started, we tune the system off-line at the first step and every parameters can be solved by hand. Then we tune the system online. We design the system which has the same transfer function as a desired band-pass filter. Since this system cannot elimi-

nate the DC component, the error between identified frequency and true frequency is relatively big. So, we did a modification of utilizing a low-pass filter instead of the band-pass filter in the next section. As what we showed in Chapter 5, our algorithm can identify the frequencies successfully and alternative algorithm also improved the performance. Comparing with continuous algorithm, our algorithm is not only more effective but also more accurate.

6.1 Conclusion

This thesis has shown the instantaneous Fourier decomposition algorithm that is based on the orthogonal state variables of an internal model principle controller. The usage of the adaptive internal model principle controller is to provide information of the frequency that we need to identify. First, we have shown how to use these orthogonal state variables to identify the signal frequencies. To ensure stability, we must solve a large set, greater than 2 times the number harmonics present, of linear equations. Further, the coefficients of these linear equations are computationally expensive to calculate. Previous work showed how to reduce this problem to a problem of solving pairs of linear equations by utilizing the rule of expansion of polynomial and one set of 4 coupled linearly independent equations. This work showed how to find explicit solutions to these last 4 parameters. This work demonstrated that these solutions methods were numerically stable. This was not known to be true prior to this work. The other main contribution of this thesis is implementing this algorithm in discrete time which saves a lot of running time. The simulation result indicates the validity of our algorithm and the ability to identify the frequencies with uncertain periodic disturbances with good convergence time speed. This Matlab impementation was able to work at 8 kHz, so it should not be difficult to increase that to 20 kHz which satisfies our original requirement.

6.2 Future Work

Although we have successfully implemented our algorithm in discrete time and shown good identification and tracking of the signals, there are still some tasks to work on. Our future work is

to improve the speed of convergence as much as possible and reduce the sensitivity of algorithm. Hopefully, we will implement our algorithm on fixed point hardware in the future. It is also needed to investigate the role of the ε_{ij} design parameters and what type of trade of between accuracy and tracking speed occurs when these parameters are selected.

Bibliography

[1] Marc Bodson and Scott C Douglas. Adaptive algorithms for the rejection of sinusoidal disturbances with unknown frequency. *IFAC Proceedings Volumes*, 29(1):5168–5173, 1996.

[2] Marc Bodson, Alexei Sacks, and Pradeep Khosla. Harmonic generation in adaptive feedforward cancellation schemes. *IEEE Transactions on Automatic control*, 39(9):1939–1944, 1994.

[3] Ronald Newbold Bracewell and Ronald N Bracewell. *The Fourier transform and its applications*, volume 31999. McGraw-Hill New York, 1986.

[4] Harold L Broberg and Richard G Molyet. Correction of periodic errors in a weather satellite servo using repetitive control. In *Control Applications, 1992., First IEEE Conference on*, pages 682–683. IEEE, 1992.

[5] Lyndon J Brown and Jin Lu. Internal model based adaptive algorithm for noise cancellation. In *Proceeding of the 12th International Conference on Sound and Vibration*, volume 5, pages 5544–5548, 2005.

[6] Lyndon J Brown and Qing Zhang. Identification of periodic signals with uncertain frequency. *IEEE Transactions on Signal Processing*, 51(6):1538–1545, 2003.

[7] Lyndon J Brown and Qing Zhang. Periodic disturbance cancellation with uncertain frequency. *Automatica*, 40(4):631–637, 2004.

[8] Kok Kia Chew and Masayoshi Tomizuka. Digital control of repetitive errors in disk drive systems. In *American Control Conference, 1989*, pages 540–548. IEEE, 1989.

[9] C Cosner, G Anwar, and M Tomizuka. Plug in repetitive control for industrial robotic manipulators. In *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on*, pages 1970–1975. IEEE, 1990.

[10] Ryan Deering. Fine-scale analysis of speech using empirical mode decomposition: insight and applications. 2006.

[11] Ryan Deering and James F Kaiser. The use of a masking signal to improve empirical mode decomposition. In *Acoustics, Speech, and Signal Processing, 2005. Proceedings.(ICASSP'05). IEEE International Conference on*, volume 4, pages iv–485. IEEE, 2005.

[12] Bruce A Francis and W Murray Wonham. The internal model principle of control theory. *Automatica*, 12(5):457–465, 1976.

[13] Xian-He Gao and Liang Tao. Gabor time–frequency representation for transient signals using multiwindow discrete gabor transform. *International Journal of Wavelets, Multiresolution and Information Processing*, 15(04):1750036, 2017.

[14] Lin Guo. Reducing the manufacturing costs associated with hard disk drives-a new disturbance rejection control scheme. *IEEE/ASME Transactions on Mechatronics*, 2(2):77–85, 1997.

[15] Shinji Hara, Yutaka Yamamoto, Tohru Omata, and Michio Nakano. Repetitive control system: A new type servo system for periodic exogenous signals. *IEEE Transactions on automatic control*, 33(7):659–668, 1988.

[16] AD Hoover, Valentina Kouznetsova, and Michael Goldbaum. Locating blood vessels in retinal images by piecewise threshold probing of a matched filter response. *IEEE Transactions on Medical imaging*, 19(3):203–210, 2000.

[17] Liu Hsu, Romeo Ortega, and Gilney Damm. A globally convergent frequency estimator. *IEEE Transactions on Automatic Control*, 44(4):698–713, 1999.

[18] Norden E Huang, Zheng Shen, and Steven R Long. A new view of nonlinear water waves: the hilbert spectrum. *Annual review of fluid mechanics*, 31(1):417–457, 1999.

[19] 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 *Proceedings of the Royal Society of London A: mathematical, physical and engineering sciences*, volume 454, pages 903–995. The Royal Society, 1998.

[20] Tian-Li Huang, Meng-Lin Lou, Hua-Peng Chen, and Ning-Bo Wang. An orthogonal hilbert-huang transform and its application in the spectral representation of earthquake accelerograms. *Soil Dynamics and Earthquake Engineering*, 104:378–389, 2018.

[21] Xiaoyi Jiang and Daniel Mojon. Adaptive local thresholding by verification-based multithreshold probing with application to vessel detection in retinal images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(1):131–137, 2003.

[22] Zhao Jin-Ping and Huang Da-ji. Mirror extending and circular spline function for empirical mode decomposition method. *Journal of Zhejiang University-Science A*, 2(3):247–252, 2001.

[23] Josef Kokes and Nghien Nguyen. Using constrained cubic spline instead of natural cubic spline to eliminate overshoot and undershoot in hht. *Annals of the Faculty of Engineering Hunedoara*, 9(3):23, 2011.

[24] Prabha Kundur, Neal J Balu, and Mark G Lauby. *Power system stability and control*, volume 7. McGraw-hill New York, 1994.

[25] Henry K Kwok and Douglas L Jones. Improved instantaneous frequency estimation using an adaptive short-time fourier transform. *IEEE transactions on signal processing*, 48(10):2964–2972, 2000.

[26] Dina Shona Laila, Arturo Roman Messina, and Bikash C Pal. A refined hilbert–huang transform with applications to interarea oscillation monitoring. *IEEE Transactions on Power Systems*, 24(2):610–620, 2009.

[27] Zhichao Liu, Bin Zhang, Keliang Zhou, and Jingcheng Wang. Virtual variable sampling discrete fourier transform based selective odd-order harmonic repetitive control of dc/ac converters. *IEEE Transactions on Power Electronics*, 33(7):6444–6452, 2018.

[28] Yan Ma. *An Iterative Approach to Automatic Music Transcription and Audio Signal Decomposition*. 2010.

[29] Stéphane Mallat. *A wavelet tour of signal processing*. Academic press, 1999.

[30] William Messner and Marc Bodson. Design of adaptive feedforward controllers using internal model equivalence. In *American Control Conference, 1994*, volume 2, pages 1619–1623. IEEE, 1994.

[31] Yves Meyer. *Wavelets and operators*, volume 1. Cambridge university press, 1995.

[32] Edris Saleh Mohsen. Realtime implementation of an internal-model-principle signal identifier. Master's thesis, 2017.

[33] Chen J Mohsen E, Brown LJ. A real time alternative to the hilbert huang transform based on internal model principle. *Journal of Electrical & Electronic Systems*, 6(2):1–6, 2017.

[34] Y Morales, R Nuñez, J Suarez, and C Torres. Digital tool for detecting diabetic retinopathy in retinography image using gabor transform. In *Journal of Physics: Conference Series*, volume 792, page 012083. IOP Publishing, 2017.

[35] Jean Morlet, G Arens, E Fourgeau, and D Glard. Wave propagation and sampling theoryâpart i: Complex signal and scattering in multilayered media. *Geophysics*, 47(2):203–221, 1982.

[36] M Nagashima, K Usui, and M Kobayashi. Rejection of unknown periodic disturbances in magnetic hard disk drives. *IEEE Transactions on Magnetics*, 43(9):3774–3778, 2007.

[37] Gilberto Pin. A direct approach for the frequency-adaptive feedforward cancellation of harmonic disturbances. *IEEE Transactions on Signal Processing*, 58(7):3523–3530, 2010.

[38] Zhang Qingjie, Zhu Huayong, and Shen Lincheng. A new method for mitigation of end effect in empirical mode decomposition. In *Informatics in Control, Automation and Robotics (CAR), 2010 2nd International Asia Conference on*, volume 1, pages 400–403. IEEE, 2010.

[39] Lawrence R Rabiner and Ronald W Schafer. *Digital processing of speech signals*, volume 100. Prentice-hall Englewood Cliffs, NJ, 1978.

[40] Phillip A Regalia. An improved lattice-based adaptive iir notch filter. *IEEE transactions on signal processing*, 39(9):2124–2128, 1991.

[41] Daniel Ruiz-Vega, Arturo R Messina, and Gilberto Enríquez-Harper. Analysis of interarea oscillations via non-linear time series analysis techniques. In *Proc. 15th Power Systems Computation Conf*, 2005.

[42] N Senroy and S Suryanarayanan. Two techniques to enhance empirical mode decomposition for power quality applications. In *Power Engineering Society General Meeting, 2007. IEEE*, pages 1–6. IEEE, 2007.

[43] Nilanjan Senroy, Siddharth Suryanarayanan, and Paulo F Ribeiro. An improved hilbert–huang method for analysis of time-varying waveforms in power quality. *IEEE Transactions on Power Systems*, 22(4):1843–1850, 2007.

[44] Yunpeng Su, Xiaoping Zheng, Xiaojiao Deng, and Yuqiang Deng. Terahertz spectral fingerprints detection with hilbert-huang transform. In *Infrared, Millimeter, and Terahertz Waves (IRMMW-THz), 2017 42nd International Conference on*, pages 1–2. IEEE, 2017.

[45] Yujuan Sun. Instantaneous fourier series estimation. *Mathesis*, 6:54 – 56, 2006.

[46] Mitsuo Takeda, Hideki Ina, and Seiji Kobayashi. Fourier-transform method of fringe-pattern analysis for computer-based topography and interferometry. *JosA*, 72(1):156–160, 1982.

[47] Tsu-Chin Tsao and Masayoshi Tomizuka. Robust adaptive and repetitive digital tracking control and application to a hydraulic servo for noncircular machining. *Journal of dynamic systems, measurement, and control*, 116(1):24–32, 1994.

[48] Tsu-Cliin Tsao, Yao-Xin Qian, and Mahadevamurty Nemani. Repetitive control for asymptotic tracking of periodic signals with an unknown period. *Urbana*, 51:61801, 2000.

[49] Zhaohua Wu and Norden E Huang. Ensemble empirical mode decomposition: a noise-assisted data analysis method. *Advances in adaptive data analysis*, 1(01):1–41, 2009.

[50] Pan Yu, Min Wu, Jinhua She, Kang-Zhi Liu, and Yosuke Nakanishi. An improved equivalent-input-disturbance approach for repetitive control system with state delay and disturbance. *IEEE Transactions on Industrial Electronics*, 65(1):521–531, 2018.

[51] Zhe Yu, Di Shi, Haifeng Li, Yishen Wang, Zhehan Yi, and Zhiwei Wang. An extended kalman filter enhanced hilbert-huang transform in oscillation detection. *arXiv preprint arXiv:1711.04644*, 2017.

[52] Gulan Zhang. Time-phase amplitude spectra based on a modified short-time fourier transform. *Geophysical Prospecting*, 66(1):34–46, 2018.

[53] Qing Zhang. Periodic disturbance cancellation with uncertain frequency. Master's thesis, 2001.

[54] Qing Zhang. *Time-Varing Frequency Estimation and Periodic Disturbance Cancellation*. PhD thesis, 2004.

[55] Qing Zhang and LJ Brown. Designing of adaptive bandpass filter with adjustable notch for frequency demodulation. In *American Control Conference, 2003. Proceedings of the 2003*, volume 4, pages 2931–2936. IEEE, 2003.

[56] Zhenyu Zhao and Lyndon Brown. Fast estimation of power system frequency using adaptive internal-model control technique. In *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, volume 1, pages 845–850. IEEE, 2004.

[57] Zhenyu Zhao and Lyndon J Brown. Musical pitch tracking using internal model control based frequency cancellation. In *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, volume 5, pages 5544–5548. IEEE, 2003.

Appendix A

Proposed algorithm Matlab code

The mat-lab code contain two main parts .First is (IFD), following this code indicates the design of Chebyshev second order band-pass filter T_{bp}, desired band-pass filter with notches T_{bpn}, tuning function $G(s)$and closed loop transfer function T_{de}. As a result, the state space variables gain K_p for each $IM_{i,j}$ and the parameters b_1,a_1, a_2, a_3, and a_4 for the tuning function are computed based on the desired estimation system behaves such as a band-pass filter with notches.Second part is the S-function which is indicate the states variables derivation $\dot{x}_{1ij}$ and $\dot{x}_{2ij}$ for the system, initial conditions for the states variables and estimated frequencies $j\hat{\omega}_1$and $j\hat{\omega}_2$ for each internal model $IM_{i,j}$. As well as the number of the inputs m, adaptive gain $K_a$and number of notches n.

A.1 IFD(bandpass)

//FUNCTION [kp_2bar,kp_bar,kp,a,b]=discretetime(ω,E)

$\omega_0 = \omega$;

$E_0 = E$;

//Design Chebyshev second order bandpass filter

// Read the input (ω, ϵ)

$\omega_H = 30 * 2 * pi/400$;

$\omega_L = 1 * 2 * pi/400$;

$[ch_b, ch_a] = cheby1(2, 1, [\ \omega_H \quad \omega_L\ ])$;

//Identify the variables

C0 = ch_a(1);

C1 = ch_a(2);

C2 = ch_a(3);

C3 = ch_a(4);

C4 = ch_a(5);

b1 = ch_b(1);

b3 = ch_b(3);

b5 = ch_b(5);

M = length(ω);

kp_2bar=ones(2*M,1);

kp_bar=ones(2*M,1);

kp=ones(2*M,1);

kpkp_2bar=ones(2*M,1)*j;

counter=0;

kept=zeros(1,M-1);

drop=zeros(1,M-1);

indx=zeros;

//FOR i=1:floor(M/2)

// Find equal or adjacent frequencies

indx=find(ω(i+1:end)>(ω(i)*(1-.001)) & ω(i+1:end)<(ω(i)*(1+.001)));

// IF true

//IF length(indx)

// increment the counter by one

counter=counter+1;

// Drop the second matched one and save it in a new variable

Dro=indx(1)+i+counter-1;

drop(counter)=Dro;

// Update the frequencies

ω(1:end-counter)=w([1:indx(1)+i-1 Dro-counter+2:end-counter+1]);

// Update the epsilons

E(1:end-counter)=E([1:indx(1)+i-1 Dro-counter+2:end-counter+1]);

// Decrement the length of input by one

M=M-1;

// Save the first matched frequency in new variable

kept(counter)=i;

//END IF

//END FOR

den_bp0 = [$C0$ $C1$ $C2$ $C3$ $C4$];

//FOR k=1:M

//Return the value of a polynomial bandpass filter denominator at ($j * \omega$)

den_bpn0_1 = C0*(exp($j * \omega(k)$))^4 + C1*(exp$j * \omega(k)$))^3 + C2*(exp($j * \omega(k)$))^2 + C3*(exp($j * \omega(k)$)) + C4;

//Substitute notch filter denominator by ($j * \omega$)

rho=exp($-E.*\omega$);

den__bpn_1 = (exp($j * \omega(k)$))^2 - 2*rho.*cos(ω).*(exp($j * \omega(k)$)) + rho.^2;

//Calculate the bandpass filter with notches by multiplying the bandpass filter with all of the notches

den_bpn_1 = den_bpn0_1*prod(den__bpn_1);

//Calculate the parameters from T_{de} to get Ks

para_de_1_1 = (exp($j * \omega(k)$) - exp($j * \omega$)).*(exp($j * \omega(k)$) - exp($-j * \omega$));

para_de_1_1(para_de_1_1==0) = 1;

para_de_1 = b1 * ((exp($j * \omega(k)$))^4 - 2*(exp($j * \omega(k)$))^2 + 1)*prod(para_de_1_1);

//Return the value of a polynomial bandpass filter denominator at $(-j*\omega)$

den_bpn0_2 = C0*(exp($-j*\omega(k)$))^4 + C1*(exp($-j*\omega(k)$))^3 + C2*(exp($-j*\omega(k)$))^2 + C3*(exp($-j*\omega(k)$)) + C4;

//Substitute notch filter denominator by $(-j*\omega)$

rho=exp($-E.*\omega$);

den__bpn_2 = (exp($-j*\omega(k)$))^2 - 2*rho.*cos(ω).*(exp($-j*\omega(k)$)) + rho.^2;

//Calculate the bandpass filter with notches by multiplying the bandpass filter with all of the notches

den_bpn_2 = den_bpn0_1*prod(den__bpn_1);

// Calculate the parameters from T_{de} to get Ks

para_de_1_2 = (exp($j*\omega(k)$) - exp($j*\omega$)).*(exp($j*\omega(k)$) - exp($-j*\omega$));

para_de_1_2(para_de_1_2==0) = 1;

para_de_2 = b1 * ((exp($-j*\omega(k)$))^4 - 2*(exp($-j*\omega(k)$))^2 + 1)*prod(para_de_1_2);

//Calculate kpkp_2bar

kpkp_2bar(2*k) = (den_bpn_2*para_de_1 - den_bpn_1*para_de_2)/(para_de_1*para_de_2*(exp($-j*\omega(k)$) - exp$j*\omega(k)$)));

kpkp_2bar(2*k-1) = (den_bpn_2-para_de_2*kpkp_2bar(2*k)*exp($-j*\omega(k)$))/para_de_2;

//Eliminate the calculation error of imaginary part

//IF abs(imag(kpkp_2bar(2*k)))<=10^-10

kp_2bar(2*k) = real(kpkp_2bar(2*k));

//END IF

//IF abs(imag(kpkp_2bar(2*k-1)))<=10^-10

kp_2bar(2*k-1) = real(kpkp_2bar(2*k-1));

//END IF

//Calculate the kp and kp_bar

kp(2*k) = ((kp_2bar(2*k)-kp_2bar(2*k-1))*$\omega(k)$)/(2*sin($\omega(k)$));

kp(2*k-1) = ((kp_2bar(2*k)+ kp_2bar(2*k-1))*$\omega(k)$)/(2*(1-cos($\omega(k)$)));

kp_bar(2*k) = kp(2*k)*sin($\omega(k)$)/$\omega(k)$ + kp(2*k-1)*(1-cos($\omega(k)$))/$\omega(k)$;

kp_bar(2*k-1) = kp(2*k-1)*sin($\omega(k)$)/$\omega(k)$ - kp(2*k)*(1-cos($\omega(k)$))/$\omega(k)$;

//END FOR

//IF the counter is true then

//Dropping an internal model $IM_{l;k}$ from the design stage

//FOR i=1:counter

kp_2bar(2*kept(i)-1:2*kept(i))=kp_2bar(2*kept(i)-1:2*kept(i))/2;

kp_2bar(2*drop(i)-1:end)=[kp_2bar(2*kept(i)-1:2*kept(i))/2 kp_2bar(2*drop(i)-1:end-2)];

//END FOR

//IF the counter is true then

//Dropping an internal model $IM_{l;k}$ from the design stage

FOR i=1:counter

kp_bar(2*kept(i)-1:2*kept(i))=kp_bar(2*kept(i)-1:2*kept(i))/2;

kp_bar(2*drop(i)-1:end)=[kp_bar(2*kept(i)-1:2*kept(i))/2 kp_bar(2*drop(i)-1:end-2)];

END FOR

//IF the counter is true then

//Dropping an internal model $IM_{l;k}$ from the design stage

FOR i=1:counter

kp(2*kept(i)-1:2*kept(i))=kp(2*kept(i)-1:2*kept(i))/2;

kp(2*drop(i)-1:end)=[kp(2*kept(i)-1:2*kept(i))/2 kp(2*drop(i)-1:end-2)];

END FOR

//END IF

// Computing the a parameter for the tuning function $G(s)$

ω_2eb=-2*cos(ω_0);

a2_4eb=ω_2eb(1:9).*cumsum(ω_2eb(2:end),'reverse');

ω_2eb1=-2*exp($-E_0 * \omega_0$).*cos(ω_0);

a2_4eb1=ω_2eb1(1:9).*cumsum(ω_2eb1(2:end),'reverse');

```
a4 = C4*prod(exp(-2*E_0 * ω_0)) - b1*sum(kp_2bar(1:2:end));
a3 = C3*prod(exp(-2*E_0 * ω_0)) + C4*sum(((-2*exp(-E_0 * ω_0).*cos(ω_0))*prod(exp(-2*E_0 * ω_0))).
/(exp(-2*E_0*ω_0))) - a4*sum(-2*cos(ω_0)) - b1*(sum(kp_2bar(2:2:end)) + sum(kp_2bar(1:2:end).*(sum(-
2*cos(ω_0))- (-2*cos(ω_0)))));
a1 = C1 + C0*sum(-2*exp(-E_0*ω_0).*cos(ω_0)) - sum(-2*cos(ω_0)) - b1*sum(kp_2bar(2:2:end));
a2 = C0*(sum(exp(-2*E_0 * ω_0))+sum(a2_4eb1)) + C1*sum(-2*exp(-E_0 * ω_0).*cos(ω_0)) + C2 -
(10+sum(a2_4eb)) - a1*sum(-2*cos(ω_0)) - b1*(sum(kp_2bar(2:2:end).*(sum(-2*cos(ω_0))- (-2*cos(ω_0))))
+ sum(kp_2bar(1:2:end)));
a = [a1 a2 a3 a4];
b = [b1 b3 b5];
//END FUNCTION
```

A.2 S-Function(bandpass)

```
FUNTION [sys,x0,str,ts] = IFD_dis_1(t,x,u,flag,Ka,IN,n,m)
    %%%Initialize values%%%
    //Switch Flag
    //Case 0
        //IF
            length(n) ~=1 & length(n)~=m
            error('number of IM_{i,j} inconsistent')
        //END IF
        s = simsizes ;
        s.NumContStates = 0;
        s.NumDiscStates = 2*sum(n)+m ;
        s.NumOutputs = 2*sum(n)+m;
        s.NumInputs = 2*sum(n)+1 ;
        s.DirFeedthrough = 0 ;
        s.NumSampleTimes = 1 ;
        sys = simsizes(s) ;
        x0 = IN;
        str = [];
        ts = [ 1   0 ];
    //States variables Derivation
    //Case 2
        e = u(1);
        n1 = 0;
        n2 = 2*sum(n);
    //FOR i=1:m;
    //FOR j=2:2:2*n(i);
```

```
    k1 = u(n1+j)
    k2 = u(n1+j+1);
    ω = (j/2)*x(n2+i);
    dx(n1+j-1) = cos(ω)*x(n1+j-1) - sin(ω)*x(n1+j) + (k1*sin(ω)/ω - k2*(1-cos(ω))/ω)*e;
    dx(n1+j) = sin(ω)*x(n1+j-1) + cos(ω)*x(n1+j) + (k2*sin(ω)/ω + k1*(1-cos(ω))/ω)*e;
//END FOR
//Applying the adaptive gain into estimated frequency
//IF t<100| x(n1+1)^2+x(n1+2)^2<.00004;
    dx(n2+i+1) = x(n2+i);
//ELSE
    dx(n2+i+1) = x(n2+i)+Ka(i)*(x(n1+1)*u(n1+3)-x(n1+2)*u(n1+2))*e/(x(n1+1)^2+x(n1+2)^2);
    n1 = n1+2*n(i);
//END IF
//END FOR
    sys = dx;
//CASE 3,
    sys = x;
//CASE {9}
    sys = []
//Otherwise
    error(['Unhandled flag = ',num2str(flag)]);
//END SWITCH CASE
```

Appendix B

Matlab alternative approach code

The Matlab code below shows the alternative approch in discrete time of IFD and SF. Also, the code of using bandpass filter is shown. Thus, the code can be given as

B.1 IFD(lowpass)

//FUNCTION [kp_2bar,kp_bar,kp,a,b]=discretetime(ω,E)

// Read the input (ω, E)

$\omega_0 = \omega$;

$E_0 = E$;

rho = 0.001;

//Design Chebyshev second order lowpass filter

$\omega_H = 30 * 2 * pi/400$;

$\omega_L = 1 * 2 * pi/400$;

$BW = \omega_H - \omega_L$

$[ch_b, ch_a] = cheby1(2, 1, BW)$;

//Identify the variables

b1 = ch_b(1);

b2 = ch_b(2);

b3 = ch_b(3);

C1 = ch_a(2);

C2 = ch_a(3);

M = length(ω);

kp_2bar = ones(2*M,1);

kp_bar = ones(2*M,1);

kp = ones(2*M,1);

kpkp_2bar = ones(2*M,1)*j;

counter = 0;

kept = zeros(1,M-1);

drop = zeros(1,M-1);

indx = zeros;

//FOR i = 1:floor(M/2)

```
// Find equal or adjacent frequencies
    indx = find(ω(i+1:end)>(ω(i)*(1-.001)) & ω(i+1:end)<(ω(i)*(1+.001)));
// IF true
    //IF length(indx)
// increment the counter by one
        counter = counter+1;
// Drop the second matched one and save it in a new variable
        Dro = indx(1)+i+counter-1;
        drop(counter) = Dro;
// Update the frequencies
        ω(1:end-counter) = ω([1:indx(1)+i-1 Dro-counter+2:end-counter+1]);
// Update the epsilons
        E(1:end-counter) = E([1:indx(1)+i-1 Dro-counter+2:end-counter+1]);
// Decrement the length of input by one
        M = M-1;
// Save the first matched frequency in new variable
        kept(counter) = i;
    //END IF
//END FOR
    den_bp0 = [ 1   C1   C2 ];
//FOR k=1:M
//Return the value of a polynomial bandpass filter denominator at (j * ω)
    den_bpn0_1 = (exp(j * ω(k)))^2 + C1*(exp(j * ω(k))) + C2;
//Substitute notch filter denominator by (j * ω)
    rho=exp(−E.*ω);
    den__bpn_1 = (exp(j * ω(k)))^2 - 2*rho1.*cos(w).*(exp(j * ω(k))) + rho1.^2; %1*10
//Calculate the bandpass filter with notches by multiplying the bandpass filter with all of the
```

notches

den_bpn_1 = den_bpn0_1*prod(den__bpn_1)*(exp($j * \omega(k)$)-rho);

//Calculate the bandpass filter with notches by multiplying the bandpass filter with all of the notches

den_bpn_1 = den_bpn0_1*prod(den__bpn_1)*(exp(j*w(k))-rho);

//Calculate the parameters from T_{de} to get Ks

para_de_1_1 = (exp($j * \omega(k)$) - exp($j * \omega$)).*(exp($j * \omega(k)$) - exp($-j * \omega$));

para_de_1_1(para_de_1_1==0) = 1;

para_de_1 = b1 * (exp($j * \omega(k)$) + 1)^2*(exp$j * \omega(k)$) - 1)*prod(para_de_1_1);

//Return the value of a polynomial bandpass filter denominator at $(-j * \omega)$

den_bpn0_2 = (exp(-$j * \omega(k)$))^2 + C1*(exp(-$j * \omega(k)$)) + C2;

//Substitute notch filter denominator by $(-j * \omega)$

rho=exp($-E.*\omega$);

den__bpn_2 = (exp(-$j * \omega(k)$))^2 - 2*rho2.*cos(w).*(exp(-$j * \omega(k)$)) + rho2.^2;

//Calculate the bandpass filter with notches by multiplying the bandpass filter with all of the notches

den_bpn_2 = den_bpn0_2*prod(den__bpn_2)*(exp(-$j * \omega(k)$)-rho);

// Calculate the parameters from T_{de} to get Ks

para_de_1_2 = (exp(-$j * \omega(k)$) - exp$j * \omega$)).*(exp(-$j * \omega(k)$) - exp(-$j * \omega$));

para_de_1_2(para_de_1_2==0) = 1;

para_de_2 = b1 * (exp(-$j * \omega(k)$) + 1)^2*(exp(-$j * \omega(k)$) - 1)*prod(para_de_1_2);

//Calculate kpkp_2bar

kpkp_2bar(2*k) = ((den_bpn_1/para_de_1) - (den_bpn_2/para_de_2))/(exp($j*\omega(k)$) - exp(-$j * \omega(k)$));

kpkp_2bar(2*k-1) = (den_bpn_1/para_de_1) - kpkp_2bar(2*k)*exp($j * \omega(k)$);

//Eliminate the calculation error of imaginary part

//IF abs(imag(kpkp_2bar(2*k)))<=10^-10

```
    kp_2bar(2*k) = real(kpkp_2bar(2*k));
//END IF
//IF abs(imag(kpkp_2bar(2*k-1)))<=10^-10
kp_2bar(2*k-1) = real(kpkp_2bar(2*k-1));
//END IF
//Calculate the kp and kp_bar
kp(2*k) = ((kp_2bar(2*k)-kp_2bar(2*k-1))*ω(k))/(2*sin(ω(k)));
kp(2*k-1) = ((kp_2bar(2*k)+ kp_2bar(2*k-1))*ω(k))/(2*(1-cos(ω(k))));
kp_bar(2*k) = kp(2*k)*sin(ω(k))/ω(k) + kp(2*k-1)*(1-cos(ω(k)))/ω(k);
kp_bar(2*k-1) = kp(2*k-1)*sin(ω(k))/ω(k) - kp(2*k)*(1-cos(ω(k)))/ω(k);
//END FOR
//IF the counter is true then
//Dropping an internal model IM_{l;k} from the design stage
    //FOR i = 1:counter
        kp_2bar(2*kept(i)-1:2*kept(i))=kp_2bar(2*kept(i)-1:2*kept(i))/2;
        kp_2bar(2*drop(i)-1:end)=[kp_2bar(2*kept(i)-1:2*kept(i))/2 kp_2bar(2*drop(i)-1:end-2)];
    //END FOR
//IF the counter is true then
//Dropping an internal model IM_{l;k} from the design stage
    FOR i = 1:counter
        kp_bar(2*kept(i)-1:2*kept(i)) = kp_bar(2*kept(i)-1:2*kept(i))/2;
        kp_bar(2*drop(i)-1:end) = [kp_bar(2*kept(i)-1:2*kept(i))/2 kp_bar(2*drop(i)-1:end-2)];
    END FOR
//IF the counter is true then
//Dropping an internal model IM_{l;k} from the design stage
    FOR i=1:counter
        kp(2*kept(i)-1:2*kept(i)) = kp(2*kept(i)-1:2*kept(i))/2;
```

```
        kp(2*drop(i)-1:end) = [kp(2*kept(i)-1:2*kept(i))/2 kp(2*drop(i)-1:end-2)];
    END FOR
//END IF
    K0_para_1 = 1-2*exp(-E0. * ω0).*cos(ω0)+(exp(-E0. * ω0)).^2;
    K0_para_2 = 1-2*cos(ω0)+1;
    K0 = ((1+C1+C2)*prod(K0_para_1)*(1-rho))/(4*b1*prod(K0_para_2));
// Computing the a parameter for the tuning function G(s)
    a2 = -C2*prod((exp(-E0. * ω0)).^2)*(-rho)-b1*sum(kp_2bar(1:2:end))+b1*K0;
    a1 = -(C1*prod((exp(-E0.*ω0)).^2)*(-rho) + C2*(-rho)*sum(((-2*exp(-E0.*ω0).*cos(ω0))*prod((exp(-
E0.*ω0)).^2))./((exp(-E0.*ω0)).^2)) + C2*prod((exp(-E0.*ω0)).^2)) + a2 - a2*sum(-2*cos(ω0)) +
b1*2*K0 +b1*K0*sum(-2*cos(ω0)) + b1*(-2*sum(kp_2bar(1:2:end)) + sum(kp_2bar(1:2:end)) -
(sum(kp_2bar(2:2:end)) + sum(kp_2bar(1:2:end).*(sum(-2*cos(ω0))- (-2*cos(ω0))))));
    a = [a1 a2];
    b = [b1 b2 b3];
//END FUNCTION
```

B.2 S-Function(lowpass)

```
FUNTION [sys,x0,str,ts] = IFD_dis_1_low_new(t,x,u,flag,Ka,IN,n,m)
    %%%Initialize values%%%
    //Switch Flag
    //Case 0
        //IF
            length(n) ~= 1 & length(n)~=m
            error('number of IMi,j inconsistent')
        //END IF
        s = simsizes ;
```

```
    s.NumContStates = 0;
    s.NumDiscStates = 2*sum(n)+m+1 ;
    s.NumOutputs = 2*sum(n)+m+1;
    s.NumInputs = 2*sum(n)+1+1 ;
    s.DirFeedthrough = 0 ;
    s.NumSampleTimes = 1 ;
    sys = simsizes(s) ;
    x0 = IN;
    str = [];
    ts = [ 1   0 ];
//States variables Derivation
//Case 2
    e = u(1);
    K0 = u(22);
    n1 = 0;
    n2 = 2*sum(n);
//FOR i = 1:m;
//FOR j = 2:2:2*n(i);
    k1 = u(n1+j)
    k2 = u(n1+j+1);
    ω = (j/2)*x(n2+i+1);
    dx(n1+j-1) = cos(ω)*x(n1+j-1) - sin(ω)*x(n1+j) + (k1*sin(ω)/ω - k2*(1-cos(ω))/ω)*e;
    dx(n1+j) = sin(ω)*x(n1+j-1) + cos(ω)*x(n1+j) + (k2*cos(ω)/ω + k1*(1-cos(ω))/ω)*e;
//END FOR
//Applying the adaptive gain into estimated frequency
//IF t<100| x(n1+1)^2+x(n1+2)^2<.00004;
    dx(n2+i+1) = x(n2+i+1);
```

```
//ELSE
    dx(n2+i+1) = x(n2+i+1)+Ka(i)*(x(n1+1)*u(n1+3)-x(n1+2)*u(n1+2))*e/(x(n1+1)^2+x(n1+2)^2);
    n1 = n1+2*n(i);
//END IF
//END FOR
    sys = dx;
//CASE 3,
    sys = x;
//CASE {9}
    sys = []
//Otherwise
    error(['Unhandled flag = ',num2str(flag)]);
//END SWITCH CASE
```

Curriculum Vitae

Name:	Jie Chen
Post-Secondary Education and Degrees:	University of Western Ontario London, ON, Canada 2018 Master of Science in Electrical and Computer Engineering Beijing University of Posts and Telecommunications Beijing, China 2016 Bachelor (Automatics)
Honours and Awards:	Ministry of Higher Education and Scientific Research Scholarship (MHESR) College of Electronic Technology 2008
Related Work Experience:	Graduate Teaching and Research Assistant University of Western Ontario 2016-2018

Publications: E. Mohsen L. J. Brown and J. Chen. A real time alternative to the hilbert huang transform based on internal model principle. J Electr Electron, (Syst 6: 233), July 2017.