



Circuits and Systems

Mekelweg 4,  
2628 CD Delft  
The Netherlands

<http://ens.ewi.tudelft.nl/>

CAS-2015-07

## M.Sc. Thesis

---

# Compressive Power Spectrum Estimation: Further study on Non Uniform sampling and Parametric Approach

Zhang Ruijie

### Abstract

In the recent development of wireless communication several applications, such as spectrum sensing for cognitive radio are only interested in the power spectrum. These applications do not require the reconstruction of the original analog signal. According to the Whittaker-Kotelnikov-Shannon-Nyquist theorem, the sampling rate must be at least twice the maximum frequency present in the signal if we want to recover the signal from its samples. If we estimate the power spectrum directly by using a high-rate analog-to-digital converter, we will find that such high-rate ADCs consume a large amount of power because of its high sampling rate. To reduce the burden on the ADCs, we investigate compressive power spectrum sensing. Since the power spectrum is calculated based on the autocorrelations of the signal, we do not need to recover the signals. This allows a reduction of the sampling rate compared with the Nyquist rate while maintaining perfect power spectrum reconstruction. In this thesis, we study power spectrum estimation of a wide-sense stationary signal. In general, the signal, whose power spectrum is to be estimated, is sampled by multi-coset sampling. The parametric method to estimate the power spectrum of the signal is also evaluated and the study of the performance of non-uniform sampling is explored as well.



# Compressive Power Spectrum Estimation: Further study on Non Uniform sampling and Parametric Approach

---

THESIS

submitted in partial fulfillment of the  
requirements for the degree of

MASTER OF SCIENCE

in

ELECTRICAL ENGINEERING

by

Zhang Ruijie  
born in Shanxi, China

This work was performed in:

Circuits and Systems Group  
Department of Microelectronics & Computer Engineering  
Faculty of Electrical Engineering, Mathematics and Computer Science  
Delft University of Technology



**Delft University of Technology**

Copyright © 2015 Circuits and Systems Group  
All rights reserved.

DELFT UNIVERSITY OF TECHNOLOGY  
DEPARTMENT OF  
MICROELECTRONICS & COMPUTER ENGINEERING

The undersigned hereby certify that they have read and recommend to the Faculty of Electrical Engineering, Mathematics and Computer Science for acceptance a thesis entitled “**Compressive Power Spectrum Estimation: Further study on Non Uniform sampling and Parametric Approach**” by **Zhang Ruijie** in partial fulfillment of the requirements for the degree of **Master of Science**.

Dated: 2015-07-22

Chairman:

---

prof.dr.ir. Alle-Jan van der Veen

Advisor:

---

prof.dr.ir. Geert Leus

Committee Members:

---

dr.ir Bert-Jan Kooij

---

dr.ir.Dyonisius Dony Ariananda



# Abstract

---

In the recent development of wireless communication several applications, such as spectrum sensing for cognitive radio are only interested in the power spectrum. These applications do not require the reconstruction of the original analog signal. According to the Whittaker-Kotelnikov-Shannon-Nyquist theorem, the sampling rate must be at least twice the maximum frequency present in the signal if we want to recover the signal from its samples. If we estimate the power spectrum directly by using a high-rate analog-to-digital converter, we will find that such high-rate ADCs consume a large amount of power because of its high sampling rate. To reduce the burden on the ADCs, we investigate compressive power spectrum sensing. Since the power spectrum is calculated based on the autocorrelations of the signal, we do not need to recover the signals. This allows a reduction of the sampling rate compared with the Nyquist rate while maintaining perfect power spectrum reconstruction. In this thesis, we study power spectrum estimation of a wide-sense stationary signal. In general, the signal, whose power spectrum is to be estimated, is sampled by multi-coset sampling. The parametric method to estimate the power spectrum of the signal is also evaluated and the study of the performance of non-uniform sampling is explored as well.



# Acknowledgments

---

First of all, I would like to thank my thesis supervisor prof.dr.ir.Geert Leus, for giving me the opportunity to work in a challenging and interesting topic, and for his support during the whole thesis. His advice and positive attitude encouraged me to keep on working. When I had discussions with him, his sharpened insight was really impressive for me. He is a nice professor with good temper. I always feel nervous, but he took care of me every time. Thank you for your kindness and patience.

Then I want to express my gratitude to my daily supervisor, Dyonisius Dony Ari-ananda, who has been with me during the whole thesis. He is very nice, friendly and responsible. When I get confused, I can always get advices from him. I appreciate his effort of helping me with my questions, revised my reports and valuable advice on my work. He is patient and full of intelligent ideas. His positive attitude was really impressive for me. He is always working, doing research, reviewing papers. On the one hand I should learn from him, but on the other hand, I really hope that he can enjoy his life with more relaxation and more sport activities. Besides, he gave invaluable advice in life, taught me how to deal with emotional issues. Thank you very much. I would like to thank prof.dr.ir. Alle-Jan van der Veen and dr.ir. Bert-Jan Kooij as well for participating in my thesis defense committee.

Of course, I am very grateful to my family. They raised me to grow up, supported me study abroad. Without their help and support, my experience would not have been possible. When I met problems, they chatted with me via the video call, comforting me and encouraging me. When I felt upset, it is them who cheered me up and told me no worry. Thank you very much.

Finally, I would like to thank my friends in Delft, China and Hongkong, who made my life wonderful. You are too many to list here, but I am really lucky to met all of you.

Zhang Ruijie  
Delft, The Netherlands  
2015-07-22



# Contents

---

<b>Abstract</b>	<b>v</b>
<b>Acknowledgments</b>	<b>vii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Outline and Contributions . . . . .	4
<b>2 Background discussion</b>	<b>7</b>
2.1 Review of Compressive power spectrum estimation . . . . .	7
2.2 Alternative time domain approach of [1] . . . . .	8
<b>3 Comparative analysis on Parametric and Non-parametric estimation</b>	<b>11</b>
3.1 Autoregressive (AR) model parameters estimation . . . . .	11
3.2 Two options in parametric power spectrum estimation . . . . .	11
3.3 Formulation of the experiments . . . . .	12
3.4 Simulation study . . . . .	14
3.5 Conclusion . . . . .	19
<b>4 Parametric Estimation of Power Spectrum using AR model</b>	<b>23</b>
4.1 Implementation of the methods in AR model . . . . .	23
4.1.1 Two estimation methods . . . . .	25
4.1.2 Newton Raphson method . . . . .	26
4.1.3 Nonlinear Least Squares . . . . .	27
4.1.4 Simulation result . . . . .	28
4.2 Implementation of the methods in ARMA model . . . . .	29
4.2.1 Newton Raphson method . . . . .	30
4.2.2 Nonlinear Least Square Error estimation . . . . .	32
4.2.3 Simulation result . . . . .	32
4.3 Conclusion and analysis . . . . .	33
<b>5 Sub-optimal sparse ruler</b>	<b>35</b>
5.1 Attempt to Reduce the Compression . . . . .	35
5.2 Sub Optimal Sparse Ruler Algorithms . . . . .	36
5.2.1 The first algorithm (Algorithm Sparse Ruler-1) . . . . .	37
5.2.2 The second algorithm (Algorithm Sparse Ruler-2) . . . . .	38
5.2.3 The convex optimization based algorithm (Algorithm Sparse Ruler-3) . . . . .	38
5.3 Comparison with other algorithm . . . . .	41
5.3.1 Coprime Sampling [21] . . . . .	41
5.3.2 Two Levels Nested Array [20] . . . . .	41
5.4 Numerical Study . . . . .	42

<b>6</b>	<b>Performance Analysis of Alternative Time Domain Approach in [1] and Cosets Selection</b>	<b>47</b>
6.1	Theoretical Analysis . . . . .	47
6.2	Simulation study . . . . .	51
6.3	Minimal Compression Rate with certain NMSE . . . . .	55
6.4	Simulation Study . . . . .	62
<b>7</b>	<b>Conclusion and Future Works</b>	<b>67</b>
7.1	Conclusion . . . . .	67
7.2	Suggestion for Further Research . . . . .	68
	7.2.1 The optimization of the parameters estimation . . . . .	68
	7.2.2 The best solution of the cosets selection . . . . .	68
<b>A</b>	<b>Appendix</b>	<b>69</b>
A.1	RELAXATION AND CONVEX PROOF OF CHAPTER 6 . . . . .	69

# List of Figures

---

1.1	The NTIA frequency allocation chart [18] . . . . .	2
1.2	The nature of spectrum holes in the television bands [24] . . . . .	2
1.3	Main sensing methods in terms of their sensing accuracies and complexities	3
2.1	Digital interpretation of the sampling device with a bank of $M$ branches, where each branch consists of a digital filtering operation followed by a down-sampling operation [1] . . . . .	9
3.1	The autocorrelations picked to estimate parameters . . . . .	12
3.2	Experiment I, Case 1 . . . . .	15
3.3	Experiment I, Case 2 . . . . .	16
3.4	Experiment I, Case 3 . . . . .	17
3.5	Experiment I, Case 4 . . . . .	18
3.6	Experiment II, Case 1 . . . . .	19
3.7	Experiment II, Case 2 . . . . .	20
3.8	Experiment II, Case 3 . . . . .	20
3.9	Experiment II, Case 4 . . . . .	21
4.1	Comparison of power spectrums in AR model . . . . .	29
4.2	Comparison of power spectrums in ARMA model . . . . .	33
5.1	Illustration of sum of the diagonals in matrix $\mathbf{V}$ . . . . .	39
5.2	Performance of two sub optimal sparse ruler algorithms . . . . .	42
5.3	Comparison of sparse rulers for covering 10 correlation lags . . . . .	44
5.4	Comparison of sparse rulers for covering 19 correlation lags . . . . .	44
5.5	Comparison of sparse rulers for covering 27 correlation lags . . . . .	45
6.1	Distribution of indices differences in Case 1 . . . . .	52
6.2	Distribution of indices differences in Case 2 . . . . .	52
6.3	Distribution of indices differences in Case 3 . . . . .	53
6.4	Distribution of indices differences in Case 4 . . . . .	53
6.5	Theoretical and simulated NMSE for all the four cases . . . . .	54
6.6	Theoretical and simulated NMSE for all the four cases in dB . . . . .	55
6.7	Power spectrum of the WGN passed through low-pass filter . . . . .	56
6.8	Power spectrum of the WGN passed through high-pass filter . . . . .	57
6.9	Power spectrum of the WGN passed through bandpass filter . . . . .	58
6.10	Power spectrum of the WGN passed through multi-bandpass filter . . . . .	59
6.11	NMSE comparison of WGN signal passed through low-pass filter . . . . .	59
6.12	NMSE comparison of WGN signal passed through low-pass filter (in dB)	60
6.13	NMSE comparison of WGN signal passed through high-pass filter . . . . .	60
6.14	NMSE comparison of WGN signal passed through high-pass filter (in dB)	61
6.15	NMSE comparison of WGN signal passed through bandpass filter . . . . .	61
6.16	NMSE comparison of WGN signal passed through bandpass filter (in dB)	62

6.17	NMSE comparison of WGN signal passed through multi-bandpass filter	62
6.18	NMSE comparison of WGN signal passed through multi-bandpass filter (in dB) . . . . .	63
6.19	The distribution of the correlation lags when $N = 15$ . . . . .	63
6.20	The distribution of the correlation lags when $N = 27$ . . . . .	64
6.21	The distribution of the correlation lags when $N = 36$ . . . . .	65
6.22	The distribution of the correlation lags when $N = 58$ . . . . .	65

# List of Tables

---

5.1	Optimal allocation of the inner samples and the outer samples with fixed total $N$ samples . . . . .	41
5.2	Comparison of sub-optimal sparse ruler algorithms and minimal sparse ruler . . . . .	43



In this thesis, we study compressive power spectrum estimation using non-uniform sampling. Power spectrum estimation is the problem of determining the distribution of the power of a signal over frequency. The word "compressive" means that the number of samples we used to estimate the power spectrum of the signal is less than the one produced by the Nyquist rate sampling. The purpose of this chapter is to introduce the problem addressed in the thesis, motivate the need for studying compressive power spectrum estimation and explain the organization and content of the following chapters.

## 1.1 Motivation

Nowadays, due to the development of the wireless technologies, the demand for spectrum resources has been rapidly increased. Wireless systems require spectrum to operate, but interference is likely to happen if radios operate on the same band at a certain place. Therefore, spectrum is a potentially scarce resource. In most of the countries, spectrum is regulated and most bands are allocated to licensed users in given locations. On one hand, such static spectrum allocation policies lead to significant underuse of spectrum [15]. On the other hand, most of the spectrum has already been allocated to licensed users. Take the United States frequency allocations for instance. From Fig. 1.1 we can see that National Telecommunications and Information Administration's (NTIA) frequency allocation chart indicates full allocations over all of the frequency bands.

With the current spectrum allocation, no available bandwidth is left for future wireless systems. And the Federal Communications Commission (FCC) has started considering dynamic approaches for spectrum sharing and the IEEE has launched the 802.22 standards process to use TV-band spectrum holes for wide-area Internet service [8], [25]. A statistical nation-wide perspective is given by the plot overlaid on the Midwest. Sampling the USA uniformly by area, on average 56% of the 67 television channels are free [24].

The background of Fig 1.2 is a map of the USA with the shading representing the population density. The red dots indicate the locations of all TV transmitters while the purple dots correspond to transmitters for channel 40. The green zone on the left zooms in on the San Francisco Bay Area to show the footprints where different stations can be received with an electric field strength above 41.19dBu for 50% of the locations more than 90% of the time. From this picture, it is clear that spectrum holes are inevitable. There is always going to be room for non-interfering radio transmissions in the interstices between channel footprints [27]. The plot along the top of Fig 1.2 shows the number of free television channels on a simulated drive from Berkeley, CA to Washington, DC along Interstate 80. The upper blue curve is the size of the opportunity

UNITED STATES FREQUENCY ALLOCATIONS THE RADIO SPECTRUM

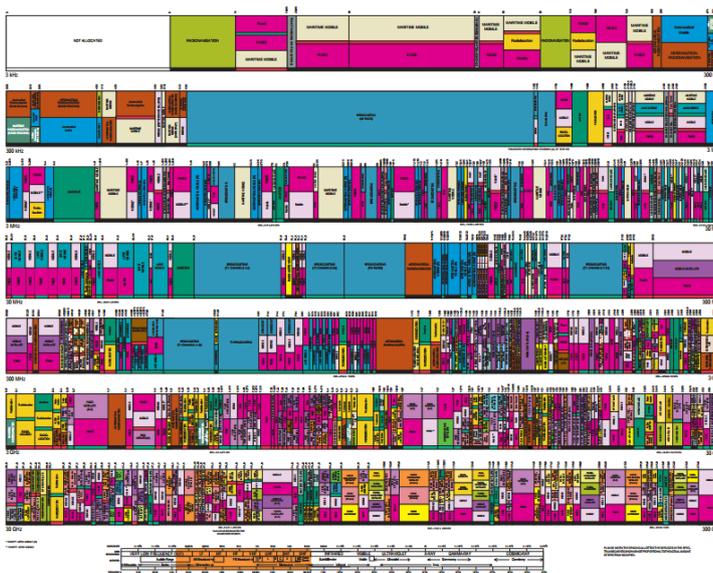


Figure 1.1: The NTIA frequency allocation chart [18]

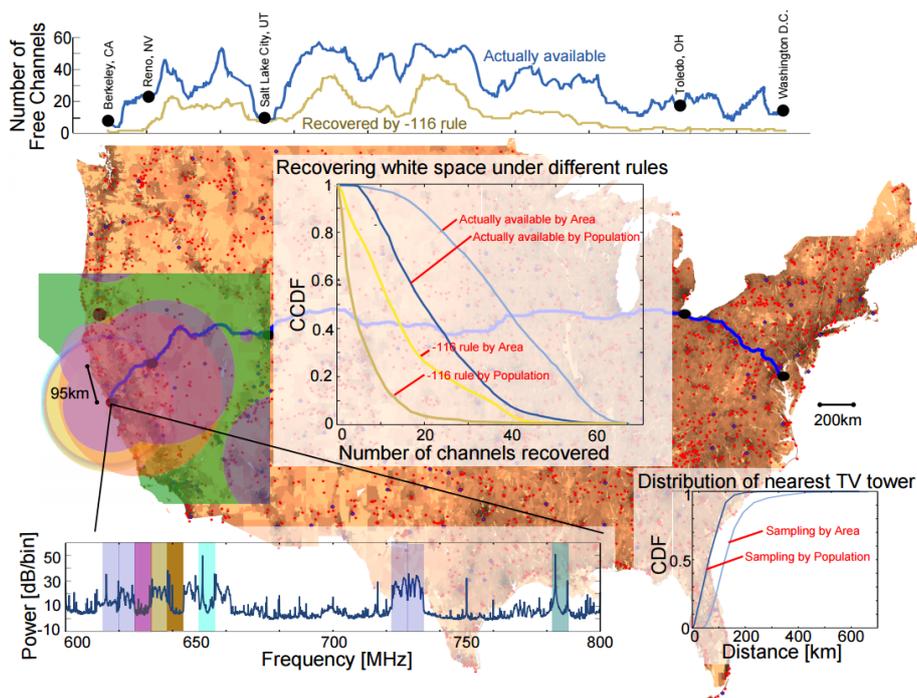


Figure 1.2: The nature of spectrum holes in the television bands [24]

based on International Telecommunications Union (ITU) models for wireless signal propagation. The lower tan curve illustrates the challenge in using cognitive radios, which is about making such radios smart enough to share spectrum [24].

Cognitive radio, first proposed in [19], is a technology with great promise to ex-

exploit the under-utilized spectrum. It is expected to be booming in the next wireless communication development. The cognitive radio has the ability to sense the communication environment dynamically and it can intelligently adapt the communication parameters (carrier frequency, bandwidth, power, coding schemes, modulation scheme etc.) [14]. The unused portion of licensed spectrum is called as white space. White space could be defined by time, frequency and maximum transmission power at a specific location. Cognitive user should be able to sense the environment for available white space. Besides, cognitive user should be able to know the application requirements and adopt their performance parameters based on user request and available resources [16]. Secondary (cognitive) user can utilize the white space in licensed spectrum without affecting the utilization of the spectrum by primary user. As we can see, it maximizes the efficiency of licensed spectrum utilization indeed.

In order to protect the primary users from unlicensed users, spectrum sensing is a key function. With the spectrum sensing, secondary users are able to decide whether a frequency band is empty or not. There are several spectrum sensing methods for cognitive radio, such as energy detector based sensing, waveform-based sensing, cyclostationarity-based sensing, radio identification based sensing, matched-filtering and other alternative spectrum sensing method. [32]. Cyclostationarity feature detection method detects primary user transmissions by making use of the cyclostationarity features of the received signals. Cyclostationary features like mean and autocorrelation, can be intentionally induced to calculate power spectrum estimation. The accuracy and complexity comparison of main sensing methods is showed in Fig 1.3. In our thesis, our power spectrum estimation also use the autocorrelations of the received signals, like cyclostationary feature detection. However, our power spectrum estimation method is different. The main difference is that the signal we used in our thesis is wide-band signal.

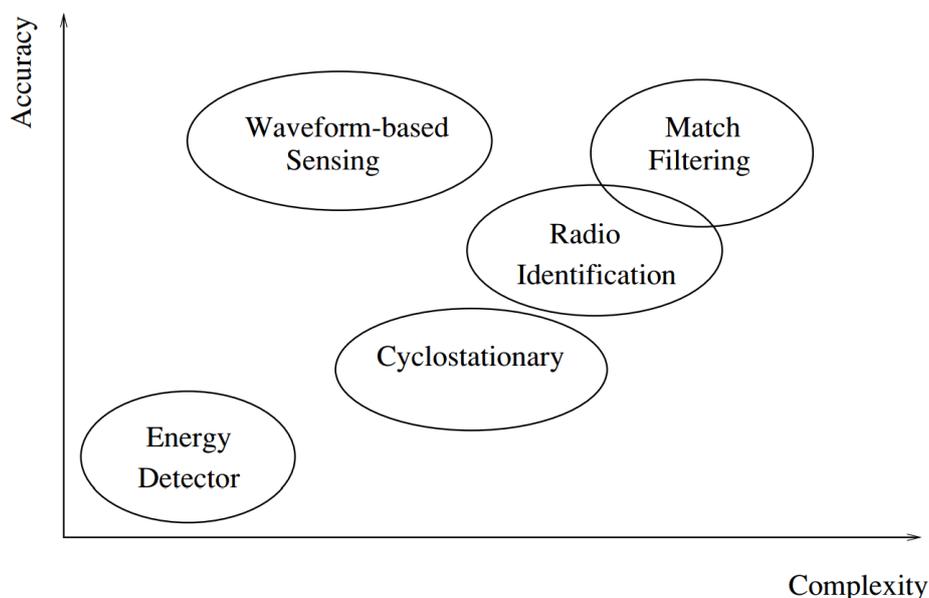


Figure 1.3: Main sensing methods in terms of their sensing accuracies and complexities

According to the classical Shannon-Nyquist-Whittaker-Kotelnikov sampling theorem [26], a band limited real signal  $x(t)$  (which means the frequency domain of the signal meet  $X(f) = 0$  for  $f > f_{max}$  and  $f < -f_{max}$ (Hertz), where  $f_{max}$  is a positive number) can be fully reconstructed from its samples  $x(nT)$  if the sampling frequency  $f_s = \frac{1}{T}$ , where  $T$  is the sampling time, satisfies  $f_s \geq 2f_{max}$ . This indicates that the sampling rate of a real analog signal should be at least twice as the maximum frequency of the signal. When the signal bandwidth is very large, the sampling rate, according to the Nyquist criterion (called Nyquist rate), should also be large. Hence sampling process executed by ADC will need a high power consumption. Our works in the thesis are aiming at estimating the power spectrum of the wide-band wide-sense stationary(WSS) signals. The outline and contribution of the thesis is showed in the next section.

## 1.2 Outline and Contributions

Here we would like to provide an overview of the work described in this thesis. The first contribution is the parametric power spectrum estimation. When using the parametric power spectrum estimation, we achieve a less compression rate comparing with the non-parametric estimation, which indicates that we can estimate the power spectrum of the signal with less samples. The second contribution is the sub-optimal minimal sparse ruler we have presented and the comparison between the sub-optimal rulers we presented and the existing sub-optimal rulers. The last contribution is the analysis of the normalized mean square error (NMSE) of the compressive power spectrum estimation of Gaussian white noise signal using alternative time domain approach discussed in [1]. The main content of each chapter in the thesis is outlined as follows.

### Chapter 1: Introduction

In this chapter, an introduction is given to describe the background of the power spectrum estimation. The motivation for the compressive power spectrum estimation is presented. We generally state that the spectrum resources we have could not match the demand of the wireless devices, and one of the solution is cognitive radio. To let the unlicensed user shares the spectrum resources with the licensed user, the unlicensed user have to detect whether the frequency band is occupied or not. Therefore, power spectrum estimation is needed. Moreover, the current analog-to-digital converter could not offer such a high speed sampling performance, thus compressive power spectrum estimation is proposed.

### Chapter 2: Background discussion

In this chapter we present a literature review of some specific compressive power spectrum estimation techniques. We end the chapter by describing the alternative time domain approach of [1] which we adapt in our thesis as the theoretical foundation. Our work is based on this approach.

### Chapter 3: Comparative analysis on Parametric and Non-parametric estimation

Based on alternative time domain approach, we compare the performances of parametric method and the non-parametric method for power spectrum estimation. In this chapter, no matter what method we employ, we need to calculate the autocorrelations of the original signal first. In non-parametric method we directly use the autocorrelations to estimate the power spectrum. However, in parametric method presented in this chapter, we use the calculated autocorrelations to estimate the parameters in the AR model which we use to model the signal. After that, we can calculate the power spectrum with the parameters in AR model. Moreover, we propose an simple sampling pattern to replace the minimal sparse ruler.

#### Chapter 4: Parametric Estimation of Power Spectrum using AR model

In this chapter, we present a new parametric methods to estimate the parameters in the signal model we adapt. Different from the method used in previous chapter, the parametric methods use the compressed signal directly, without calculating the autocorrelations of the original signal. We calculate the autocorrelations of the compressed signal, and determine the theoretical autocorrelations of the compressed signals using AR or ARMA models. Then, we can use the optimization methods to estimate the parameters. Using those methods, we can decrease the compression rate compared to what can be achieved by the non-parametric power spectrum estimation.

#### Chapter 5: Sub-optimal sparse ruler

In this chapter, we focus on the problem of designing sampling matrices for compressive sampling. The important rule for the sampling pattern is that the lags between samples should cover the desired autocorrelation range while the number of samples should be as less as possible. Certainly, the minimal sparse ruler offers the best compression. Unfortunately, there exists no quick procedure to find minimal sparse rulers: one must perform a brute-force search over the space of length- $N - 1$  sparse rulers to find a minimal one [22]. Therefore, we design some sub-optimal sparse rulers with certain procedure. The comparison of the sub-optimal sparse rulers we present and the existing sub-optimal sparse rulers has also been presented.

#### Chapter 6: Performance Analysis of Alternative Time Domain Approach and cosets selection

In this chapter, we study the performance of alternative time domain approach using non-parametric method for Gaussian white noise signal. We determine the normalized mean square error of the non-parametric method using  $M$ -branches sampling device. Then we analyse the relation between NMSE and the sampling pattern to determine the cosets selection. Besides, we study the performance of our approach for other kinds of wide-sense stationary signal and for different cosets selection.

#### Chapter 7: Conclusion and future works

This chapter summarizes the work described in the thesis and provides some suggestions for the further research and development on this topic.



## Background discussion

---

The goal of power spectrum estimation is to describe the distribution over frequency of the power of the signal we estimate, based on a finite set of data. The power spectral density (PSD) is the concept to describe the frequency distribution of the power of the signal. The PSD of a stationary random process  $x[n]$  is mathematically related to the correlation sequence  $r_x(n)$  by the discrete time Fourier transform, which is given by

$$S_x(e^{jw}) = \sum_{n=-\infty}^{+\infty} r_x(n)e^{-jwn} \quad -\pi < w \leq \pi \quad (2.1)$$

with

$$r_x(n) = E[x^*(m)x(m+n)]. \quad (2.2)$$

### 2.1 Review of Compressive power spectrum estimation

The main methods for wideband power spectrum estimation can be divided into non-parametric methods and parametric methods [11].

- 1 Nonparametric methods are those in which the power spectrum density is estimated directly from the signal itself. Two common non-parametric methods are periodogram method and correlogram method [26]. In periodogram method, the power spectrum is estimated as

$$\hat{\phi}_{x,p}(w) = \frac{1}{N} \left| \sum_{n=0}^{N-1} x[n]e^{-jwn} \right|^2, \quad (2.3)$$

while in the correlogram method, the power spectrum is estimated as

$$\hat{\phi}_{x,c}(w) = \sum_{k=1-N}^{N-1} \hat{r}_x[k]e^{-jwk}, \quad (2.4)$$

where

$$\hat{r}_x[k] = \frac{1}{N-k} \sum_{n=k}^{N-1} x[n]x^*[n-k] \quad 0 \leq k \leq N-1. \quad (2.5)$$

- 2 Parametric methods are those in which the power spectrum density is estimated from a signal that is assumed to be the output of an LTI system driven by white noise. Examples are the autoregressive (AR) and the autoregressive moving average (ARMA) methods. These methods first estimate the parameters of the system model used to describe the signal. Then according to the parameters, power spectrum is calculated.

Compressive power spectrum estimation has been carried out several years ago. It has been showed that the analog and the digital models can be treated and analyzed in a uniform way in the frequency domain [7]. Therefore we only need to pay attention to the digital model. There are several compressive power spectrum estimation for specific cases. If we have already known the detailed information about the transmitted signals, such as the carrier frequency, the bandwidth, the waveform of the signal and the second order moments of the signal, we can estimate the power spectrum by making use of those prior information [23].

Power spectrum estimation is useful in wireless sensor network applications. One example is in a cognitive radio network, where a cognitive radio user is allowed to borrow an unused licensed spectrum from licensed user. Here the CR has to perform power spectrum sensing to find the unused spectrum that can be borrowed. However, sometimes, the sensor, the CR user, may suffer from multipath fading or severe shadowing, leading to a wrong power spectrum estimation result. Therefore, multi sensor receivers can cooperate with each other, exchange results, to improve the power spectrum estimation accuracy [2]. Cooperative compressive power spectrum sensing can also be used to reduce the sampling rate for the sensors in the cooperative group [3]. Cooperative distributed compressive spectrum sensing can be used to reduce the data acquisition costs by exploiting the signal sparsity [33]. Here, however, we focus on one CR user, without any prior information, and we focus on some approaches discussed in [1], which basically classify their approaches into time domain, alternative time domain, and frequency domain reconstruction approaches.

## 2.2 Alternative time domain approach of [1]

In this thesis, all the work is based on the alternative time-domain approach discussed in [1]. The following is a short introduction to the alternative time-domain approach.

Let us define  $x(t)$  as the received wide-sense stationary signal, which is sampled using multi-coset sampling [13][17][28] with  $N$  cosets. We then define  $\mathbf{x}[k]$  as the  $N \times 1$  vector containing the  $(k + 1)$ -th block of  $N$  consecutive received samples, i.e.,

$$\mathbf{x}[k] = [x[kN], x[kN + 1], \dots, x[kN + N - 1]]^T. \quad (2.6)$$

The sub-Nyquist sampling is implemented by selecting  $M$  out of  $N$  samples in each block. This is done by activating only  $M$  cosets in the multi-coset sampler, which can be explained by using Fig 2.1 [1]. Note that Fig 2.1 is a general sampler, which can be used to implement multi cosets sampling by setting the entries  $c_i[n]$  to 1 at particular  $n$  and to 0 at other indices  $n$ . Based on Fig 2.1, the output at the  $(i + 1)$ -th activated coset is given by

$$y_i[k] = \sum_{n=1-N}^0 c_i[n]x[kN - n], \quad i = 0, 1, \dots, M - 1, \quad (2.7)$$

where  $c_i[n]$  can be perceived as the  $(1 - n)$ -th element of the  $(i + 1)$ -th row of an  $M \times N$  sampling matrix  $\mathbf{C}$ , which is a selection matrix, whose rows are selected from the rows

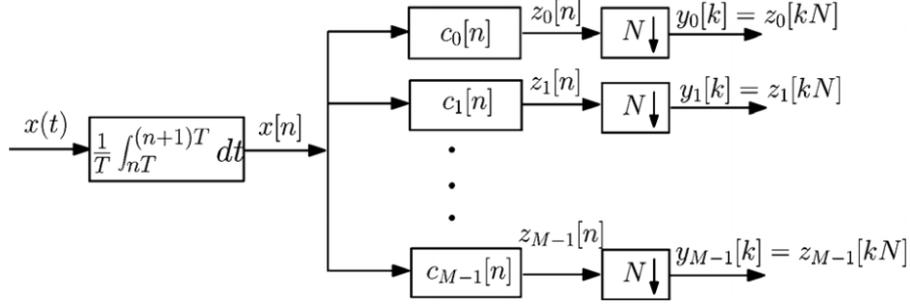


Figure 2.1: Digital interpretation of the sampling device with a bank of  $M$  branches, where each branch consists of a digital filtering operation followed by a down-sampling operation [1]

of the  $N \times N$  identity matrix  $\mathbf{I}_N$ . Note that  $\{c_i[n]\}_{i,n}$  indicates which cosets out of  $N$  cosets are selected. After rewriting the equation we have

$$\mathbf{y}[k] = \mathbf{C}\mathbf{x}[k] \quad (2.8)$$

where

$$\mathbf{y}[k] = [y_0[k], y_1[k], \dots, y_{M-1}[k]]^T \quad (2.9)$$

$$\mathbf{C} = [\mathbf{c}_0, \mathbf{c}_1, \dots, \mathbf{c}_{M-1}]^T \quad (2.10)$$

with  $\mathbf{c}_i = [c_i[0], c_i[-1], \dots, c_i[1-N]]^T$ . In our work, there is only one element of  $\mathbf{c}_i$  that will be set to one while others will be set to zero. After calculating the autocorrelation matrix of  $\mathbf{y}[k]$ , which is given by  $\mathbf{R}_y[0] = E(\mathbf{y}[k]\mathbf{y}^H[k])$ , we obtain

$$\mathbf{R}_y[0] = \mathbf{C}\mathbf{R}_x\mathbf{C}^T \quad (2.11)$$

where  $\mathbf{R}_x = E(\mathbf{x}[k]\mathbf{x}^H[k])$ . Then we stack all columns of  $\mathbf{R}_y[0]$  into the  $M^2 \times 1$  vector  $\text{vec}(\mathbf{R}_y[0])$ . We define  $\text{vec}(\cdot)$  as an operation that stacks columns of a matrix into a vector and define the  $\text{unvec}(\cdot)$  as the opposite operation. We then obtain

$$\mathbf{r}_y[0] = \text{vec}(\mathbf{R}_y[0]) = (\mathbf{C} \otimes \mathbf{C})\text{vec}(\mathbf{R}_x). \quad (2.12)$$

Since all columns of  $\mathbf{R}_x$  contain the same information,  $\text{vec}(\mathbf{R}_x)$  can also be presented by

$$\text{vec}(\mathbf{R}_x) = \mathbf{T}\mathbf{r}_x \quad (2.13)$$

where  $\mathbf{r}_x = [r_x[0], r_x[1], \dots, r_x[N-1], r_x[1-N], \dots, r_x[-1]]^T$  and  $\mathbf{T}$  is a special  $N^2 \times (2N-1)$  repetition matrix with the  $i$ -th row of  $\mathbf{T}$  given by the  $((i-1 + (N-2) \lfloor \frac{i-1}{N} \rfloor) \bmod (2N-1) + 1)$ -th row of the identity matrix  $\mathbf{I}_{2N-1}$ . By rewriting (2.12), we obtain

$$\mathbf{r}_y[0] = (\mathbf{C} \otimes \mathbf{C})\mathbf{T}\mathbf{r}_x = \mathbf{R}_c\mathbf{r}_x. \quad (2.14)$$

In practice, the autocorrelation  $\mathbf{r}_y[0]$  has to be estimated as  $\hat{\mathbf{r}}_y[0]$ , whose elements can be written as

$$\hat{r}_{y_i, y_j}[0] = \frac{1}{K} \sum_{k=0}^{K-1} y_i[k] y_j^*[k], \quad i, j = 0, 1, \dots, M-1. \quad (2.15)$$

The estimate of  $\mathbf{r}_x$  is then given by  $\hat{\mathbf{r}}_x$ , which can be computed from  $\hat{\mathbf{r}}_y[0]$  by considering (2.14) and using least squares method if  $\mathbf{R}_c$  has full column rank, i.e.,

$$\hat{\mathbf{r}}_x = (\mathbf{R}_c^T \mathbf{R}_c)^{-1} \mathbf{R}_c^T \hat{\mathbf{r}}_y[0] \quad (2.16)$$

As we know, the relationship between the estimated autocorrelation and the estimated power spectrum of the signal is given by

$$\hat{P}_x(e^{jw}) = \sum_{k=1-N}^{N-1} \hat{r}_x[k] e^{-jkw} \quad (2.17)$$

# Comparative analysis on Parametric and Non-parametric estimation

---

# 3

## 3.1 Autoregressive (AR) model parameters estimation

While we can use the computed  $\hat{\mathbf{r}}_x$  to estimate the power spectrum of the signal by directly using non-parametric method, we are here also interested in power spectrum estimation using parametric method. In the latter case, we also use the computed  $\hat{\mathbf{r}}_x$  above. Note that the parametric approach models the signal as a filtered white noise signal. Here we assume that the filter is an AR model filter with the system function given by

$$H(z) = \frac{b[0]}{1 + \sum_{k=1}^p a[k]z^{-k}}. \quad (3.1)$$

In order to estimate the parameters in the AR model, we use the Yule-Walker equations [12], which is given in matrix form as

$$\begin{bmatrix} \hat{r}_x[0] & \hat{r}_x[-1] & \dots & \hat{r}_x[1-p] \\ \hat{r}_x[1] & \hat{r}_x[0] & \dots & \hat{r}_x[2-p] \\ \vdots & \ddots & & \vdots \\ \hat{r}_x[p-1] & \hat{r}_x[p-2] & \dots & \hat{r}_x[0] \end{bmatrix} \begin{bmatrix} a[1] \\ a[2] \\ \vdots \\ a[p] \end{bmatrix} = - \begin{bmatrix} \hat{r}_x[1] \\ \hat{r}_x[2] \\ \vdots \\ \hat{r}_x[p] \end{bmatrix}. \quad (3.2)$$

We can also express it as  $\hat{\Sigma}_x \mathbf{a} = -\hat{\mathbf{r}}_x$ , where  $\hat{\Sigma}_x$  is the  $p \times p$  matrix and  $\hat{\mathbf{r}}_x$  is the  $p \times 1$  vector. Using the LS method we can estimate the parameters vector  $\hat{\mathbf{a}} = -(\hat{\Sigma}_x^H \hat{\Sigma}_x)^{-1} \hat{\Sigma}_x^H \hat{\mathbf{r}}_x$ . Then we can also estimate  $\hat{b}[0]$  in (3.1) as

$$\hat{b}[0]^2 = \hat{r}_x[0] + \sum_{k=1}^p \hat{a}[k] \hat{r}_x[k]. \quad (3.3)$$

## 3.2 Two options in parametric power spectrum estimation

To estimate the  $p$  parameters in the AR model, we only need to know the corresponding autocorrelation of the signal from lags  $1-p$  to  $p-1$ . In Fig 3.1, the red area is the autocorrelations we have to obtain. In our thesis, all signals we used are wide-sense stationary (WSS) signals, which have symmetry property,  $r_x[k] = r_x^*[-k]$ . Therefore, once we obtain the autocorrelation of the signal at lag  $k$ , we also obtain the corresponding autocorrelation at lag  $-k$ . Of course we can focus on more than  $2p-1$  lags autocorrelations (in the blue area in Fig. 3.1) to estimate the parameters, but it does not make remarkable difference. Since we do not need to know the entire autocorrelations of the  $N$  signal samples, we can make a lossy compressive sampling. We only need to

collect autocorrelations at at least  $2p - 1$  consecutive lags over  $N$  signal samples. Here we assume that we collect autocorrelation from lags  $-\bar{N} + 1$  up to lags  $\bar{N} - 1$ , with  $p \leq \bar{N} \leq N$ .

$$\begin{bmatrix}
 \hat{r}_x[0] & \hat{r}_x[-1] & \dots & \hat{r}_x[1-p] \\
 \hat{r}_x[1] & \hat{r}_x[0] & \dots & \hat{r}_x[2-p] \\
 \vdots & \ddots & & \vdots \\
 \hat{r}_x[p-1] & \hat{r}_x[p-2] & \dots & \hat{r}_x[0] \\
 \hat{r}_x[p] & \hat{r}_x[p-1] & \dots & \hat{r}_x[1] \\
 \hat{r}_x[p+1] & \hat{r}_x[p] & \dots & \hat{r}_x[2] \\
 \hat{r}_x[p+2] & \hat{r}_x[p+1] & \dots & \hat{r}_x[3] \\
 \vdots & \ddots & & \vdots \\
 \hat{r}_x[N-2] & \hat{r}_x[N-3] & \dots & \hat{r}_x[N-p-1]
 \end{bmatrix}
 \begin{bmatrix}
 a[1] \\
 a[2] \\
 \vdots \\
 a[p]
 \end{bmatrix}
 = -
 \begin{bmatrix}
 \hat{r}_x[1] \\
 \hat{r}_x[2] \\
 \vdots \\
 \hat{r}_x[p] \\
 \hat{r}_x[p+1] \\
 \hat{r}_x[p+2] \\
 \hat{r}_x[p+3] \\
 \vdots \\
 \hat{r}_x[N-1]
 \end{bmatrix}$$

Figure 3.1: The autocorrelations picked to estimate parameters

After computing the AR parameters, we have two options to estimate the power spectrum. One option is to directly estimate the power spectrum by using the estimated parameters in (3.1) since the power spectrum can be written as

$$\hat{P}_x(e^{jw}) = \sigma^2 \frac{|\hat{b}[0]|^2}{|1 + \sum_{k=1}^p \hat{a}[k]e^{-jkw}|^2} \quad (3.4)$$

where  $\sigma^2$  is the variance of the white noise (whose value usually be set as 1) we used to generate the WSS signal by passing the filter. Another option is to predict the autocorrelations of the signal starting from lags  $\pm\bar{N} - 1$  up to lags  $\pm N - 1$ . Then we can estimate the power spectrum of the signal with all the estimated and predicted autocorrelation values using (2.17).

### 3.3 Formulation of the experiments

Recall from [1] that the implementation of the non-parametric approach for estimating the power spectrum in [1] sets the multi-coset sampler to operate over each block of  $N$  samples, i.e., the multi-coset sampler in [1] is designed to theoretically collect  $M < N$  samples from every possible block of  $N$  samples according to the  $(N-1)$ -length minimal sparse ruler. In addition, [1] focuses to compute the autocorrelation of the signal of interest at lags from  $1 - N$  to  $N - 1$ .

Before we proceed, we would like to introduce the concept of  $(N-1)$ -length minimal sparse ruler. Let us introduce  $S$  as a set that consists of  $M$  indices selected from  $\{0, 1, \dots, N-1\}$  and  $\Omega_S$  as the set of differences of the indices, written as  $\Omega_S = \{|n_i - n_j| | \forall n_i, n_j \in S\}$ . The length- $(N-1)$  sparse ruler problem is then about finding the set  $S$  such that  $\{0, 1, \dots, N-1\} \subset \Omega_S$ . If we want to minimize  $|S|$  (the cardinality of  $S$ ), the problem becomes the minimal length  $-(N-1)$  sparse ruler problem given by

$$\min |S| \text{ s.t. } \{0, 1, \dots, N-1\} \subset \Omega_S. \quad (3.5)$$

Specifically,  $S$  that satisfies (3.5) is the  $(N - 1)$ -length optimal sparse ruler.

In this chapter, we focus on parametric power spectrum estimation approaches. And as indicated in the previous section, we would like to investigate if we can exploit the general properties of parametric power spectrum estimation approach to reduce the sampling rate while maintaining the ability to reconstruct the power spectrum. In order to do that, we re-emphasize our intention to first focus on the computation of the autocorrelation values at lags from  $1 - \bar{N}$  to  $\bar{N} - 1$  (instead of from  $1 - N$  to  $N - 1$ ) with  $\bar{N} < N$ . Here, we perform the comparative analysis between the non-parametric and parametric approaches by conducting two main experiments, Experiment I and II.

Experiment I consists of three sub-experiments explained as follows:

- Experiment I.A: We basically apply non-parametric approach labeled as alternative time domain approach in [1]. We reconstruct  $\{\hat{r}_x[n]\}_{n=1-N}^{N-1}$  from compressive samples and compute the power spectrum by applying discrete Fourier Transform (DFT) on the resulting  $\{\hat{r}_x[n]\}_{n=1-N}^{N-1}$ . This experiment plays a role as a benchmark.
- Experiment I.B and I.C: We implement parametric approaches. Here, out of every  $N$  samples,  $\{x[kN + n]\}_{n=0}^{N-1}$ , we only focus on the first  $\bar{N}$  samples  $\{x[kN + n]\}_{n=0}^{\bar{N}-1}$  and collect  $M < \bar{N} < N$  samples from these  $\bar{N}$  samples. In other words, we have our  $M \times N$  matrix  $\mathbf{C}$  for these experiments given by

$$\mathbf{C} = [\bar{\mathbf{C}} \quad \mathbf{0}_{M \times (N - \bar{N})}] \quad (3.6)$$

with  $\mathbf{0}_{M \times (N - \bar{N})}$  the  $M \times (N - \bar{N})$  containing zeros. Observe that, in this case, the resulting  $\mathbf{R}_c$  in (2.16) does not have full rank, because some columns of  $\mathbf{R}_c$  only contain zeros. Note that we are only interested in the reconstruction of  $\{\hat{r}_x[n]\}_{n=1-\bar{N}}^{\bar{N}-1}$  and thus, instead of forming the  $M^2 \times (2N - 1)$  matrix  $\mathbf{R}_c$ , we could form another matrix  $\mathbf{R}_{\bar{c}}$  having size of  $M^2 \times (2\bar{N} - 1)$ . Here  $\mathbf{R}_{\bar{c}}$  is formed from  $\mathbf{R}_c$  by removing its columns that contain only zeros. From here, we can rewrite  $\mathbf{r}_y[0]$  in (2.14) as

$$\mathbf{r}_y[0] = \mathbf{R}_{\bar{c}} \bar{\mathbf{r}}_x \quad (3.7)$$

with  $\bar{\mathbf{r}}_x = [r_x[0], \dots, r_x[\bar{N} - 1], r_x[1 - \bar{N}], \dots, r_x[-1]]^T$ . To ensure that  $\mathbf{R}_{\bar{c}}$  is full rank, we design the sampling pattern  $\bar{\mathbf{C}}$  based on  $(\bar{N} - 1)$ -length minimal sparse ruler. After obtaining the autocorrelation  $\{\hat{r}_x[n]\}_{n=1-\bar{N}}^{\bar{N}-1}$ , we estimate the parameters in the AR model. In Experiment I.B, we use the estimated model parameters  $\hat{a}[k]$  and  $\hat{b}[0]$  to compute the power spectrum estimate by using (3.4). In Experiment I.C, once we obtain the model parameters  $\hat{a}[k]$  and  $\hat{b}[0]$ , we use them to predict the autocorrelation estimate  $\hat{r}_x[n]$  at lags  $1 - N \leq n \leq -\bar{N}$  and lags  $\bar{N} \leq n \leq N - 1$ . We then compute the power spectrum estimate by applying DFT on the resulting  $\{\hat{r}_x[n]\}_{n=1-N}^{N-1}$ .

Experiment II also consists of three experiments.

- Experiment II.A: The experiment here is the same as what we do in Experiment I.A, i.e., using non-parametric approach as a benchmark.

- Experiment II.B and II.C: We implement parametric approaches. Here, out of every  $N$  samples,  $\{x[kN + n]\}_{n=0}^{N-1}$ , we collect  $\bar{M} < \bar{N} < N$  samples from these  $N$  samples, based on the algorithm we designed, showed below. In other words, we have our  $\bar{M} \times N$  matrix  $\mathbf{C}_{\bar{M}}$  for these experiments. Different from the  $M \times N$  matrix  $\mathbf{C}$  in Experiment I.B and I.C, the matrix  $\mathbf{C}_{\bar{M}}$  here may also collect the  $\bar{M}$  samples from  $\{x[n]\}_{n=\bar{N}}^{N-1}$  even though we again only focus on first reconstructing the autocorrelation  $\{r_x[n]\}_{n=1-\bar{N}}^{\bar{N}-1}$ . Note that besides  $\{r_x[n]\}_{n=1-\bar{N}}^{\bar{N}-1}$ , we may also obtain some of the autocorrelations  $\{r_x[n]\}_{n=\bar{N}}^{N-1}$  or  $\{r_x[n]\}_{n=1-N}^{-\bar{N}}$ . We then use  $\mathbf{C}_{\bar{M}}$  to reconstruct  $\bar{M}^2 \times (2N - 1)$  matrix  $\mathbf{R}_{c_{\bar{M}}}$  by using procedure similar to (2.14). After removing all zero columns in  $\mathbf{R}_{c_{\bar{M}}}$ , just like in Experiment I.B and I.C, we obtain a new matrix  $\mathbf{R}_{\bar{c}_{\bar{M}}}$ . Here, we can rewrite  $\mathbf{r}_y[0]$  in (2.14) as

$$\mathbf{r}_y[0] = \mathbf{R}_{\bar{c}_{\bar{M}}} \bar{\mathbf{r}}_{x_{\bar{M}}}, \quad (3.8)$$

where all elements of  $\bar{\mathbf{r}}_x$  in (3.7) exist as elements of  $\bar{\mathbf{r}}_{x_{\bar{M}}}$ . After obtaining the autocorrelation values in  $\bar{\mathbf{r}}_{x_{\bar{M}}}$ , we first only pick  $\{\hat{r}_x[n]\}_{n=1-\bar{N}}^{\bar{N}-1}$ . Then, we use them to estimate the parameters  $\hat{a}[k]$  and  $\hat{b}[0]$  in the AR model. In Experiment II.B, we use the estimated model parameters  $\hat{a}[k]$  and  $\hat{b}[0]$  to compute the power spectrum estimate by using (3.4). In Experiment II.C, once we obtain the model parameters  $\hat{a}[k]$  and  $\hat{b}[0]$ , we use them to predict the autocorrelation estimate  $\hat{r}_x[n]$  at lags  $1 - N \leq n \leq -\bar{N}$  and lags  $\bar{N} \leq n \leq N - 1$ . We then compute the power spectrum estimate by applying DFT on the resulting  $\{\hat{r}_x[n]\}_{n=1-N}^{N-1}$ .

In Experiment II.B and II.C, the algorithm that is used to decide the selection of  $\bar{M}$  samples from every  $N$  consecutive Nyquist-rate samples is given as:

1. Define  $S_N$  as the set of the selected indices and first initialize  $S_N$  as  $S_N = \{0, 1\}$ .
2. We then compute  $\Omega_{S_N}$  as  $\Omega_{S_N} = \{|n_i - n_j| | \forall n_i, n_j \in S_N\}$  but we also remove all elements that are larger than  $N - 1$  from  $\Omega_{S_N}$ . Compute  $\mathcal{U}_{S_N}$  as  $\mathcal{U}_{S_N} = \{0, 1, \dots, \bar{N} - 1\} \setminus \Omega_{S_N}$ .
3. If  $\mathcal{U}_{S_N} = \emptyset$  then  $S_N$  is what we need otherwise, we do the following steps. If  $|S_N|$  is odd, then we select the median of  $S_N$  and call it  $\tilde{n}_i$ . If  $|S_N|$  is even, we sort the elements of  $S_N$ , select the two elements in the middle, and set  $\tilde{n}_i$  to the smallest of these two elements. We then update  $S_N$  as  $S_w = S_N \cup \{\tilde{n}_i + \min(\mathcal{U}_{S_N})\}$  and repeat step 2.

### 3.4 Simulation study

To calculate the autocorrelations from the received signals, we adopt  $K = 2000$  segments. In Experiment I, we have  $\bar{N} = 11$  and  $N = 20$ . We use the solution for length- $(\bar{N} - 1)$  minimal sparse ruler to determine  $M$  and this leads to  $M = 6$  and compression rate  $\frac{M}{N} = \frac{6}{20}$ . For Experiment I.C, we use  $\{\hat{r}_x[n]\}_{n=-10}^{10}$  to predict the  $\hat{r}_x[n]$  at lags  $N = -20 < n < 1 - \bar{N} = -10$  and lags  $\bar{N} - 1 = 10 < n < N = 20$ .

The simulated received signal is generated by passing a white Gaussian noise through a filter. Note that the filter that is used to generate the wide-sense stationary (WSS) signal does not always match the examined AR model in experiments I.B and I.C. We generally have four cases. In the first case, the filter that is used to generate the WSS signal is given by

$$H(z) = \frac{1}{1 + 0.5z^{-1} + 0.4z^{-2} + 0.6z^{-3}}. \quad (3.9)$$

Meanwhile, the AR model that is assumed for the parametric approach in Experiments I.B and I.C has order of 3 (i.e., we set  $p$  in (3.1) to  $p = 3$ ). In this case the assumed AR model accurately models the received signal. As shown in Fig. 3.2, the results of Experiments I.B and I.C have better quality than that of Experiment I.A. It seems that the best result is found in Experiment I.C.

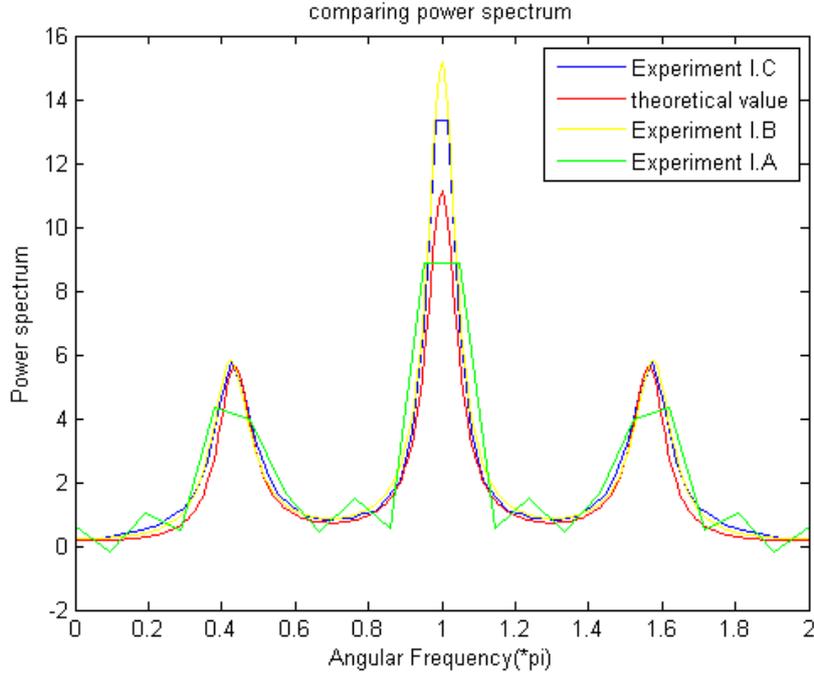


Figure 3.2: Experiment I, Case 1

In the second case of Experiment I, the filter that is used to generate the signal is the same as the one used in the first case. On the other hand, the order of the AR model assumed for Experiments I.B and I.C is now set to  $p = 2$ . It is obvious that the assumed AR model now does not accurately model the received signal. As shown in Fig. 3.3, it is clear that the quality of the result in Experiment I.B is not acceptable compared to the ones in the other two sub-experiments. The results of Experiments I.A and I.C have nearly the same quality.

In the third case of Experiment I, the filter that is used to generate the WSS signal is given by

$$H(z) = \frac{1 + 3z^{-1} + 4z^{-2}}{1 + 0.5z^{-1} + 0.4z^{-2} + 0.6z^{-3}}. \quad (3.10)$$

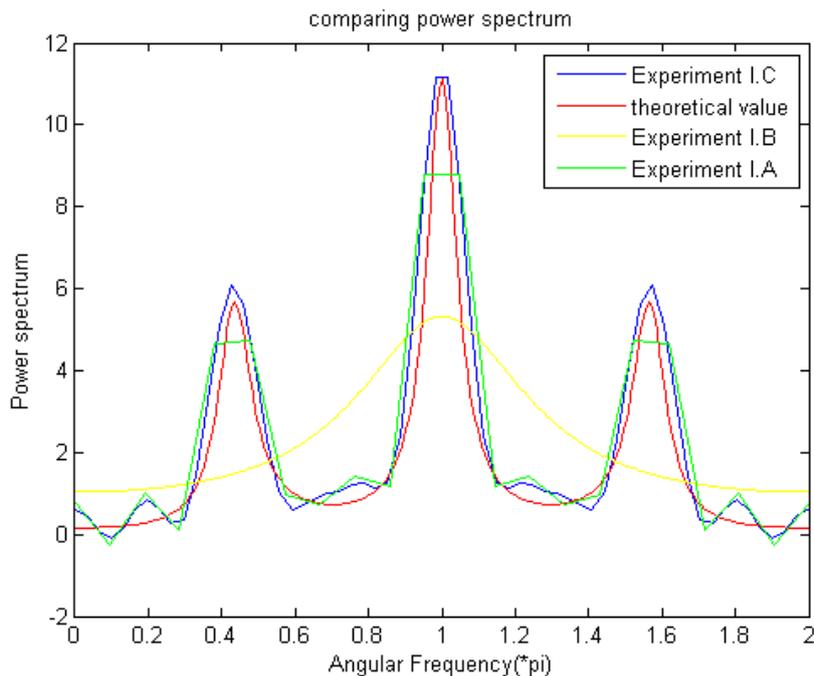


Figure 3.3: Experiment I, Case 2

We again assume an AR model for Experiments I.B and I.C with an order of  $p = 3$ . Observe that the model does not match the actual signal, since the actual signal is generated by an ARMA filter. As shown in Fig. 3.4, the estimated power spectrum in the Experiments I.A and I.C is close to the actual power spectrum but the result in Experiment I.B is not accurate.

In the fourth case of Experiment I, the filter that is used to generate the WSS signal is the same as the one used in the third case. Meanwhile, the order of the AR model assumed for Experiments I.B and I.C is now set to  $p = 2$ . Again there is a mismatch between the model and the actual signal. However, the degree of the mismatch is higher in this case because in the third case, the denominators of both the model and the generating filter is a third order polynomial. As shown in Fig. 3.5, the results are quite the same as the ones found in the third case. The only difference is that in Experiment I.B, the result is even worse than the corresponding result in the third case.

In Experiment II, we have  $\bar{N} = 11$  and  $N = 20$ . We use the solution for length- $(N - 1)$  minimal sparse ruler to determine  $M_N$  and this leads to  $M_N = 8$  and compression rate  $\frac{M_N}{N} = \frac{8}{20}$  for Experiment II.A. For Experiments II.B and II.C, we use the solution of our algorithm listed in Section 3.3 to determine  $M_{\bar{N}}$  and this leads to  $M_{\bar{N}} = 6$  and compression rate  $\frac{M_{\bar{N}}}{N} = \frac{6}{20}$ . For Experiment II.C, we use the  $\{\hat{r}_x[n]\}_{n=-10}^{10}$  to predict the  $\hat{r}_x[n]$  at lags  $N = -20 < n < 1 - \bar{N} = -10$  and lags  $\bar{N} - 1 = 10 < n < N = 20$ . The simulated received signal is also generated by passing a white Gaussian noise through a filter. And the filter used to generate the WSS signal does not always match the

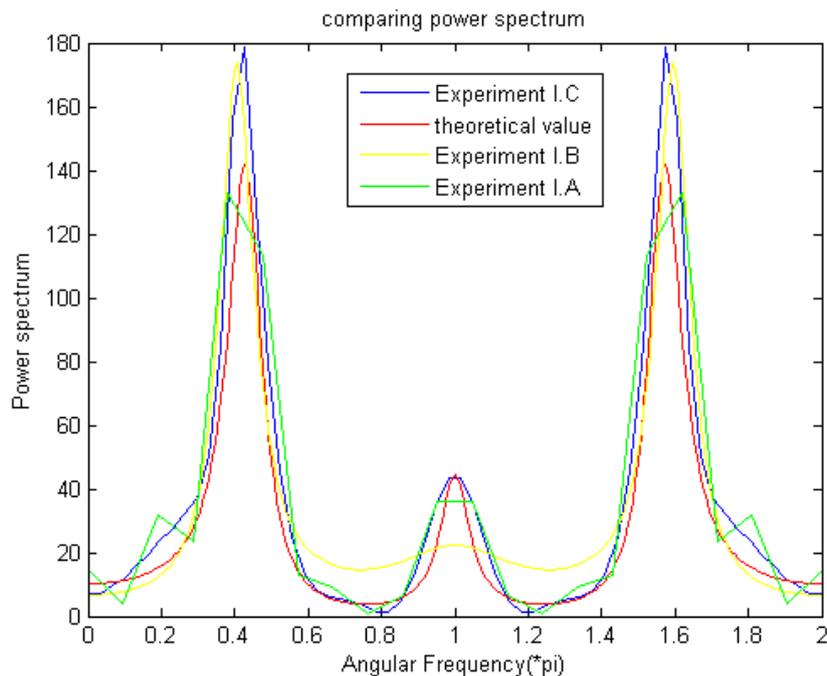


Figure 3.4: Experiment I, Case 3

examined AR model in the Experiments II.B and II.C. As in Experiment I, we here also examine four cases.

In the first case of Experiment II, the filter used to generate the WSS signal has the same system function as (3.9). Meanwhile, the AR model that is assumed for the parametric approach in Experiments II.B and II.C has order of 3 (i.e., we set  $p$  in (3.1) to  $p = 3$ ). In this case, the assumed AR model accurately models the received signal. As shown in Fig. 3.6, all the results have good quality.

In the second case of Experiment II, the filter that is used to generate the WSS signal is the same as the one used in the first case of Experiment II. On the other hand, the order of the AR model assumed for Experiments II.B and II.C is now set to  $p = 2$ . Now the assumed AR model does not accurately model the received signal. As shown in Fig. 3.7, it is clear that the quality of the result in Experiment II.B is not acceptable, the same as in the second case of Experiment I. The results of Experiments II.A and II.C have nearly the same quality.

In the third case of Experiment II, the filter used to generate the WSS signal has the system function as in (3.10). We assume an AR model for Experiments II.B and II.C with an order of  $p = 3$ . Since the actual signal is generated by an ARMA filter, the AR model does not match the actual signal. As shown in Fig. 3.8, the estimated power spectrum in the Experiments II.A and II.C are nearly the same and are close to the actual power spectrum of the signal, while the estimated power spectrum in the Experiment II.B is not accurate.

In the fourth case of Experiment II, the filter that is used to generate the WSS

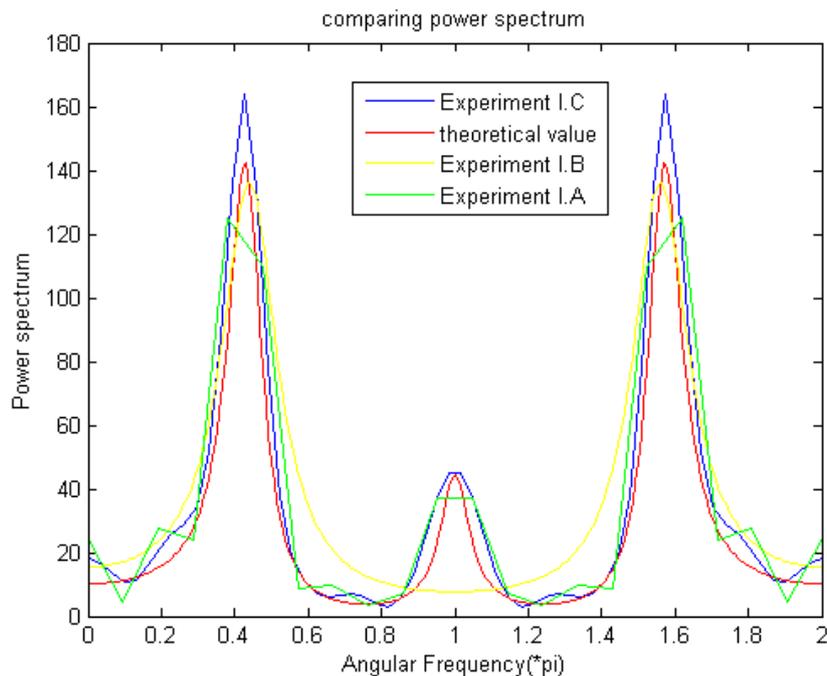


Figure 3.5: Experiment I, Case 4

signal is the same as the one used in the third case of the Experiment II. Meanwhile, the order of the AR model assumed for Experiments II.B and II.C is set to  $p = 2$ . There is a mismatch between the model and the actual signal as well. But the degree of the mismatch is higher than in the third case of Experiment II since unlike in the third case of Experiment II, the polynomial in the denominators of the model and the generating filter do not have the same order. As shown in Fig. 3.9, the results are quite the same as the ones found in the third case of Experiment II. The only difference is that the result of Experiment II.B is even worse than the corresponding result in the third case of Experiment II.

From the above simulation results, we can find that if the model in the parametric approach is not properly chosen and if the estimated parameters are directly used to estimate the power spectrum, the resulting power spectrum estimate will be very inaccurate. On the other hand, the performance of the parametric approach is quite good when the model fits the characteristic of the actual received signal. Meanwhile, the approach that focus on predicting the autocorrelation at higher lags leads to a good estimate of the power spectrum and it appears to be more robust than the other approaches. In addition, Experiment II offers a way to estimate the power spectrum with smaller compression rate than the one in Experiment I.

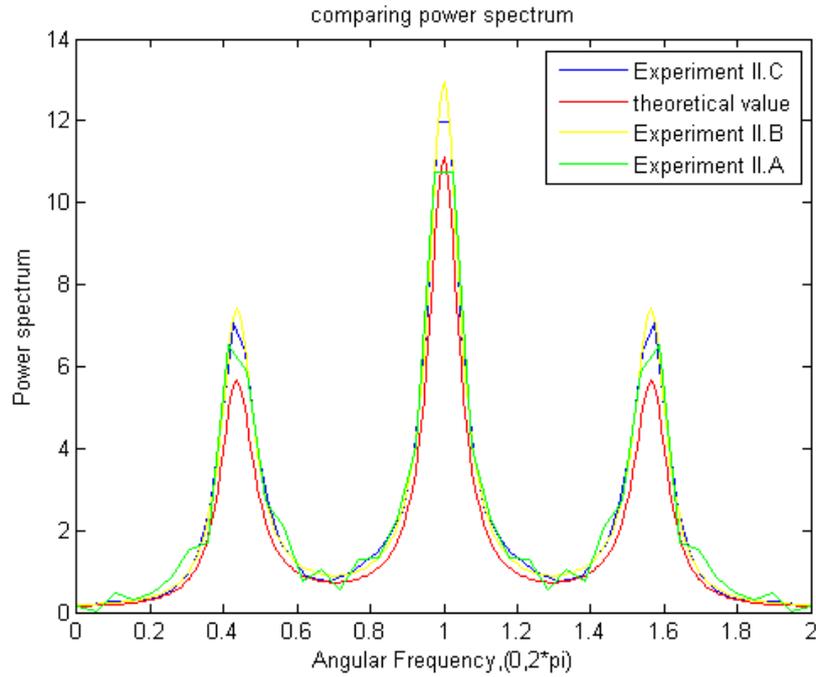


Figure 3.6: Experiment II, Case 1

### 3.5 Conclusion

From the two experiments we can find that the parametric compressive power spectrum estimation can achieve less compression compared to the one achieved by the non-parametric estimation and it results in a good performance, when we model the signal properly. However, this kind of parametric power spectrum estimation relies on the calculated autocorrelations of the signal, and therefore, we need to estimate the autocorrelations of the signal like what we do in non-parametric method first. Second, this kind of parametric power spectrum estimation could not employ ARMA model to estimate the power spectrum because it is quite difficult to calculate the  $b$  parameters from the Yule-Walker equation due to the nonlinearity.

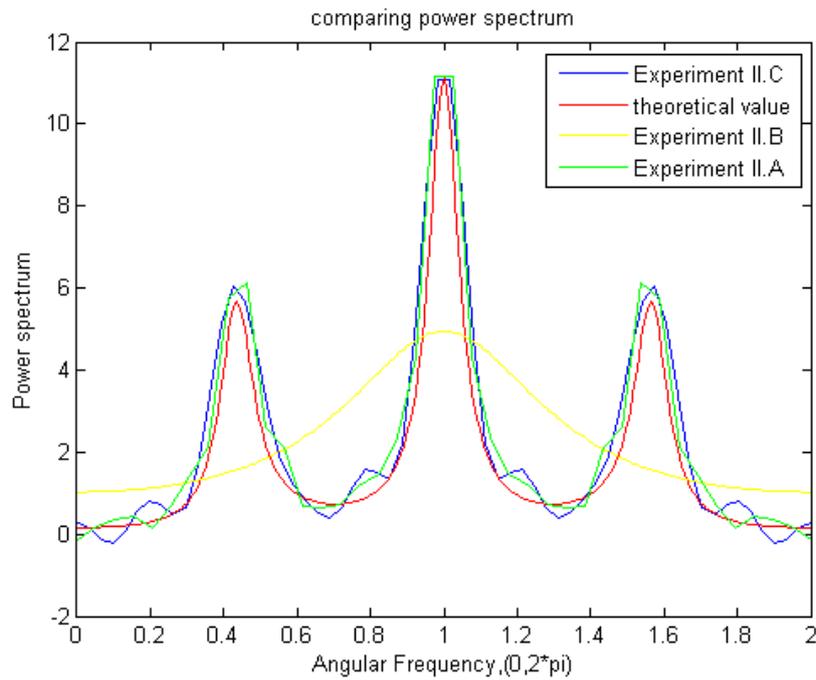


Figure 3.7: Experiment II, Case 2

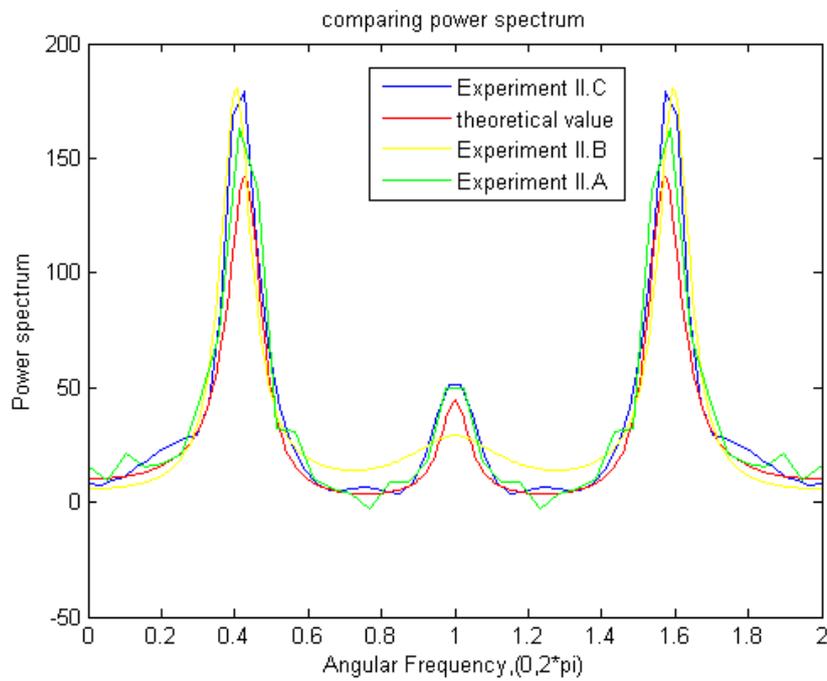


Figure 3.8: Experiment II, Case 3

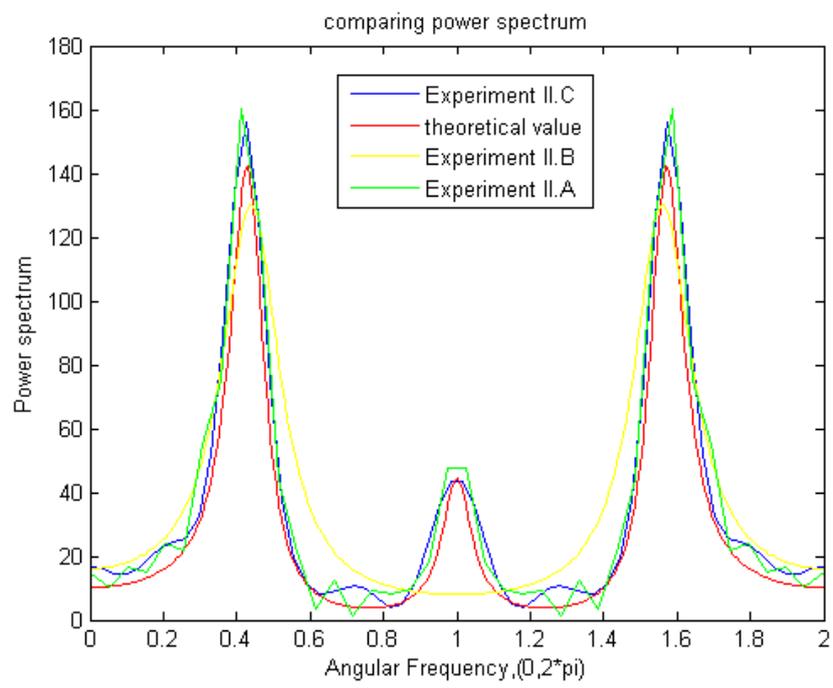


Figure 3.9: Experiment II, Case 4



# Parametric Estimation of Power Spectrum using AR model

---

# 4

In this section, to continue the study of parametric power spectrum estimation, we will introduce other methods to estimate the power spectrum from the compressive samples. The main idea of the methods is introduced as follows. First, we choose the proper model to describe the signal. Here we assume that we have already known what kind of model is selected to generate the original signal. Then, we estimate the parameters in the model directly, instead of estimating the autocorrelation of the original signal first. After obtaining the model parameters, we can calculate the corresponding power spectrum of the original signal. As we can see, the key steps there are how to estimate the parameters of the model we used to describe the original signal. We adopt two methods in this chapter to achieve it. One method is Newton Raphson method, while the other method is using nonlinear least square error algorithm. When we implement those methods, we focus on AR model first. Then, we extend the approach for ARMA model. The final purpose here is to reduce the compression rate compared to the one offered by the minimal sparse ruler in previous chapter. We use the sampling pattern with minimal sparse ruler at first. After estimating the parameters, we attempt to change the sampling pattern that leads to a less number of samples than the one produced by the minimal sparse ruler. Then, we compare those two estimated power spectrum.

## 4.1 Implementation of the methods in AR model

As we know, once we obtain the autocorrelation of the signal, we can calculate the power spectrum of the signal according to (2.17). On the other hand, once we obtain the power spectrum of the signal, we can obtain the autocorrelation of the signal as well and the equation is given by

$$r_x[n] = \frac{1}{2N-1} \sum_{k=-N+1}^{N-1} P_x(e^{jw_k}) e^{jw_k n}, \quad (4.1)$$

with  $w_k = \frac{2\pi}{2N-1}k$ . In this chapter, the signal we used has a real value. Therefore, the autocorrelation has a real value as well, which indicates  $P(e^{jw_k}) = P(e^{-jw_k})$ . In addition, according to Euler's formula, we can obtain

$$e^{j\frac{2\pi}{2N-1}nk} + e^{-j\frac{2\pi}{2N-1}nk} = 2\cos\left(\frac{2\pi}{2N-1}nk\right). \quad (4.2)$$

Then (4.1) can be rewritten as

$$r_x[n] = \frac{1}{2N-1} \left( P_x(e^{jw_0}) + \sum_{k=1}^{N-1} P_x(e^{jw_k}) 2\cos\left(\frac{2\pi}{2N-1}nk\right) \right). \quad (4.3)$$

The power spectrum of the AR model signal is introduced as follows.

$$P_x(e^{jw_k}) = \sigma^2 \frac{|b[0]|^2}{|1 + \sum_{l=1}^p a[l]e^{-jlw_k}|^2} \quad (4.4)$$

where  $\sigma^2$  (that is set to 1 here) is the variance of the white noise used to generate the WSS signal.

Since we use the AR model to model the signal, we can plug (4.4) into (4.1) to obtain the relation between autocorrelation and the parameters in the AR model, which is given by

$$r_x[n] = \frac{1}{2N-1} \sum_{k=-N+1}^{N-1} \frac{|b[0]|^2}{|1 + \sum_{l=1}^p a[l]e^{-jlw_k}|^2} e^{jnw_k}, \quad (4.5)$$

with  $w_k = \frac{2\pi}{2N-1}k$ . Then in the AR model, (4.3) can be expressed as

$$r_x[n] = \frac{1}{2N-1} \left( \frac{|b[0]|^2}{|1 + \sum_{l=1}^p a[l]|^2} + \sum_{k=1}^{N-1} \frac{|b[0]|^2}{|1 + \sum_{l=1}^p a[l]e^{(-jl\frac{2\pi}{2N-1}k)}|^2} 2\cos\left(\frac{2\pi}{2N-1}nk\right) \right). \quad (4.6)$$

Based on the above derivations, the autocorrelations of generated signal can be ex-

pressed as

$$\begin{aligned}
\mathbf{r}_x(\mathbf{a}, b[0]) &= \begin{bmatrix} r_x[0] \\ r_x[1] \\ \vdots \\ r_x[N-1] \\ r_x[1-N] \\ \vdots \\ r_x[-1] \end{bmatrix} = \frac{1}{2N-1} \left( \frac{|b[0]|^2}{|1 + \sum_{l=1}^p a(l)|^2} \mathbf{1}_{2N-1} \right. \\
&+ 2 \begin{bmatrix} 1 & 1 & \dots & 1 \\ \cos(\frac{2\pi}{2N-1}) & \cos(\frac{2\pi}{2N-1}2) & \dots & \cos(\frac{2\pi}{2N-1}(N-1)) \\ \vdots & \vdots & \ddots & \vdots \\ \cos(\frac{2\pi}{2N-1}(N-1)) & \cos(\frac{2\pi}{2N-1}(N-1)2) & \dots & \cos(\frac{2\pi}{2N-1}(N-1)(N-1)) \\ \cos(\frac{2\pi}{2N-1}(1-N)) & \cos(\frac{2\pi}{2N-1}(1-N)2) & \dots & \cos(\frac{2\pi}{2N-1}(1-N)(N-1)) \\ \vdots & \vdots & \ddots & \vdots \\ \cos(\frac{2\pi}{2N-1}(-1)) & \cos(\frac{2\pi}{2N-1}(-1)2) & \dots & \cos(\frac{2\pi}{2N-1}(-1)(N-1)) \end{bmatrix} \\
&\left. \begin{bmatrix} \frac{|b[0]|^2}{|1 + \sum_{l=1}^p a(l)e^{-jl\frac{2\pi}{2N-1}}|^2} \\ \frac{|b[0]|^2}{|1 + \sum_{l=1}^p a(l)e^{-jl\frac{2\pi}{2N-1}2}|^2} \\ \vdots \\ \frac{|b[0]|^2}{|1 + \sum_{l=1}^p a(l)e^{-jl\frac{2\pi}{2N-1}(N-1)}|^2} \end{bmatrix} \right), \quad (4.7)
\end{aligned}$$

where  $\mathbf{1}_{2N-1}$  is a  $(2N-1) \times 1$  vector containing 1 in all of its entries. For expressing it more succinctly, we rewrite the equation as

$$\mathbf{r}_x(\mathbf{a}, b[0]) = \frac{1}{2N-1} (\alpha_{AR}(\mathbf{a}, b[0]) + 2\mathbf{M}_{cos}\mathbf{p}_{AR}(\mathbf{a}, b[0])), \quad (4.8)$$

where  $\alpha_{AR}(\mathbf{a}, b[0])$ ,  $\mathbf{M}_{cos}$  and  $\mathbf{p}_{AR}(\mathbf{a}, b[0])$  present the corresponding matrix or vectors in (4.7). As we already know that  $\mathbf{r}_y[0]$  is a linear function of  $\mathbf{r}_x$  given by (2.14), we can insert (4.8) into (2.14). Then, we can write  $\mathbf{r}_y[0]$  in (2.14) as

$$\mathbf{r}_y[0] = \mathbf{R}_c \mathbf{r}_x(\mathbf{a}, b[0]) = \mathbf{R}_c \frac{1}{2N-1} (\alpha_{AR}(\mathbf{a}, b[0]) + 2\mathbf{M}_{cos}\mathbf{p}_{AR}(\mathbf{a}, b[0])). \quad (4.9)$$

#### 4.1.1 Two estimation methods

In order to compute the estimate for the parameters  $\mathbf{a}$  and  $b[0]$  in the AR model, we employ two methods. One is using Newton Raphson method, while the other one is using nonlinear least squares. First of all, we have to calculate the autocorrelation of the compressive samples defined as  $\hat{\mathbf{r}}_y[0]$ . After that, we will use  $\hat{\mathbf{r}}_y[0]$  to estimate the parameters of the AR model.

### 4.1.2 Newton Raphson method

As we know, Newton Raphson method is a method for finding successively better approximations for the roots of a real valued function [30],

$$x : f(x) = 0 \quad (4.10)$$

Given a function  $f(x)$  over a real variable  $x$  and its derivative  $f'(x)$ , we begin with a first guess  $x_0$  for a root of the function  $f(x)$ . Then, a better approximation  $x_1$  can be obtained by the equation

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}. \quad (4.11)$$

Then we repeat the process as

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \quad (4.12)$$

where  $x_n$  is the  $n$ -th approximation. When the last approximation achieves the error we can tolerate, the process will end. This is the definition of the Newton Raphson method for one dimensional problem. We can extend the problem to multiple dimension Newton Raphson method since the principle is the same as in the one-dimension Newton Raphson method. We also need to initially guess a root for the vector-valued functions  $\mathbf{F}(\mathbf{x})$  given by  $\mathbf{x}_0$ . Then the repeated process is shown as

$$\mathbf{x}_{n+1} = \mathbf{x}_n - ((\mathbf{F}'(\mathbf{x}_n))^H (\mathbf{F}'(\mathbf{x}_n)))^{-1} (\mathbf{F}'(\mathbf{x}_n))^H \mathbf{F}(\mathbf{x}_n), \quad (4.13)$$

where  $\mathbf{x}_n$  is the  $n$ -th approximations, and  $\mathbf{F}'(\mathbf{x})$  is the Jacobian matrix of  $\mathbf{F}(\mathbf{x})$ . In our case, we define the function as

$$\mathbf{F}(\mathbf{a}, b[0]) = \hat{\mathbf{r}}_y[0] - \mathbf{R}_c \frac{1}{2N-1} (\alpha_{AR}(\mathbf{a}, b[0]) + 2\mathbf{M}_{cos}\mathbf{p}_{AR}(\mathbf{a}, b[0])), \quad (4.14)$$

where  $\mathbf{a}, b[0]$  are the roots we want to obtain. Moreover, we can also obtain the Jacobian matrix  $\mathbf{F}'(\mathbf{a}, b[0])$ , which is expressed as

$$\mathbf{F}'(\mathbf{a}, b[0]) = -\mathbf{R}_c \mathbf{r}'_x(\mathbf{a}, b[0]), \quad (4.15)$$

where  $\mathbf{r}'_x(\mathbf{a}, b[0])$  is the Jacobian matrix of  $\mathbf{r}_x(\mathbf{a}, b[0])$ , expressed as

$$\mathbf{r}'_x(\mathbf{a}, b[0]) = \begin{bmatrix} \frac{\partial r_x[0]}{\partial a[1]} & \frac{\partial r_x[0]}{\partial a[2]} & \cdots & \frac{\partial r_x[0]}{\partial a[p]} & \frac{\partial r_x[0]}{\partial b[0]} \\ \frac{\partial r_x[1]}{\partial a[1]} & \frac{\partial r_x[1]}{\partial a[2]} & \cdots & \frac{\partial r_x[1]}{\partial a[p]} & \frac{\partial r_x[1]}{\partial b[0]} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{\partial r_x[N-1]}{\partial a[1]} & \frac{\partial r_x[N-1]}{\partial a[2]} & \cdots & \frac{\partial r_x[N-1]}{\partial a[p]} & \frac{\partial r_x[N-1]}{\partial b[0]} \\ \frac{\partial r_x[1-N]}{\partial a[1]} & \frac{\partial r_x[1-N]}{\partial a[2]} & \cdots & \frac{\partial r_x[1-N]}{\partial a[p]} & \frac{\partial r_x[1-N]}{\partial b[0]} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{\partial r_x[-1]}{\partial a[1]} & \frac{\partial r_x[-1]}{\partial a[2]} & \cdots & \frac{\partial r_x[-1]}{\partial a[p]} & \frac{\partial r_x[-1]}{\partial b[0]} \end{bmatrix}. \quad (4.16)$$

It is easy to find the expression of the elements in  $\mathbf{r}'_x(\mathbf{a}, b[0])$  as

$$\frac{\partial r_x[n]}{\partial a[l]} = \frac{1}{2N-1} \left( \frac{-2|b[0]|^2}{(1 + \sum_{l=1}^p a[l])^3} + 2 \sum_{k=1}^{N-1} \cos\left(\frac{2\pi}{2N-1}nk\right) \frac{\partial P_x(e^{jw_k})}{\partial a[l]} \right), \quad (4.17)$$

where  $\frac{\partial P_x(e^{jw_k})}{\partial a[l]}$  is expressed as

$$\begin{aligned} \frac{\partial P_x(e^{jw_k})}{\partial a[l]} = & - \frac{|b[0]|^2}{((1 + \sum_{l=1}^p a[l] \cos(\frac{2\pi}{2N-1}kl))^2 + (\sum_{l=1}^p a[l] \sin(\frac{2\pi}{2N-1}kl))^2)^2} \\ & \left( 2 \left( 1 + \sum_{l=1}^p a[l] \cos\left(\frac{2\pi}{2N-1}kl\right) \right) \cos\left(\frac{2\pi}{2N-1}kl\right) \right. \\ & \left. + 2 \left( \sum_{l=1}^p a[l] \sin\left(\frac{2\pi}{2N-1}kl\right) \right) \sin\left(\frac{2\pi}{2N-1}kl\right) \right). \end{aligned} \quad (4.18)$$

And the  $\frac{\partial r_x[n]}{\partial b[0]}$  is expressed as

$$\frac{\partial r_x[n]}{\partial b[0]} = \frac{1}{2N-1} \left( \frac{2|b[0]|}{(1 + \sum_{l=1}^p a[l])^2} + 2 \sum_{k=1}^{N-1} \cos\left(\frac{2\pi}{2N-1}nk\right) \frac{\partial P_x(e^{jw_k})}{\partial b[0]} \right), \quad (4.19)$$

where  $\frac{\partial P_x(e^{jw_k})}{\partial b[0]}$  is expressed as

$$\frac{\partial P_x(e^{jw_k})}{\partial b[0]} = \frac{2|b[0]|}{((1 + \sum_{l=1}^p a[l] \cos(\frac{2\pi}{2N-1}kl))^2 + (\sum_{l=1}^p a[l] \sin(\frac{2\pi}{2N-1}kl))^2)}. \quad (4.20)$$

Then, we can estimate the parameters by Newton Raphson method as showed in 4.12.

### 4.1.3 Nonlinear Least Squares

Besides Newton Raphson method, we also implement the Nonlinear Least Squares as well. In this method, we also need to calculate  $\hat{\mathbf{r}}_y[0]$  at first. From (4.7), the estimated parameters can be obtained from

$$\min_{\mathbf{a}, b[0]} \left\| \mathbf{r}_y[0] - \mathbf{R}_c \frac{1}{2N-1} (\alpha_{AR}(\mathbf{a}, b[0]) + 2\mathbf{M}_{\cos} \mathbf{p}_{AR}(\mathbf{a}, b[0])) \right\|_2^2. \quad (4.21)$$

However, what we have known is only the estimated  $\hat{\mathbf{r}}_y[0]$ , so we replace  $\mathbf{r}_y[0]$  in (4.21) with  $\hat{\mathbf{r}}_y[0]$ . Then we obtain

$$\min_{\mathbf{a}, b[0]} \|\mathbf{f}(\mathbf{a}, b[0])\|_2^2, \quad (4.22)$$

where

$$\mathbf{f}(\mathbf{a}, b[0]) = \hat{\mathbf{r}}_y[0] - \mathbf{R}_c \frac{1}{2N-1} (\alpha_{AR}(\mathbf{a}, b[0]) + 2\mathbf{M}_{\cos} \mathbf{p}_{AR}(\mathbf{a}, b[0])). \quad (4.23)$$

Then, we can use the optimization tools in Matlab to estimate the parameters. After we obtain the vector-valued function  $\mathbf{f}(\mathbf{a}, b[0])$ , we can write this optimization problem as

$$\min_{\mathbf{a}, b[0]} \|\mathbf{f}(\mathbf{a}, b[0])\|_2^2 = \min_{\mathbf{a}, b[0]} (f_1(\mathbf{a}, b[0])^2 + f_2(\mathbf{a}, b[0])^2 + \cdots + f_n(\mathbf{a}, b[0])^2) \quad (4.24)$$

where  $f_1(\mathbf{a}, b[0]), f_2(\mathbf{a}, b[0]), \dots, f_n(\mathbf{a}, b[0])$  are elements of  $\mathbf{f}(\mathbf{a}, b[0])$ .

After that, we can use the optimization tools in Matlab to estimate the parameters. To solve this, the trust region algorithm is implemented. The trust region approach is strongly associated with approximation. Assume that we have a current guess of the solution of the optimization problem, an approximate model can be constructed nearby the current point. A solution of the approximate model can be taken as the next iteration point. In a trust region algorithm, the approximate model is only trusted in a region nearby the current iterate. This seems reasonable, because for general nonlinear functions, local approximating models (such as linear approximation and quadratic approximation) can only fit the original function locally. The region where the approximating model is trusted is called the trust region. A trust region is normally a neighbourhood centered at the current iterate. The trust region is adjusted from iteration to iteration. Roughly speaking, if the computation indicates that the approximating model fits the original problem well, the trust region can be enlarged. Otherwise, when the approximating model does not work well enough (for example, a solution of the approximating model turns out to be a bad point), the trust region should be reduced [31].

#### 4.1.4 Simulation result

In our simulation study, the variance of the white Gaussian noise used to generate the signal is set to 1. The parameters in the AR model filter are  $\mathbf{a} = [1, 0.5, 0.4, 0.3]^T$  and  $b[0] = 2$ , respectively. First of all, we estimate the parameters in the AR model from the compressive signals which are obtained by sampling using sampling pattern based on minimal sparse ruler. The estimated parameters in AR model are  $\hat{\mathbf{a}} = [1, 0.5013, 0.3916, 0.3048]^T$  and  $\hat{b}[0] = 2.0041$ . We obtain the same results from the experiments using the Newton Raphson method and using the Nonlinear Least Square Error method. Then, we changed the sampling pattern such that we have less samples than the number of samples that is produced by sampling pattern based on optimal sparse ruler. The results produced by two methods are still same. The estimated parameters in this case are  $\hat{\mathbf{a}} = [1, 0.5008, 0.4005, 0.2961]^T$  and  $\hat{b}[0] = 2.0056$ . The estimated power spectrums for all the parameters are showed in the Fig 4.1.

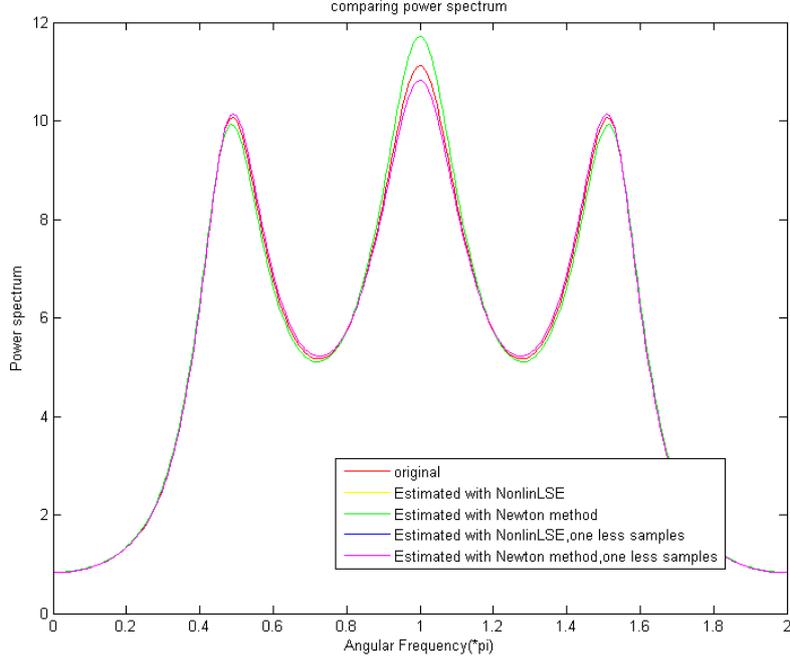


Figure 4.1: Comparison of power spectrums in AR model

## 4.2 Implementation of the methods in ARMA model

After implementing the methods in AR model, now we extend the model to ARMA model. First of all, we generate the signal by using a white Gaussian noise signal passed into an ARMA model filter. Because we do not need to obtain the variance of the white Gaussian noise signal, we assume that the variance of the noise is 1. Then we estimate the parameters in ARMA model from the compressed signal. After that we can calculate the estimated power spectrum of the generated signal based on the parameters we obtained. As expressed in AR model, we can obtain the autocorrelations of the signal when we know its power spectrum (4.1). In our case, the values of the signal are real, which implies that the autocorrelations are real. So we can also adopt (4.3) here. In the context of ARMA model, we can rewrite (4.1) as

$$r_x[n] = \frac{1}{2N-1} \sum_{k=-N+1}^{N-1} \frac{|\sum_{l=0}^q b[l]e^{-jlw_k}|^2}{|1 + \sum_{l=1}^p a[l]e^{-jlw_k}|^2} e^{jnw_k}, \quad (4.25)$$

where  $w_k = \frac{2\pi}{2N-1}k$ . After employing (4.2), the equation can be rewritten as

$$r_x[n] = \frac{1}{2N-1} \left( \frac{|\sum_{l=0}^q b[l]|^2}{|1 + \sum_{l=1}^p a[l]|^2} + \sum_{k=1}^{N-1} \frac{|\sum_{l=0}^q b[l]e^{-jlw_k}|^2}{|1 + \sum_{l=1}^p a[l]e^{-jlw_k}|^2} 2\cos\left(\frac{2\pi}{2N-1}nk\right) \right). \quad (4.26)$$

Based on the above derivations, the autocorrelations of generated signal can be expressed as

$$\begin{aligned}
\mathbf{r}_x(\mathbf{a}, \mathbf{b}) &= \begin{bmatrix} r_x[0] \\ r_x[1] \\ \vdots \\ r_x[N-1] \\ r_x[1-N] \\ \vdots \\ r_x[-1] \end{bmatrix} = \frac{1}{2N-1} \left( \frac{|\sum_{l=0}^q b[l]|^2}{|1 + \sum_{l=1}^p a[l]|^2} \mathbf{1}_{2N-1} \right. \\
&+ 2 \begin{bmatrix} 1 & 1 & \dots & 1 \\ \cos(\frac{2\pi}{2N-1}) & \cos(\frac{2\pi}{2N-1}2) & \dots & \cos(\frac{2\pi}{2N-1}(N-1)) \\ \vdots & \vdots & \ddots & \vdots \\ \cos(\frac{2\pi}{2N-1}(N-1)) & \cos(\frac{2\pi}{2N-1}(N-1)2) & \dots & \cos(\frac{2\pi}{2N-1}(N-1)(N-1)) \\ \cos(\frac{2\pi}{2N-1}(1-N)) & \cos(\frac{2\pi}{2N-1}(1-N)2) & \dots & \cos(\frac{2\pi}{2N-1}(1-N)(N-1)) \\ \vdots & \vdots & \ddots & \vdots \\ \cos(\frac{2\pi}{2N-1}(-1)) & \cos(\frac{2\pi}{2N-1}(-1)2) & \dots & \cos(\frac{2\pi}{2N-1}(-1)(N-1)) \end{bmatrix} \\
&\left. \begin{bmatrix} \frac{|\sum_{l=0}^q b[l]e^{-jl\frac{2\pi}{2N-1}}|^2}{|1 + \sum_{l=1}^p a[l]e^{-jl\frac{2\pi}{2N-1}}|^2} \\ \frac{|\sum_{l=0}^q b[l]e^{-jl\frac{2\pi}{2N-1}2}|^2}{|1 + \sum_{l=1}^p a[l]e^{-jl\frac{2\pi}{2N-1}2}|^2} \\ \vdots \\ \frac{|\sum_{l=0}^q b[l]e^{-jl\frac{2\pi}{2N-1}(N-1)}|^2}{|1 + \sum_{l=1}^p a[l]e^{-jl\frac{2\pi}{2N-1}(N-1)}|^2} \end{bmatrix} \right). \tag{4.27}
\end{aligned}$$

For expressing it more succinctly, we rewrite (4.27) as

$$\mathbf{r}_x(\mathbf{a}, \mathbf{b}) = \frac{1}{2N-1} (\alpha_{ARMA}(\mathbf{a}, \mathbf{b}) + 2\mathbf{M}_{cos}\mathbf{p}_{ARMA}(\mathbf{a}, \mathbf{b})), \tag{4.28}$$

where  $\alpha_{ARMA}(\mathbf{a}, \mathbf{b})$ ,  $\mathbf{M}_{cos}$  and  $\mathbf{p}_{ARMA}(\mathbf{a}, \mathbf{b})$  present the corresponding matrix or vectors in (4.27). As we already know that  $\mathbf{r}_y[0]$  is a linear function of  $\mathbf{r}_x$  given by (2.14), we can insert (4.28) into (2.14). Then, we write  $\mathbf{r}_y[0]$  as a function of the parameters in the ARMA model, i.e.,

$$\mathbf{r}_y[0] = \mathbf{R}_c \mathbf{r}_x(\mathbf{a}, \mathbf{b}) = \mathbf{R}_c \frac{1}{2N-1} (\alpha_{ARMA}(\mathbf{a}, \mathbf{b}) + 2\mathbf{M}_{cos}\mathbf{p}_{ARMA}(\mathbf{a}, \mathbf{b})). \tag{4.29}$$

#### 4.2.1 Newton Raphson method

The principle of the Newton Raphson method has been introduced in section 4.1.2. Now, the model used to generate the signal is changed. The most significant change

from the Newton's method in AR model is the roots. The roots we are going to obtain becomes  $\mathbf{a}, \mathbf{b}$ . Then, the function we defined turn to be

$$\mathbf{F}(\mathbf{a}, \mathbf{b}) = \hat{\mathbf{r}}_y[0] - \mathbf{R}_c \frac{1}{2N-1} (\alpha_{ARMA}(\mathbf{a}, \mathbf{b}) + 2\mathbf{M}_{cos} \mathbf{p}_{ARMA}(\mathbf{a}, \mathbf{b})). \quad (4.30)$$

We can follow the derivation in section 4.1.2, where the Jacobian matrix  $\mathbf{F}'(\mathbf{a}, \mathbf{b})$  can be expressed using (4.15), However  $\mathbf{r}'_x(\mathbf{a}, \mathbf{b})$  here is not the same as the one in 4.15. Here we have

$$\mathbf{r}'_x(\mathbf{a}, \mathbf{b}) = \begin{bmatrix} \frac{\partial r_x[0]}{\partial a[1]} & \frac{\partial r_x[0]}{\partial a[2]} & \cdots & \frac{\partial r_x[0]}{\partial a[p]} & \frac{\partial r_x[0]}{\partial b[0]} & \frac{\partial r_x[0]}{\partial b[1]} & \cdots & \frac{\partial r_x[0]}{\partial b[q]} \\ \frac{\partial r_x[1]}{\partial a[1]} & \frac{\partial r_x[1]}{\partial a[2]} & \cdots & \frac{\partial r_x[1]}{\partial a[p]} & \frac{\partial r_x[1]}{\partial b[0]} & \frac{\partial r_x[1]}{\partial b[1]} & \cdots & \frac{\partial r_x[1]}{\partial b[q]} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial r_x[N-1]}{\partial a[1]} & \frac{\partial r_x[N-1]}{\partial a[2]} & \cdots & \frac{\partial r_x[N-1]}{\partial a[p]} & \frac{\partial r_x[N-1]}{\partial b[0]} & \frac{\partial r_x[N-1]}{\partial b[1]} & \cdots & \frac{\partial r_x[N-1]}{\partial b[q]} \\ \frac{\partial r_x[1-N]}{\partial a[1]} & \frac{\partial r_x[1-N]}{\partial a[2]} & \cdots & \frac{\partial r_x[1-N]}{\partial a[p]} & \frac{\partial r_x[1-N]}{\partial b[0]} & \frac{\partial r_x[1-N]}{\partial b[1]} & \cdots & \frac{\partial r_x[1-N]}{\partial b[q]} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial r_x[-1]}{\partial a[1]} & \frac{\partial r_x[-1]}{\partial a[2]} & \cdots & \frac{\partial r_x[-1]}{\partial a[p]} & \frac{\partial r_x[-1]}{\partial b[0]} & \frac{\partial r_x[-1]}{\partial b[1]} & \cdots & \frac{\partial r_x[-1]}{\partial b[q]} \end{bmatrix}. \quad (4.31)$$

It is easy to find the expression of the elements in  $\mathbf{r}'_x(\mathbf{a}, \mathbf{b})$  as

$$\frac{\partial r_x[n]}{\partial a[l]} = \frac{1}{2N-1} \left( \frac{-2|\sum_{l=0}^q b[l]|^2}{(1 + \sum_{l=1}^p a[l])^3} + 2 \sum_{k=1}^{N-1} \cos\left(\frac{2\pi}{2N-1}nk\right) \frac{\partial P_x(e^{jw_k})}{\partial a[l]} \right), \quad (4.32)$$

where  $\frac{\partial P_x(e^{jw_k})}{\partial a[l]}$  is expressed as

$$\begin{aligned} \frac{\partial P_x(e^{jw_k})}{\partial a[l]} &= - \frac{(\sum_{l=0}^q b[l] \cos(\frac{2\pi}{2N-1}kl))^2 + (\sum_{l=0}^q b[l] \sin(\frac{2\pi}{2N-1}kl))^2}{((1 + \sum_{l=1}^p a[l] \cos(\frac{2\pi}{2N-1}kl))^2 + (\sum_{l=1}^p a[l] \sin(\frac{2\pi}{2N-1}kl))^2)^2} \\ &\quad \left( 2 \left( 1 + \sum_{l=1}^p a[l] \cos\left(\frac{2\pi}{2N-1}kl\right) \right) \cos\left(\frac{2\pi}{2N-1}kl\right) \right. \\ &\quad \left. + 2 \left( \sum_{l=1}^p a[l] \sin\left(\frac{2\pi}{2N-1}kl\right) \right) \sin\left(\frac{2\pi}{2N-1}kl\right) \right). \end{aligned} \quad (4.33)$$

And the  $\frac{\partial r_x[n]}{\partial b[l]}$  is expressed as

$$\frac{\partial r_x[n]}{\partial b[l]} = \frac{1}{2N-1} \left( \frac{2|\sum_{l=0}^q b[l]|}{(1 + \sum_{l=1}^p a[l])^2} + 2 \sum_{k=1}^{N-1} \cos\left(\frac{2\pi}{2N-1}nk\right) \frac{\partial P_x(e^{jw_k})}{\partial b[l]} \right), \quad (4.34)$$

where  $\frac{\partial P_x(e^{jw_k})}{\partial b[l]}$  is expressed as

$$\frac{\partial P_x(e^{jw_k})}{\partial b[l]} = \frac{2(b[0] + \sum_{l=1}^q b[l] \cos(\frac{2\pi}{2N-1}kl)) \cos(\frac{2\pi}{2N-1}kl) + 2(\sum_{l=1}^q b[l] \sin(\frac{2\pi}{2N-1}kl)) \sin(\frac{2\pi}{2N-1}kl)}{((1 + \sum_{l=1}^p a[l] \cos(\frac{2\pi}{2N-1}kl))^2 + (\sum_{l=1}^p a[l] \sin(\frac{2\pi}{2N-1}kl))^2)}. \quad (4.35)$$

Then, we can estimate the parameters by Newton Raphson method.

## 4.2.2 Nonlinear Least Square Error estimation

Since we have known (4.27) already, the estimated parameter can be obtained when it satisfies the condition as

$$\min_{\mathbf{a}, \mathbf{b}} \left\| \mathbf{r}_y[0] - \mathbf{R}_c \frac{1}{2N-1} (\alpha_{ARMA}(\mathbf{a}, \mathbf{b}) + 2\mathbf{M}_{cos} \mathbf{p}_{ARMA}(\mathbf{a}, \mathbf{b})) \right\|_2^2. \quad (4.36)$$

However, what we have known is only the estimated  $\hat{\mathbf{r}}_y[0]$ , so we replace  $\mathbf{r}_y[0]$  in (4.21) with  $\hat{\mathbf{r}}_y[0]$ . Then we obtain

$$\min_{\mathbf{a}, \mathbf{b}} \|\mathbf{f}(\mathbf{a}, \mathbf{b})\|_2^2 \quad (4.37)$$

where

$$\mathbf{f}(\mathbf{a}, \mathbf{b}) = \hat{\mathbf{r}}_y[0] - \mathbf{R}_c \frac{1}{2N-1} (\alpha_{ARMA}(\mathbf{a}, \mathbf{b}) + 2\mathbf{M}_{cos} \mathbf{p}_{ARMA}(\mathbf{a}, \mathbf{b})). \quad (4.38)$$

Then, we can use the optimization tools in Matlab to estimate the parameters. After we obtain the vector-valued function  $\mathbf{f}(\mathbf{a}, \mathbf{b})$ , we can write this optimization problem as

$$\min_{\mathbf{a}, \mathbf{b}} \|\mathbf{f}(\mathbf{a}, \mathbf{b})\|_2^2 = \min_{\mathbf{a}, \mathbf{b}} (f_1(\mathbf{a}, \mathbf{b})^2 + f_2(\mathbf{a}, \mathbf{b})^2 + \cdots + f_n(\mathbf{a}, \mathbf{b})^2) \quad (4.39)$$

where  $f_1(\mathbf{a}, \mathbf{b}), f_2(\mathbf{a}, \mathbf{b}), \dots, f_n(\mathbf{a}, \mathbf{b})$  are elements of  $\mathbf{f}(\mathbf{a}, \mathbf{b})$ . The optimization problem is solved using trust region algorithm, which is the same used in AR model case.

## 4.2.3 Simulation result

In this simulation study, the variance of the white Gaussian noise used to generate the signal is set to 1. The parameters in the ARMA model filter are  $\mathbf{a} = [1, 0.5, 0.4, 0.3]^T$  and  $\mathbf{b} = [1, 0.5]^T$  respectively. First of all, we estimate the parameters in the ARMA model from compressive signals which are obtained by sampling using sampling pattern based on minimal sparse ruler. The estimated parameters in ARMA model are  $\hat{\mathbf{a}} = [1, 0.5447, 0.3860, 0.3249]^T$  and  $\hat{\mathbf{b}} = [0.9971, 0.5507]^T$ . We obtain the same results from the experiments using the Newton Raphson method and using the Nonlinear Least Square Error. Then, we changed the sampling pattern such that we have less samples than the number of samples that is produced by sampling pattern based on optimal sparse ruler. The results produced by two methods are still same. The estimated parameters in this case are  $\hat{\mathbf{a}} = [1, 0.5714, 0.3753, 0.3399]^T$  and  $\hat{\mathbf{b}} = [1.0048, 0.5819]^T$ . The estimated power spectrums for all the parameters are showed in Fig 4.2.

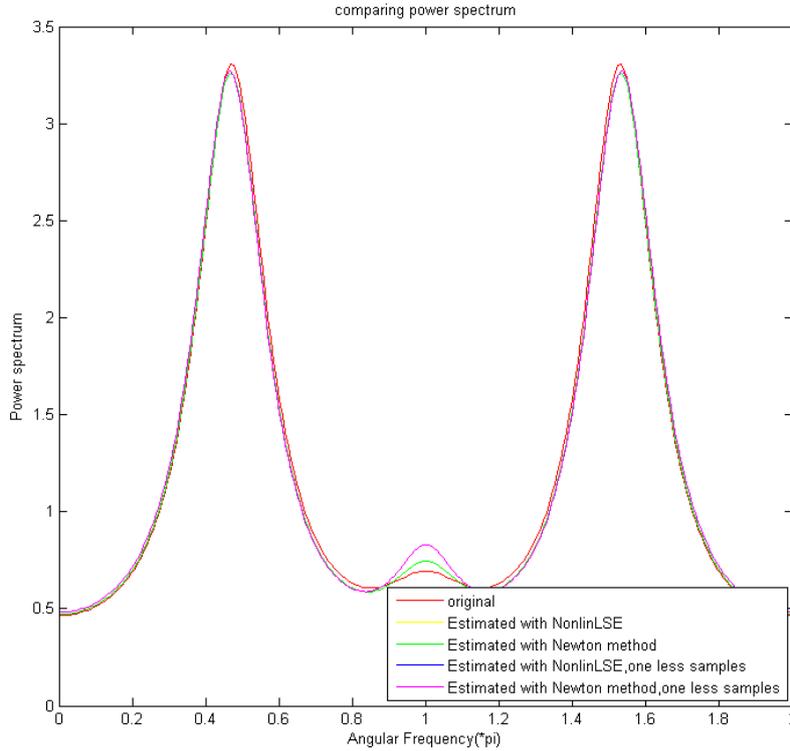


Figure 4.2: Comparison of power spectrums in ARMA model

### 4.3 Conclusion and analysis

Newton Raphson method is one of the methods having fastest convergences to the root, and it is easy to convert for handling multi-dimensions cases. However, this method must meet some conditions. The gradient of the function of interest could not be zero or very close to zero. Besides, since the Newton Raphson method is a local convergence method, we have to carefully pick the starting point. If the initial point is far away from the actual roots, the method may not converge [29]. Different from the Newton Raphson method, the trust region algorithm aims for finding the minima. Trust region algorithms are reliable and robust and they can be applied to ill-conditioned problems. Moreover, they have very strong convergence properties [31]. However, trust region algorithm is also a local optimization algorithm [9], which means the final result we find may not be accurate.

As we can see, those two methods can work well in our implementations. From the figures we can find that we can estimate the power spectrum well with less number of samples than the one produced by optimal sparse ruler. This means that we indeed reduced compressive rate. However, there are still some problems left. The first one is that when we estimate the parameters in ARMA model, we have to use more samples of each segment than we used in AR model. That is because the number of parameters in

ARMA model is more than the one in AR model. The second problem is that, we have to collect more autocorrelations comparing with the autocorrelations we collect in the parametric approach in previous chapter. Because the problem is nonlinear, we have to obtain more equations to estimate the parameters in the models. The last problem is that the two methods we used here are not suitable for all cases, which means that they may fail when the parameters are changed. In Newton Raphson method, one limitation is that the method only works well when the step size is small. Another limitation is that if the Jacobian matrix of the function we defined is singular or poorly conditioned, the problem can not be solved. Compared to Newton's method, the Nonlinear Least Square is better. It can still work in some cases when the Newton's method fails. However, it is not always as good as we want. In some cases, the results of estimation using the Nonlinear Least Square are not accurate. The the Nonlinear Least Square Error method here is based on trust region reflective least square algorithm. The Jacobian matrix is also used here, and we have to avoid that the Jacobian matrix will be singular.

## Sub-optimal sparse ruler

---

In this chapter, we want to optimize the sampling pattern in non-parametric power spectrum estimation. At first we attempt to design a sparse ruler with less compression compared to  $(N - 1)$ -length minimal sparse ruler. It is impossible to achieve the goal if we select from  $N$ -sample blocks, hence, in the sparse ruler we designed, we extend the sampling range from  $N$  to  $W$  to cover the lags within  $N$  lags. Then, we focus on sub-optimal sparse ruler for  $N$  samples. In the current time, there is no approach to obtain an optimal sparse ruler other than a brute force approach. In order to reduce the compression rate for the non-parametric method, we attempt a genetic algorithm. Then, in this section, we attempt to design some sub-optimal sparse ruler algorithms. Compared to the minimal sparse ruler, those algorithms generally lead to a larger number of marks for a given length  $N$  but they consume considerably less amount of time. Therefore, these algorithms might be useful when  $N$  is large.

### 5.1 Attempt to Reduce the Compression

As we know from previous chapters, in order to estimate the autocorrelation lags from 0 up to  $N - 1$ , the minimal number of samples we need to select from the indices range  $[0, 1, \dots, N - 1]$  is  $M$ , where  $M$  is the number of marks in the  $(N - 1)$ -length minimal sparse ruler. It is impossible to further reduce the number of samples if we collect indices samples from 1 up to  $N$  to cover the autocorrelation lags from 0 up to  $N - 1$ . Therefore, we have to make some changes in order to reduce the compression rate. Here we attempt to reduce the compression rate from the one offered by optimal sparse ruler after adding some conditions. We are wondering whether it exists such a case where we can cover the autocorrelation lags from 0 up to  $N - 1$  with less samples when we selected the samples from a larger indices range  $[1, 2, \dots, W]$  with  $W$  larger than  $N$ . It seems that we have more freedom than before. To achieve this, we adopt genetic algorithm to solve the optimal solution in this case.

Genetic algorithm is a search heuristic that mimics the process of natural selection. This heuristic is used to generate useful solutions to optimization. In a genetic algorithm, the candidate solutions are called individuals, the population consists of a set of solutions. In our case, what we want is to cover the autocorrelation lags from 0 up to  $N - 1$  after collecting samples at indices between 1 and  $W$ . Hence, the set of indices that satisfies the condition is the individual. According to our definition, we can notice that the number of samples corresponding to each individual are different. Our purpose is to find the individual that corresponds to minimal number of samples. The genetic algorithm simulates the evolution in nature. First, we build a population by selecting the individual from all the solutions randomly. The number of individuals in a popula-

tion is defined by ourselves. When the number of individuals in the population is too large, it will cost too much running time. However, when the number of individuals in the population is too small, the solution may fall into a local optimal solution and the optimal solution may be missed. After selecting the initial population, we need to adopt some rules to evolve the population toward better solutions. The main methods there are simulating crossover and mutation. As we know, crossover and mutation in nature are processed by exchanging the genes and changing the genes respectively. In our case, we define a gene as a indices pair. For instance, for lag 1, there are  $W - 1$  genes such as  $\{1, 2\}, \{2, 3\}, \dots, \{W - 1, W\}$ . An individual consists of  $N$  genes covering lags from 0 up to  $N - 1$ . The crossover means we need to exchange some genes corresponding to the same lags between two individuals (where the crossover probability is  $P_c$ ), and the mutation means that we need to change some genes corresponding to the same lags (where the mutation probability is  $P_m$ ). Here we notice that in order to operate crossover and mutation, we repeat some indices in the solution. When we find the optimal solution, we need to remove the repeated samples. By so far, we know how to build a population, how to do crossover and mutation operation, but do not know how to evaluate the quality of the individuals. Therefore, we need a fitness function to do it. Fitness function determines the possibility of the individual to be chosen as the next generation. In our case, if the individual in the experiment has less number of elements, it will get a higher fitness value in the evaluation of whether it will be chosen as the next generation or not. In evolution, we also need to use fitness function to decide which individuals can be selected as the next generation. This is the genetic algorithm we adopt. After a number of times of evolution, we can get an optimal solution.

The pseudo-code of the genetic algorithm is given in Algorithm 1.

First of all, we need to build a gene bank. The gene bank consists of  $N$  groups representing  $N$  lags. For example, group-1 represents a collection of pairs of indices selected from  $[0, 1, \dots, W - 1]$ , whose indices difference is 1. See Algorithm 1. Genetic algorithm is a good method to find a optimal solution in many cases, but if we do not choose the proper fitness function and proper parameters such as the scale of the population, the crossover probability and the mutation probability, we may fail to find the optimal solution and just find a local optimal solution. However, after trying many times and trying different parameters, we find that the minimal number of samples we have found is equal to the number of samples in optimal sparse ruler with indices range from 1 up to  $N$ . Although we can not theoretically demonstrate that we can not reduce the compressive rate in our case, with many experiments, it seems that we can not find a less number of samples even if we increase the indices range.

## 5.2 Sub Optimal Sparse Ruler Algorithms

Sampling pattern is one of the key part in compressive power spectrum estimation. The important rule for the sampling pattern is that the desired autocorrelation range should be covered while the number of samples should be as less as possible. Certainly, the minimal sparse ruler sampling pattern offers the best compression. Unfortunately, there exists no quick procedure to find minimal sparse rulers: one must perform a brute-force search over the space of length- $(N - 1)$  sparse rulers to find a minimal one

---

**Algorithm 1** Genetic algorithm for selecting samples

---

Build an  $M$  individuals population named POPULATION, whose individual is made up by randomly selecting an indices pair from each group in gene bank  
**while** Number of samples is larger than what we want **do**  
    Create a fitness function to select the individual  
    Initialize a new  $M$ -individuals population named NEWPOP  
    **for** times= 1 **to** times=  $M/2$  **do**  
        Choose two individuals from POPULATION according to fitness function  
        **if**  $\text{randm}(0,1) < P_c$  **then**  
            Do crossover with two selected individuals  
        **end if**  
        **if**  $\text{randm}(0,1) < P_m$  **then**  
            Do mutation with two selected individuals  
        **end if**  
        Add those two individuals into NEWPOP  
    **end for**  
    Replace POPULATION with NEWPOP  
    Find the individuals in POPULATION with minimal number of samples  
**end while**

---

[22]. Therefore, we design some sub-optimal sparse rulers with certain procedure.

### 5.2.1 The first algorithm (Algorithm Sparse Ruler-1)

1. We define  $S_m$  as the set of the selected indices and first initialize  $S_m$  as  $S_m = \{0, N-1\}$ . We then compute  $\Omega_{S_m}$  as  $\Omega_{S_m} = \{|n_i - n_j| \mid \forall n_i, n_j \in S_m\}$ . We define  $\bar{U}_{S_m}$  as  $\bar{U}_{S_m} = \{0, 1, \dots, N-1\} \setminus \Omega_{S_m}$ .
2. Given  $A = \{a_0, a_1, \dots, a_{N-1}\}$  and  $B = \{b_0, b_1, \dots, b_{N-1}\}$ , we define

$$\begin{aligned} A \oplus B = \{ & a_0 + b_0, a_0 + b_1, \dots, a_0 + b_{N-1}, \\ & a_1 + b_0, a_1 + b_1, \dots, a_1 + b_{N-1}, \dots, \\ & a_{N-1} + b_0, a_{N-1} + b_1, \dots, a_{N-1} + b_{N-1} \} \end{aligned} \quad (5.1)$$

and

$$\begin{aligned} A \ominus B = \{ & a_0 - b_0, a_0 - b_1, \dots, a_0 - b_{N-1}, \\ & a_1 - b_0, a_1 - b_1, \dots, a_1 - b_{N-1}, \dots, \\ & a_{N-1} - b_0, a_{N-1} - b_1, \dots, a_{N-1} - b_{N-1} \}. \end{aligned} \quad (5.2)$$

We then first define

$$\bar{S}_m = (S_m \oplus \bar{U}_{S_m}) \cup (S_m \ominus \bar{U}_{S_m}), \quad (5.3)$$

where  $\bar{S}_m$  is a special set that allows for repeated elements. Also note that we then remove all elements of  $\bar{S}_m$  that are larger than  $N-1$  and smaller than 0. For example, if we have  $S_m = \{3, 5, 7\}$ ,  $N = 11$  and  $\bar{U}_{S_m} = \{2, 6\}$ , then we have

$$\bar{S}_m = \{1, 1, 3, 5, 5, 7, 9, 9\} \quad (5.4)$$

3. Set  $\tilde{n}_g$  to the value that most frequently appears in  $\bar{S}_m$ . If there are more than one most frequent values in  $\bar{S}_m$  then set  $\tilde{n}_g$  to the smallest one. For example, if  $\bar{S}_m = \{1, 2, 2, 3, 3, 4, 4, 5\}$  then  $\tilde{n}_g = 2$ . We then perform  $S_m = S_m \cup \{\tilde{n}_g\}$ .
4. Recompute  $\mathcal{U}_{S_m}$  as  $\mathcal{U}_{S_m} = \{0, 1, \dots, N - 1\} \setminus \Omega_{S_m}$ .
5. If  $\mathcal{U}_{S_m} = \emptyset$ ,  $S_m$  is what we want. Otherwise, we repeat step 2 to step 5.

### 5.2.2 The second algorithm (Algorithm Sparse Ruler-2)

1. We define  $S_m$  as the set of the selected indices and first initialize  $S_m$  as  $S_m = \{0, N - 1\}$ . We then compute  $\Omega_{S_m}$  as  $\Omega_{S_m} = \{|n_i - n_j| | \forall n_i, n_j \in S_m\}$ .
2. We define  $S'_m$  as the set of remaining indices and compute  $S'_m$  as  $S'_m = \{0, 1, 2, \dots, N - 1\} \setminus S_m$ .
3. Let us then find  $\bar{n}_i$  such that:

$$\bar{n}_i = \arg \max_{n_i \in S'_m} |\Omega_{S_m \cup \{n_i\}}|. \quad (5.5)$$

If there are multiple  $\bar{n}_i$  that lead to the same maximum value of  $|\Omega_{S_m \cup \{n_i\}}|$ , then we pick the smallest value of them as our  $\bar{n}_i$ . For example, if  $S'_m = \{1, 2, 3, 4\}$  and  $S_m = \{5, 6\}$ , then  $n_i = 1$ ,  $n_i = 2$  and  $n_i = 3$  will maximize the value of  $|\Omega_{S_m \cup \{n_i\}}|$ , which is  $|\Omega_{S_m \cup \{n_i\}}| = 3$ . In this case, we set  $\bar{n}_i = 1$ . We then perform  $S_m = S_m \cup \{\bar{n}_i\}$  and recompute  $S'_m$  and  $\Omega_{S_m}$ .

4. If  $\{0, 1, 2, \dots, N - 1\} \setminus \Omega_{S_m} = \emptyset$ ,  $S_m$  is what we want. Otherwise, we repeat step 3 to step 5.

### 5.2.3 The convex optimization based algorithm (Algorithm Sparse Ruler-3)

In this algorithm, first of all, we adopt an iterative algorithm [5] to obtain the compressive samples. We define a vector  $\mathbf{v}$  as the transpose of the sum of the rows of sampling matrix  $\mathbf{C}$ . For instance, if the sampling matrix  $\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ , then the

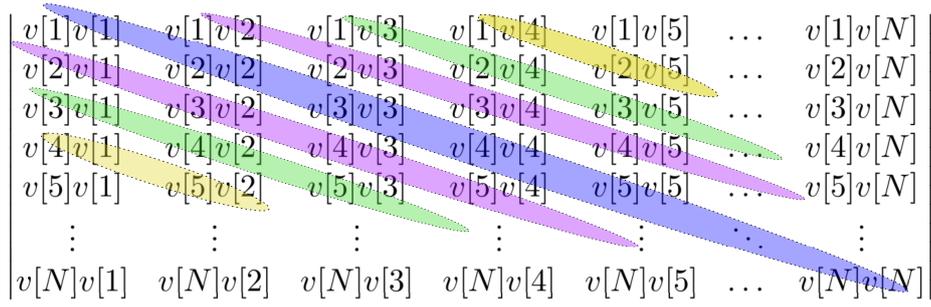


Figure 5.1: Illustration of sum of the diagonals in matrix  $\mathbf{V}$

$\mathbf{v} = [1, 1, 0, 1]^T$ . In other words  $\mathbf{v}$  is an  $N \times 1$  vector, and the indices of the elements with value 1 in  $\mathbf{v}$  constitute the selected indices set  $S_m$ .

From Fig 5.1 we can see that the diagonals of matrix  $\mathbf{V}$ , which is defined as  $\mathbf{V} = \mathbf{v}\mathbf{v}^T$ , indicate autocorrelation lags. For example, if  $v[i]v[j] = 1$ , it means that  $r_x[i - j]$  is collected, otherwise, if  $v[i]v[j] = 0$ , it means that  $r_x[i - j]$  is not collected due to the nature of the sampling pattern. Note that our goal is to minimize the compression rate which is equivalent to minimizing  $\|\mathbf{v}\|_0$ . However, we do relaxation and minimizing  $\|\mathbf{v}\|_1$ . In fact, we try to follow [5] and minimize  $\mathbf{w}^T \mathbf{v}$  where  $\mathbf{w}$  is a  $N \times 1$  weight vector.

The iteration algorithm to find the proper  $\mathbf{v}$  for the sampling matrix is introduced as follows, where  $\mathbf{w} = [w[0], w[1], \dots, w[N - 1]]^T$ .

1. Set the iteration count  $l$  to zero and initialize all the elements in weight vector  $\mathbf{w}^{(l)}$ , with  $l = 0$ , to 1.
2. Solve the weighted minimization problem: We want to minimize  $\mathbf{w}^{(l)T} \mathbf{v}$  over the vector  $\mathbf{v}$ , with the constraint that the index differences set  $\Omega_{S_m}$  covers all integers from  $1 - N$  up to  $N - 1$ .
3. Update the weights  $w[i]$  for each  $i = 0, 1, 2, \dots, N - 1$

$$w^{(l+1)}[i] = \frac{1}{|v^{(l)}[i]| + \epsilon} \quad (5.6)$$

where  $\epsilon$  is a small value in order to provide stability and ensure that a zero-valued component in  $\mathbf{v}^{(l)}$  does not strictly prohibit a nonzero estimate at the next step.

4. Terminate on convergence or when  $(l)$  attains a specified maximum number of iterations  $l_{max}$ . Otherwise, increment  $(l)$  and go to step 2.

Using this iteration algorithm, we can obtain the vector  $\mathbf{v}$  as we expect. However, it is hard to implement the algorithm for the minimization in step 2. To solve this, we adopt convex optimization after relaxing some conditions. From [10] we can find that,

for any symmetric matrix,  $\mathbf{M}$ , of the form  $\mathbf{M} = \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{C} \end{pmatrix}$ , and for invertible matrix  $\mathbf{C}$ , the following properties hold:

1.  $\mathbf{M} \succ 0$  iff  $\mathbf{C} \succ 0$  and  $\mathbf{A} - \mathbf{B}\mathbf{C}^{-1}\mathbf{B}^T \succ 0$ .

2. If  $\mathbf{C} \succ 0$ , then  $\mathbf{M} \succeq 0$  iff  $\mathbf{A} - \mathbf{B}\mathbf{C}^{-1}\mathbf{B}^T \succeq 0$ .

Now we are doing some relaxation on the equation  $\mathbf{V} = \mathbf{v}\mathbf{v}^T$ . Since  $\mathbf{V} - \mathbf{v}\mathbf{v}^T = 0$ , we can relax it to obtain  $\mathbf{V} - \mathbf{v}\mathbf{v}^T \succeq 0$ . Using the properties above, we can obtain

$$\begin{pmatrix} \mathbf{V} & \mathbf{v} \\ \mathbf{v}^T & 1 \end{pmatrix} \succeq 0. \quad (5.7)$$

From Fig 5.1 we know that in order to obtain the index differences cover from  $1 - N$  up to  $N - 1$ , we have to satisfy that for each diagonal, there should be at least one entry with value 1. Here, we use a new convex optimization expression instead of the equation in step 2. The convex optimization expression is as follows.

$$\begin{aligned} & \min_{\mathbf{v}} \mathbf{w}^T \mathbf{v} \\ & \text{subject to} \\ & \mathbf{v} \leq \mathbf{1}_{N \times 1}, \\ & \mathbf{v} \geq \mathbf{0}_{N \times 1}, \end{aligned} \quad (5.8)$$

$$\text{sum}(\text{diag}(\mathbf{V}, i)) \geq 1, \quad \text{for each } i = 1 - N, \dots, N - 1$$

$$\text{diag}(\mathbf{V}, 0) = \mathbf{v},$$

$$\begin{pmatrix} \mathbf{V} & \mathbf{v} \\ \mathbf{v}^T & 1 \end{pmatrix} \succeq 0,$$

where  $\mathbf{1}_{N \times 1}$  is the  $N \times 1$  vector containing 1 in all entries and  $\mathbf{0}_{N \times 1}$  is the  $N \times 1$  vector containing 0 in all entries. The first and second constraints  $\mathbf{v} \leq \mathbf{1}_{N \times 1}$  and  $\mathbf{v} \geq \mathbf{0}_{N \times 1}$  are the relaxation we made to turn the value of the element of  $\mathbf{v}$  from integer value  $\{0, 1\}$  to real value between 0 and 1. The second constraint is to ensure that all the correlation lags from  $1 - N$  to  $N - 1$  are covered. The reason for this has been explained before. The third constraint is to ensure the relation between  $\mathbf{V}$  and  $\mathbf{v}$  after relaxation. The last constraint is the relaxation of  $\mathbf{V} = \mathbf{v}\mathbf{v}^T$ . Now we have a convex optimization problem. The matrix  $\mathbf{V}$  and the vector  $\mathbf{v}$  are to be estimated.

After obtaining the final estimated  $\hat{\mathbf{v}}$ , we notice that the components in  $\hat{\mathbf{v}}$  do not satisfy the condition that all the elements in  $\mathbf{v}$  should have the value 0 or 1. Actually, the elements in  $\hat{\mathbf{v}}$  are between 0 and 1. Therefore, we design two methods to obtain the final  $\mathbf{v}$ . The first one is very easy, we set a threshold to decide the value in  $\mathbf{v}$ . If the value of  $\hat{v}[i]$  is larger than 0.5, then we set the  $v[i]$  to 1, otherwise  $v[i]$  will be zero. The second method is little complicated than the first one. We using randomized rounding algorithm [6]. It is as follows.

1. Generate  $L$  candidate estimates of the form  $v_{n,l} = 1(l = 1, 2, \dots, L)$  with a probability  $\hat{v}_n$  for  $n = 1, 2, \dots, N$ . The  $l$  starts from 1.

2. Define  $\mathbf{v}_l = [v_{1,l}, \dots, v_{N,l}]^T$ . Find the  $S_m$  of the  $\mathbf{v}_l$ .
3. If the index differences set  $\Omega_{S_m}$  of  $S_m$  satisfying the condition that  $\Omega_{S_m}$  covers all integers from  $1 - N$  up to  $N - 1$ ,  $\mathbf{v}_l$  is the solution, otherwise go back to step 2 and move to another  $\mathbf{v}_{l+1}$ .

### 5.3 Comparison with other algorithm

Since we have proposed the sub-optimal algorithms, it is necessary to compare them with the existing sub-optimal algorithms. Here we choose two algorithms, coprime sampling and two levels nested array algorithm, which are also used to reduce the sampling rate.

#### 5.3.1 Coprime Sampling [21]

Consider a set  $I_{M,N}$  of intergers given by  $I_{M,N} = \{Mn, 0 \leq n \leq N - 1\} \cup \{Nm, 0 \leq m \leq M - 1\}$ , where  $M$  and  $N$  are coprime integers. The difference set of this set includes the cross differences

$$S_{N,M} = \{(Mn - Nm), 0 \leq n \leq N - 1, 0 \leq m \leq M - 1\}. \quad (5.9)$$

It is shown that  $S_{M,N}$  consist of exactly  $MN$  distinct integers in the range  $[-N(M - 1) \quad M(N - 1)]$  but they are not in a continuous range. In order to cover integers  $k$  continuously in the range  $-MN \leq k \leq MN$ , we need to meet the condition that  $0 \leq n \leq N - 1$  and  $0 \leq m \leq 2M - 1$ , assuming that  $N > M$ .

#### 5.3.2 Two Levels Nested Array [20]

For a two levels nested array, there are two kinds of arrays, inner array and outer array. Since we want to cover the indices difference continuously, we set the space between elements in the inter array to 1. The inter array has  $N_1$  elements with spacing 1 while the outer array has  $N_2$  elements with spacing  $N_1 + 1$ . If we apply it in sampling, the samples can be grouped into two parts. The first part is a sample set  $S_{inner} = \{m, \quad m = 1, 2, \dots, N_1\}$  and the second part is a sample set  $S_{outer} = \{n(N_1 + 1), \quad n = 1, 2, \dots, N_2\}$ . For a given number of samples, the optimal allocation of the inner and outer samples are shown as follow.

$N$	optimal $N_1, N_2$	indices difference
even	$N_1 = N_2 = \frac{1}{2}N$	$\frac{N^2 - 2}{2} + N$
odd	$N_1 = \frac{N - 1}{2}, N_2 = \frac{N + 1}{2}$	$\frac{N^2 - 1}{2} + N$

Table 5.1: Optimal allocation of the inner samples and the outer samples with fixed total  $N$  samples

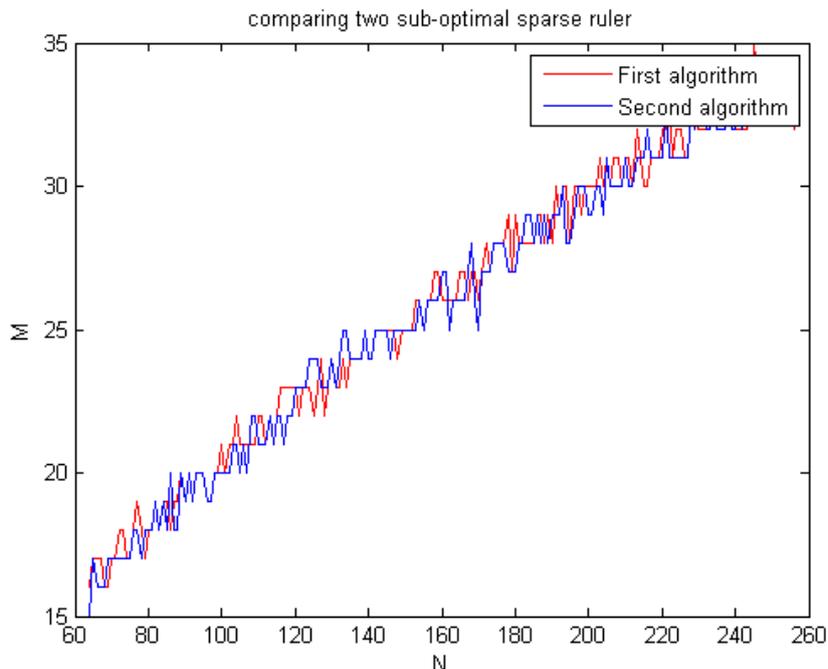


Figure 5.2: Performance of two sub optimal sparse ruler algorithms

## 5.4 Numerical Study

From Fig. 5.2, we can find that the performances of the first and second algorithms we designed are nearly the same. Sometimes the first algorithm works better than the second one, while sometimes the opposite is true. It depends on the value of  $N$ .

We then compare the sub-optimal sparse rulers produced by the those two algorithms with the optimal sparse ruler. It is clear that the resulting sub-optimal sparse rulers generally have a larger number of marks than the corresponding optimal sparse ruler. Thus the compression rate offered by sub-Nyquist sampling using the sub-optimal sparse rulers is weaker than the one offered by optimal sparse ruler based sampling. For example, Table 5.2 shows that, for  $N = 124$ , the two sub-optimal sparse rulers offer compression rate of 0.185 compared to 0.153 offered by the optimal sparse ruler. Note, however, the sub-optimal sparse ruler algorithms offer a smaller amount of computation time which will be useful for a large  $N$ .

When we use the coprime algorithm to cover the lags over a certain continuous range, we also produce some lags that we are not interested in. For example, if we want to cover the lags  $-MN = -15 \leq k \leq MN = 15$ , with  $M = 3$  and  $N = 5$ , we need to collect the samples set  $\{0, 3, 5, 6, 9, 10, 12, 15, 20, 25\}$ , and the indices difference set is  $\{-25, -22, -20, -19, -17, -16\} \cup \{-15, -14, \dots, -1, 0, 1, \dots, 14, 15\} \cup \{16, 17, 19, 20, 22, 25\}$ . So it is impossible to obtain a  $\{-NM, \dots, NM\}$  indices differences set with the samples selected from  $\{1, 2, \dots, NM + 1\}$ . When we use

Table 5.2: Comparison of sub-optimal sparse ruler algorithms and minimal sparse ruler

$N$ samples	lags range	$M$	$M$	$M$	$M$	$M$	$M$
$N$	$N - 1$	first	second	cvx based	coprime	nested	optimal
2	1	2	2	2	2	2	2
3	2	3	3	3	3	3	3
4	3	3	3	3	4	3	3
5	4	4	4	4	6	4	4
6	5	4	4	4	6	4	4
7	6	5	4	4	6	5	4
8	7	5	5	5	6	5	5
9	8	6	5	5	8	5	5
10	9	6	6	5	8	6	5
11	10	6	6	6	8	6	6
12	11	6	6	6	8	6	6
13	12	6	6	6	10	6	6
14	13	7	7	6	10	7	6
15	14	7	7	7	10	7	7
16	15	7	7	7	10	7	7
17	16	7	7	7	10	8	7
18	17	8	8	8	10	8	7
19	18	8	9	8	12	8	8
20	19	8	8	8	12	8	8
21	20	8	8	8	12	9	8
22	21	9	9	8	12	9	8
23	22	9	9	9	12	9	8
24	23	9	9	10	12	9	8
25	24	9	9	9	16	9	9
26	25	9	9	9	16	10	9
27	26	10	10	10	16	10	9
28	27	10	10	10	16	10	9
29	28	11	10	10	16	10	9
30	29	10	10	11 failed	16	10	9
36	35	11	11	11	16	11	10
37	36	12	12	12	16	12	10
44	43	12	12	13	20	13	11
47	46	13	13	14	20	13	12
51	50	14	14	16	20	14	12
58	57	15	15	18	20	15	13
59	58	15	15	16	20	15	13
69	68	16	16	19	22	16	14
80	79	17	18	19	24	17	15
91	90	19	19		28	19	16
102	101	20	20		30	20	17
113	112	21	21		30	21	18
124	123	23	23		30	22	19
139	138	24	24		32	23	20

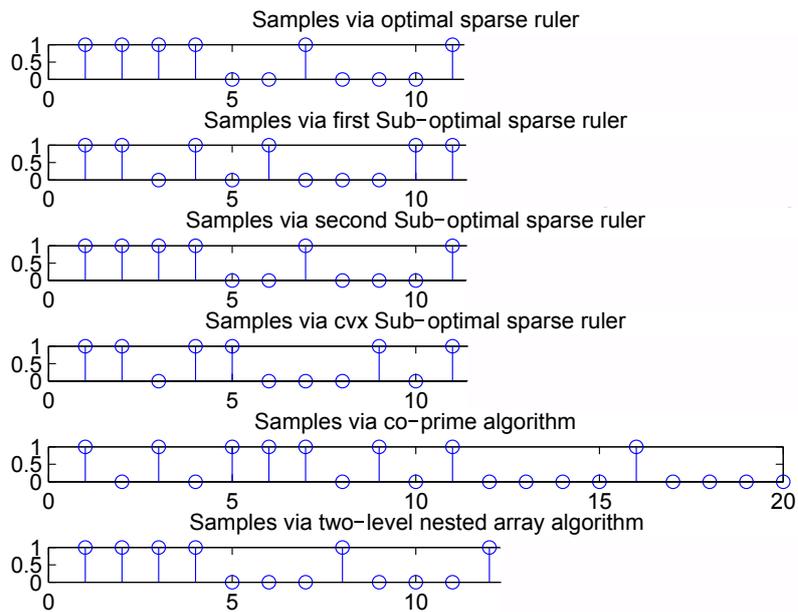


Figure 5.3: Comparison of sparse rulers for covering 10 correlation lags

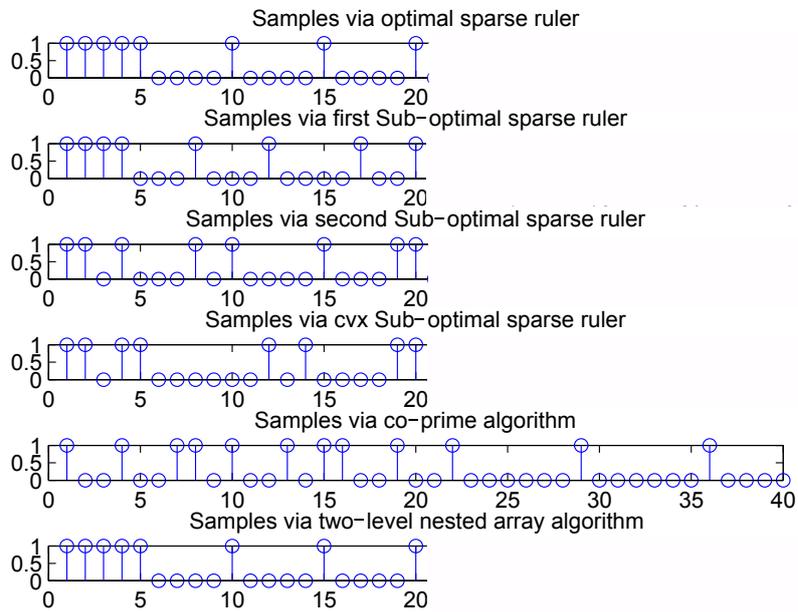


Figure 5.4: Comparison of sparse rulers for covering 19 correlation lags

the two-level nested array to obtain the samples, the performance is much better

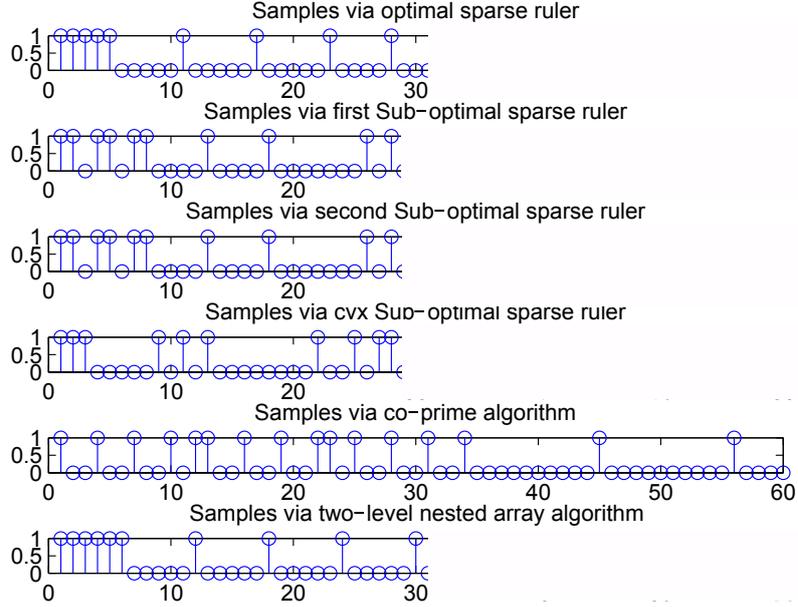


Figure 5.5: Comparison of sparse rulers for covering 27 correlation lags

than the one in coprime algorithm. We can select the samples from  $\{1, 2, \dots, 36\}$  and cover the lags  $\{-35, \dots, -1, 0, 1, \dots, 35\}$ , where the number of samples here is  $N = 11$ . It appears that this algorithm is quite good. Take some examples from Table 5.2. Here we present three cases for  $(N - 1)$ -length correlation lags, where  $N$  equal to 11, 20 and 28 respectively. From Fig 5.3 we can see, the indices of samples (the index starts from 1) for different algorithms are as follow. The indices for optimal sparse ruler is  $\{1, 2, 3, 4, 7, 11\}$ . The indices for the first algorithm we designed is  $\{1, 2, 4, 6, 10, 11\}$ . The indices for the second algorithm we designed is  $\{1, 2, 3, 4, 7, 11\}$ . The indices for the convex optimization based algorithm is  $\{1, 2, 4, 5, 9, 11\}$ . The indices for the coprime algorithm is  $\{1, 3, 5, 6, 7, 9, 11, 16\}$ . And the indices for the two-level nested array algorithm is  $\{1, 2, 3, 4, 8, 12\}$ . From Fig 5.4, the indices for optimal sparse ruler is  $\{1, 2, 3, 4, 5, 10, 15, 20\}$ . The indices for the first algorithm we designed is  $\{1, 2, 3, 4, 8, 12, 17, 20\}$ . The indices for the second algorithm we designed is  $\{1, 2, 4, 8, 10, 15, 19, 20\}$ . The indices for the convex optimization based algorithm is  $\{1, 2, 4, 5, 12, 14, 19, 20\}$ . The indices for the coprime algorithm is  $\{1, 4, 7, 8, 10, 13, 15, 16, 19, 22, 29, 36\}$ . And the indices for the two-level nested array algorithm is  $\{1, 2, 3, 4, 5, 10, 15, 20\}$ . From Fig 5.5, the indices for optimal minimal sparse ruler is  $\{1, 2, 3, 4, 5, 11, 17, 23, 28\}$ . The indices for the first algorithm we designed is  $\{1, 2, 4, 5, 7, 8, 13, 18, 26, 28\}$ . The indices for the second algorithm we designed is  $\{1, 2, 4, 5, 7, 8, 13, 18, 26, 28\}$ . The indices for the convex optimization based algorithm is  $\{1, 2, 3, 9, 11, 13, 22, 25, 27, 28\}$ . The indices for the coprime algorithm is  $\{1, 4, 7, 10, 12, 13, 16, 19, 22, 23, 25, 28, 31, 34, 45, 56\}$ . And the indices for the two-level nested array algorithm is  $\{1, 2, 3, 4, 5, 6, 12, 18, 24, 30\}$ . From

the figures we can find that, for coprime algorithm and two-level nested array algorithm, sometimes we have to collect samples out of the range  $N$  for  $(N - 1)$ -length lags. This does not occur in our algorithms. In our algorithms, we can decide the sampling range and the corresponding indices difference range we want perfectly. Moreover, the first half of samples in the two-level nested algorithm always need the same sampling rate as Nyquist rate. Now let have a look at convex based algorithm. It has a good performance for selecting sub-optimal sparse ruler, especially when  $N$  is small, however, it cost too much time. And when  $N$  increases, the performance of the convex based algorithm is not as good as the performances of the first algorithm, the second algorithm and the two-level nested array algorithm. Moreover, we notice that the convex optimization tool fail to obtain the optimal result since  $N = 29$ , so the results after  $N = 29$  may be not the optimal solution of the problem. That is also the reason why its performance becomes worse when  $N$  increases.

# Performance Analysis of Alternative Time Domain Approach in [1] and Cosets Selection

# 6

It is clear that when we use the minimum sparse ruler to construct sampling matrix  $\mathbf{C}$ , with compression  $\frac{M}{N}$ , we can reconstruct the power spectrum of the original signal using Least Square method (LS). However, we can improve the performance of the power spectrum estimation by adding  $X$  additional cosets to the selected  $M$  cosets. Now the compression becomes  $\frac{M+X}{N}$ . Let us now analyze how the cosets addition affect the performance of the estimated power spectrum. However, the analysis is still performed for the Gaussian white noise signal.

## 6.1 Theoretical Analysis

Since the signal is white noise, the estimated power spectrum is unbiased which we will explain later in the simulation study. Here we focus on the variance analysis. From equation (2.14), we can use LS method to construct  $\hat{\mathbf{r}}_{x,LS}$  from estimated  $\hat{\mathbf{r}}_y[0]$  in (2.16) as

$$\hat{\mathbf{r}}_{x,LS} = (\mathbf{R}_c^T \mathbf{R}_c)^{-1} \mathbf{R}_c^T \hat{\mathbf{r}}_y[0] = (\mathbf{R}_c^T \mathbf{R}_c)^{-1} \mathbf{R}_c^T (\mathbf{C} \otimes \mathbf{C}) \text{vec}(\hat{\mathbf{R}}_x), \quad (6.1)$$

where the element of  $\hat{\mathbf{R}}_x$  is given by  $[\hat{\mathbf{R}}_x]_{i+1,j+1} = \hat{\mathbf{r}}_x[i-j] = \frac{1}{K} \sum_{k=0}^{K-1} x[kN+i]x^*[kN+j]$ . We define the set  $\mathcal{M} = \{n_0, n_1, \dots, n_{m-1}\}$  selected from  $\{0, 1, \dots, N-1\}$  as the indices which are used to construct the sampling matrix  $\mathbf{C}$ . The indices  $\{1+n_i | \forall n_i \in \mathcal{M}\}$  are the indices of the identity matrix  $\mathbf{I}_N$  which is used to construct  $\mathbf{C}$ .

Since  $\mathbf{C}$  is constructed by selecting the rows of  $\mathbf{I}_N$ , and  $\mathbf{R}_c = (\mathbf{C} \otimes \mathbf{C})\mathbf{T}$  in (2.14), we can find that the rows of  $\mathbf{R}_c$  are given by  $(Nn_i + n_j + 1)$ -th and  $(Nn_j + n_i + 1)$ -th rows of  $\mathbf{T}$ , where  $n_i, n_j \in \mathcal{M}$ .  $\mathbf{T}$  is a special  $N^2 \times (2N-1)$  repetition matrix with the  $i$ -th row of  $\mathbf{T}$  given by the  $((i-1 + (N-2) \lfloor \frac{i-1}{N} \rfloor) \bmod (2N-1) + 1)$ -th row of the identity matrix  $\mathbf{I}_{2N-1}$ . As a result, we can find that the rows of  $\mathbf{R}_c$  is given by the  $((n_i - n_j) \bmod (2N-1) + 1)$ -th and  $((n_j - n_i) \bmod (2N-1) + 1)$ -th rows of identity matrix  $\mathbf{I}_{2N-1}$ , where  $n_i, n_j \in \mathcal{M}$ .

Observe that  $(\mathbf{R}_c^T \mathbf{R}_c)^{-1}$  is a  $(2N-1) \times (2N-1)$  diagonal matrix. The  $k$ -th diagonal element of  $(\mathbf{R}_c^T \mathbf{R}_c)^{-1}$  is  $\frac{1}{\gamma_k}$ , where  $\gamma_k$  is the number of times the  $k$ -th row of  $\mathbf{I}_{2N-1}$  appears in  $\mathbf{R}_c$ , which also indicates the number of times the corresponding  $k$ -th element of  $\mathbf{r}_x$  in (2.14) appears in  $\mathbf{R}_y[0]$  in (2.14). Since the total number of the elements of  $\mathbf{R}_y[0]$  is  $M^2$ , it is clear that  $\sum_{k=1}^{2N-1} \gamma_k = M^2$  and  $\gamma_1 = M$ .

Now let us have a look at  $\mathbf{R}_c^T (\mathbf{C} \otimes \mathbf{C})$  in (6.1), which can also be rewritten as  $\mathbf{T}^T (\mathbf{C} \otimes \mathbf{C})^T (\mathbf{C} \otimes \mathbf{C})$ . It is easy to find that  $(\mathbf{C} \otimes \mathbf{C})^T (\mathbf{C} \otimes \mathbf{C})$  is a  $N^2 \times N^2$  diagonal

matrix whose  $(Nn_i + n_j + 1)$ -th diagonal elements are equal to one for all  $n_i, n_j \in \mathcal{M}$  and zero elsewhere. We define  $\hat{\boldsymbol{\rho}}_x = (\mathbf{C} \otimes \mathbf{C})^T (\mathbf{C} \otimes \mathbf{C}) \text{vec}(\hat{\mathbf{R}}_x)$ , and the element of  $\hat{\boldsymbol{\rho}}_x$  can be written as

$$[\hat{\boldsymbol{\rho}}_x]_{Nn+n'+1} = \begin{cases} [\text{vec}(\hat{\mathbf{R}}_x)]_{Nn+n'+1} = [\hat{\mathbf{R}}_x]_{n'+1, n+1}, & \text{if } n, n' \in M \\ 0, & \text{otherwise} \end{cases} \quad (6.2)$$

Then, because  $\mathbf{T}$  is a special  $N^2 \times (2N - 1)$  repetition matrix with the  $i$ -th row of  $\mathbf{T}$  given by the  $((i - 1 + (N - 2) \lfloor \frac{i-1}{N} \rfloor) \bmod (2N - 1) + 1)$ -th row of the identity matrix  $\mathbf{I}_{2N-1}$ , we can find that for  $i = 1 - N, \dots, 0, \dots, N - 1$ , the  $((i \bmod (2N - 1)) + 1)$ -th row of  $\mathbf{T}^T$  contains ones only at the  $\{1 + i + (N + 1)n\}_{n=\lfloor \min(0, i) \rfloor}^{N-1+\min(0, -i)}$ -th elements. Then, we can write the element of  $\hat{\mathbf{r}}_{x,LS}$  in (6.1) as

$$\begin{aligned} \hat{r}_{x,LS}[i] &= [\hat{\mathbf{r}}_{x,LS}]_{((i \bmod (2N-1))+1)} \\ &= \frac{1}{\gamma_{((i \bmod (2N-1))+1)}} [\mathbf{T}^T \hat{\boldsymbol{\rho}}_x]_{((i \bmod (2N-1))+1)} \\ &= \frac{1}{\gamma_{((i \bmod (2N-1))+1)}} \sum_{n=\lfloor \min(0, i) \rfloor}^{N-1+\min(0, -i)} [\hat{\boldsymbol{\rho}}_x]_{Nn+n+i+1} \\ &= \frac{1}{\gamma_{((i \bmod (2N-1))+1)}} \sum_{n=\lfloor \min(0, i) \rfloor}^{N-1+\min(0, -i)} [\text{unvec}(\hat{\boldsymbol{\rho}}_x)]_{n+i+1, n+1} \end{aligned} \quad (6.3)$$

Because the signal is definitely a WSS signal, we can write  $[\text{vec}(\hat{\mathbf{R}}_x)]_{Nn+n'+1} = [\hat{\mathbf{R}}_x]_{n'+1, n+1} = \hat{r}_x[n - n'] = \hat{r}_x^*[n' - n]$ . As we know, the power spectrum can be written as

$$\hat{\mathbf{s}}_{x,LS} = [\hat{P}_{x,LS}(0), \hat{P}_{x,LS}(2\pi \frac{1}{2N-1}), \dots, \hat{P}_{x,LS}(2\pi \frac{2N-2}{2N-1})]^T = \mathbf{F}_{2N-1} \hat{\mathbf{r}}_{x,LS}, \quad (6.4)$$

where  $\mathbf{F}_{2N-1}$  is the  $(2N - 1) \times (2N - 1)$  DFT matrix. The covariance matrix of  $\hat{\mathbf{s}}_{x,LS}$  can be expressed as

$$\mathbf{C}_{\hat{\mathbf{s}}_{x,LS}} = E[\hat{\mathbf{s}}_{x,LS} \hat{\mathbf{s}}_{x,LS}^H] - E[\hat{\mathbf{s}}_{x,LS}] E[\hat{\mathbf{s}}_{x,LS}]^H = \mathbf{F}_{2N-1} \mathbf{C}_{\hat{\mathbf{r}}_{x,LS}} \mathbf{F}_{2N-1}^H, \quad (6.5)$$

where  $\mathbf{C}_{\hat{\mathbf{r}}_{x,LS}}$  is the covariance matrix of the  $\hat{\mathbf{r}}_{x,LS}$  which is given by

$$\mathbf{C}_{\hat{\mathbf{r}}_{x,LS}} = E[\hat{\mathbf{r}}_{x,LS} \hat{\mathbf{r}}_{x,LS}^H] - E[\hat{\mathbf{r}}_{x,LS}] E[\hat{\mathbf{r}}_{x,LS}]^H = (\mathbf{R}_c^T \mathbf{R}_c)^{-1} \mathbf{T}^T \mathbf{C}_{\hat{\boldsymbol{\rho}}_x} \mathbf{T} (\mathbf{R}_c^T \mathbf{R}_c)^{-1}, \quad (6.6)$$

where  $\mathbf{C}_{\hat{\boldsymbol{\rho}}_x}$  is the covariance matrix of the  $\hat{\boldsymbol{\rho}}_x$ .  $\mathbf{C}_{\hat{\boldsymbol{\rho}}_x}$  is given by

$$\mathbf{C}_{\hat{\boldsymbol{\rho}}_x} = E[\hat{\boldsymbol{\rho}}_x \hat{\boldsymbol{\rho}}_x^H] - E[\hat{\boldsymbol{\rho}}_x] E[\hat{\boldsymbol{\rho}}_x]^H = (\mathbf{C} \otimes \mathbf{C})^T (\mathbf{C} \otimes \mathbf{C}) \mathbf{C}_{\hat{\mathbf{R}}_x} (\mathbf{C} \otimes \mathbf{C})^T (\mathbf{C} \otimes \mathbf{C}), \quad (6.7)$$

where  $\mathbf{C}_{\hat{\mathbf{R}}_x}$  is the covariance matrix of the  $\text{vec}(\hat{\mathbf{R}}_x)$  expressed as

$$\mathbf{C}_{\hat{\mathbf{R}}_x} = E[\text{vec}(\hat{\mathbf{R}}_x) \text{vec}(\hat{\mathbf{R}}_x)^H] - E[\text{vec}(\hat{\mathbf{R}}_x)] E[\text{vec}(\hat{\mathbf{R}}_x)]^H. \quad (6.8)$$

We have obtained that  $[\text{vec}(\hat{\mathbf{R}}_x)]_{Nn'+n+1} = [\hat{\mathbf{R}}_x]_{n+1, n'+1} = \hat{r}_x[n-n'] = \hat{r}_x^*[n'-n]$ , so the element of  $\mathbf{C}_{\hat{R}_x}$  at the  $Nn'+n+1$ -th row and the  $Nv'+v+1$ -th column is given by

$$\text{Cov}([\text{vec}(\hat{\mathbf{R}}_x)]_{Nn'+n+1}, [\text{vec}(\hat{\mathbf{R}}_x)]_{Nv'+v+1}) = \text{Cov}(\hat{r}_x[n-n'], \hat{r}_x[v-v']), \quad (6.9)$$

where  $n, n', v, v' \in \{0, 1, 2, \dots, N-1\}$ . From [1] we obtain for white Gaussian noise  $x[n]$

$$\text{Cov}(\hat{\mathbf{r}}_{y_i y_j}[0], \hat{\mathbf{r}}_{y_w y_v}[0]) = \frac{\sigma^4 \delta[n_i - n_w] \delta[n_j - n_v]}{K} = \frac{\sigma^4 \delta[i - w] \delta[j - v]}{K}, \quad (6.10)$$

where  $\hat{\mathbf{r}}_{y_i y_j}[0] = \hat{r}_x[n_i - n_j]$ . Inserting (6.10) into (6.9) we can obtain that

$$\text{Cov}(\hat{r}_x[n-n'], \hat{r}_x[v-v']) = \frac{\sigma^4 \delta[n-v] \delta[n'-v']}{K} \quad (6.11)$$

It is clear that  $\mathbf{C}_{\hat{R}_x}$  is a diagonal matrix with all diagonal elements equal to  $\frac{\sigma^4}{K}$ . Because  $(\mathbf{C} \otimes \mathbf{C})^T (\mathbf{C} \otimes \mathbf{C})$  is also diagonal matrix, we can see that  $\mathbf{C}_{\hat{\rho}_x}$  in (6.7) is also a diagonal matrix. By considering (6.2), (6.7), (6.9) and (6.11), we can find that the  $(Nn_i+n_j+1)$ -th diagonal elements of the  $\mathbf{C}_{\hat{\rho}_x}$  are equal to  $\frac{\sigma^4}{K}$ , whenever  $n_i, n_j \in \mathcal{M}$ . Once we obtain  $\mathbf{C}_{\hat{\rho}_x}$ , we can use it in (6.6) by taking (6.3) into account. The elements of  $\mathbf{C}_{\hat{r}_{x,LS}}$  can be expressed as

$$\begin{aligned} & [\mathbf{C}_{\hat{r}_{x,LS}}]_{(i \bmod (2N-1))+1, (j \bmod (2N-1))+1} = E[\hat{r}_{x,LS}[i] \hat{r}_{x,LS}[j]^*] - E[\hat{r}_{x,LS}[i]] E[\hat{r}_{x,LS}[j]^*] \\ & = \frac{1}{\gamma(i \bmod (2N-1))+1 \gamma(j \bmod (2N-1))+1} \sum_{n=|\min(0,i)|}^{N-1+\min(0,-i)} \sum_{n'=|\min(0,j)|}^{N-1+\min(0,-j)} E[[\hat{\rho}_x]_{Nn+n+i+1} [\hat{\rho}_x^*]_{Nn'+n'+j+1}] \\ & - \frac{1}{\gamma(i \bmod (2N-1))+1 \gamma(j \bmod (2N-1))+1} \sum_{n=|\min(0,i)|}^{N-1+\min(0,-i)} \sum_{n'=|\min(0,j)|}^{N-1+\min(0,-j)} E[[\hat{\rho}_x]_{Nn+n+i+1}] E[[\hat{\rho}_x]_{Nn'+n'+j+1}]^* \\ & = \frac{1}{\gamma(i \bmod (2N-1))+1 \gamma(j \bmod (2N-1))+1} \sum_{n=|\min(0,i)|}^{N-1+\min(0,-i)} \sum_{n'=|\min(0,j)|}^{N-1+\min(0,-j)} \text{Cov}([\hat{\rho}_x]_{Nn+n+i+1} [\hat{\rho}_x]_{Nn'+n'+j+1}) \\ & = \frac{1}{\gamma(i \bmod (2N-1))+1 \gamma(j \bmod (2N-1))+1} \sum_{n=|\min(0,i)|}^{N-1+\min(0,-i)} \sum_{n'=|\min(0,j)|}^{N-1+\min(0,-j)} \text{Cov}([\hat{\rho}_x]_{Nn+n+i+1} [\hat{\rho}_x]_{Nn'+n'+j+1}) \\ & \delta[N(n-n') + n + i - n' - j] \\ & = \frac{1}{\gamma(i \bmod (2N-1))+1 \gamma(j \bmod (2N-1))+1} \sum_{n=|\min(0,i)|}^{N-1+\min(0,-i)} \sum_{n'=|\min(0,j)|}^{N-1+\min(0,-j)} \text{Cov}([\hat{\rho}_x]_{Nn+n+i+1} [\hat{\rho}_x]_{Nn'+n'+j+1}) \end{aligned} \quad (6.12)$$

$$\begin{aligned} & \delta[n-n'] \delta[i-j] \\ & = \frac{1}{\gamma^2(i \bmod (2N-1))+1} \sum_{n=|\min(0,i)|}^{N-1+\min(0,-i)} \text{Cov}([\hat{\rho}_x]_{Nn+n+i+1} [\hat{\rho}_x]_{Nn+n+j+1}) \delta[i-j], \end{aligned} \quad (6.13)$$

where  $i, j = -N + 1, \dots, -1, 0, 1, \dots, N - 1$  and  $\delta[n]$  is Dirac delta function. Because matrix  $\mathbf{C}_{\rho_x}$  is a diagonal matrix, the matrix  $\mathbf{C}_{\hat{\rho}_{x,LS}}$  is also a diagonal matrix. By using the (6.13), we can calculate the element of  $\mathbf{C}_{\hat{\rho}_{x,LS}}$  as

$$\begin{aligned} & \text{Cov}(\hat{P}_{x,LS}(2\pi \frac{n}{2N-1}), \hat{P}_{x,LS}(2\pi \frac{n'}{2N-1})) \\ &= \sum_{i=1-N}^{N-1} \sum_{j=1-N}^{N-1} [\mathbf{F}_{2N-1}]_{n+1, (i \bmod (2N-1))+1} [\mathbf{C}_{\hat{\rho}_{x,LS}}]_{(i \bmod (2N-1))+1, (j \bmod (2N-1))+1} \end{aligned} \quad (6.14)$$

$$\begin{aligned} & [\mathbf{F}_{2N-1}^H]_{(j \bmod (2N-1))+1, n'+1} \\ &= \sum_{i=1-N}^{N-1} \sum_{j=1-N}^{N-1} e^{-\frac{j2\pi n(i \bmod (2N-1))}{2N-1}} [\mathbf{C}_{\hat{\rho}_{x,LS}}]_{(i \bmod (2N-1))+1, (j \bmod (2N-1))+1} e^{\frac{j2\pi n'(j \bmod (2N-1))}{2N-1}} \\ &= \sum_{i=1-N}^{N-1} \sum_{j=1-N}^{N-1} e^{-\frac{j2\pi ni}{2N-1}} e^{\frac{j2\pi n'j}{2N-1}} [\mathbf{C}_{\hat{\rho}_{x,LS}}]_{(i \bmod (2N-1))+1, (j \bmod (2N-1))+1} \\ &= \sum_{i=1-N}^{N-1} \sum_{j=1-N}^{N-1} e^{-\frac{j2\pi ni}{2N-1}} e^{\frac{j2\pi n'j}{2N-1}} \frac{1}{\gamma^2_{(i \bmod (2N-1))+1}} \sum_{g=|\min(0,i)|}^{N-1+\min(0,-i)} \text{Cov}([\hat{\rho}_x]_{Ng+g+i+1} [\hat{\rho}_x]_{Ng+g+j+1}) \end{aligned} \quad (6.15)$$

$$\begin{aligned} & \delta[i - j] \\ &= \sum_{i=1-N}^{N-1} \frac{e^{\frac{j2\pi i(n'-n)}{2N-1}}}{\gamma^2_{(i \bmod (2N-1))+1}} \sum_{g=|\min(0,i)|}^{N-1+\min(0,-i)} \text{Cov}([\hat{\rho}_x]_{Ng+g+i+1} [\hat{\rho}_x]_{Ng+g+i+1}) \\ &= \sum_{i=1-N}^{N-1} \frac{e^{\frac{j2\pi i(n'-n)}{2N-1}}}{\gamma^2_{(i \bmod (2N-1))+1}} \sum_{g=|\min(0,i)|}^{N-1+\min(0,-i)} \text{Var}([\hat{\rho}_x]_{Ng+g+i+1}) \end{aligned} \quad (6.16)$$

Since we have derived the  $(Nn_i + n_j + 1)$ -th diagonal elements of the  $\mathbf{C}_{\hat{\rho}_x}$  for all  $n_i$  and  $n_j$ , which are equal to  $\frac{\sigma^4}{K}$ , whenever  $n_i, n_j \in M$ , we can obtain

$$\text{Var}([\hat{\rho}_x]_{Ng+g+i+1}) = \begin{cases} \frac{\sigma^4}{K}, & \text{if } g, g+i \in M \\ 0, & \text{otherwise} \end{cases}. \quad (6.17)$$

Inserting (6.17) into (6.16), we can obtain

$$\text{Cov}(\hat{P}_{x,LS}(2\pi \frac{n}{2N-1}), \hat{P}_{x,LS}(2\pi \frac{n'}{2N-1})) \quad (6.18)$$

$$\begin{aligned} &= \sum_{i=1-N}^{N-1} \frac{e^{\frac{j2\pi i(n'-n)}{2N-1}}}{\gamma^2_{(i \bmod (2N-1))+1}} \sum_{g=|\min(0,i)|}^{N-1+\min(0,-i)} \text{Var}([\hat{\rho}_x]_{Ng+g+i+1}) \\ &= \sum_{i=1-N}^{N-1} \frac{e^{\frac{j2\pi i(n'-n)}{2N-1}}}{\gamma^2_{(i \bmod (2N-1))+1}} \gamma_{(i \bmod (2N-1))+1} \frac{\sigma^4}{K}. \end{aligned} \quad (6.19)$$

From (6.19), we can obtain the variance of the estimated power spectrum by setting  $n = n'$ . Then we obtain the variance of the estimated power spectrum as follows,

$$\text{Var}(\hat{P}_{x,LS}(2\pi\frac{n}{2N-1})) = \frac{\sigma^4}{K} \sum_{l=0}^{2N-2} \frac{1}{\gamma_{l+1}}. \quad (6.20)$$

Then the problem become how to allocate  $\gamma_k$  s.t.  $\sum_{k=1}^{2N-1} \gamma_k = M^2$  and  $\gamma_1 = M$ .

## 6.2 Simulation study

After obtaining the theoretical variance of the power spectrum for white Gaussian noise case in (6.20), we can calculate the theoretical normalized mean square error (NMSE) of the estimated power spectrum for white Gaussian noise signal. It is well known that the NMSE of the estimated power spectrum  $\hat{\mathbf{s}}_x$  is given by

$$\text{NMSE}(\hat{\mathbf{s}}_x) = \frac{E(\|\hat{\mathbf{s}}_x - \mathbf{s}_x\|_2^2)}{\|\mathbf{s}_x\|_2^2} = \frac{\text{tr}(\mathbf{C}_{\hat{\mathbf{s}}_x}) + \|E(\hat{\mathbf{s}}_x) - \mathbf{s}_x\|_2^2}{\|\mathbf{s}_x\|_2^2} \quad (6.21)$$

where  $\text{tr}(\cdot)$  is the trace operator. Since the signal we used here is the white Gaussian noise, the power spectrum of the signal at every frequency point is equal to the variance of the Gaussian noise. Combining (6.1) and (6.4) we can find that  $\hat{\mathbf{s}}_x$  is a linear function of  $\hat{\mathbf{r}}_y[0]$ . Since all the elements in  $\hat{\mathbf{r}}_y[0]$  are computed by an unbiased estimation using (2.15),  $\hat{\mathbf{s}}_x$  is an unbiased estimate of  $\mathbf{s}_x$ , which indicates  $\|E(\hat{\mathbf{s}}_x) - \mathbf{s}_x\|_2^2 = 0$ . Then, the NMSE of  $\hat{\mathbf{s}}_x$  becomes

$$\text{NMSE}(\hat{\mathbf{s}}_x) = \frac{\text{tr}(\mathbf{C}_{\hat{\mathbf{s}}_x})}{\|\mathbf{s}_x\|_2^2}. \quad (6.22)$$

Then, we can obtain that the theoretical NMSE of the estimated power spectrum reconstructed using LS method and for white Gaussian noise is given by

$$\text{NMSE}(\hat{\mathbf{s}}_{x,LS}) = \frac{\sigma^4}{K} \sum_{l=0}^{2N-2} \frac{1}{\gamma_{l+1}}. \quad (6.23)$$

In the simulation, we focus on the  $M + M_{add}$  samples selected from  $N$  consecutive Nyquist-rate samples, where  $M$  and  $M_{add}$  present the number of samples based on length- $(N - 1)$  minimal sparse ruler and the number of additional samples respectively. We here set  $N$  and  $M$  to  $N = 20$  and  $M = 8$ . The indices set of the  $M$  samples is given by  $S = \{0, 1, 2, 3, 4, 9, 14, 19\}$ . The indices set of the additional samples is defined as  $S_{add}$ . We compare NMSEs of the estimated power spectrum in four cases. In the first case, we have  $S_{add} = \{11, 12, 13\}$ . In the second case, we have  $S_{add} = \{5, 7, 8\}$ . In the third case, we have  $S_{add} = \{10, 15, 18\}$ . In the fourth case, we have  $S_{add} = \emptyset$ . From the set of the indices of the collected samples, we can obtain a description about the distribution of the appearance of the indices differences, which are shown in Fig. 6.1, Fig. 6.2, Fig. 6.3, and Fig. 6.4.

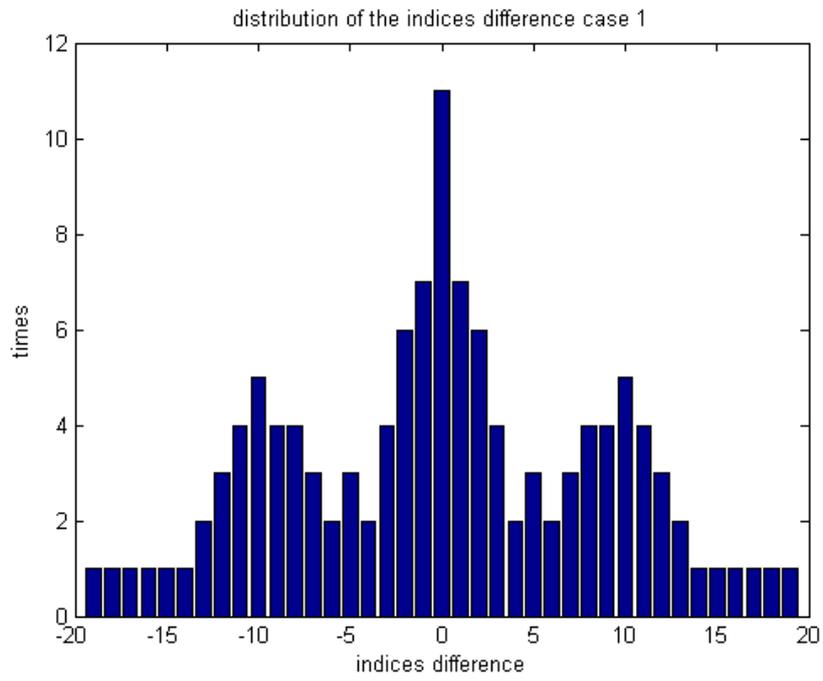


Figure 6.1: Distribution of indices differences in Case 1

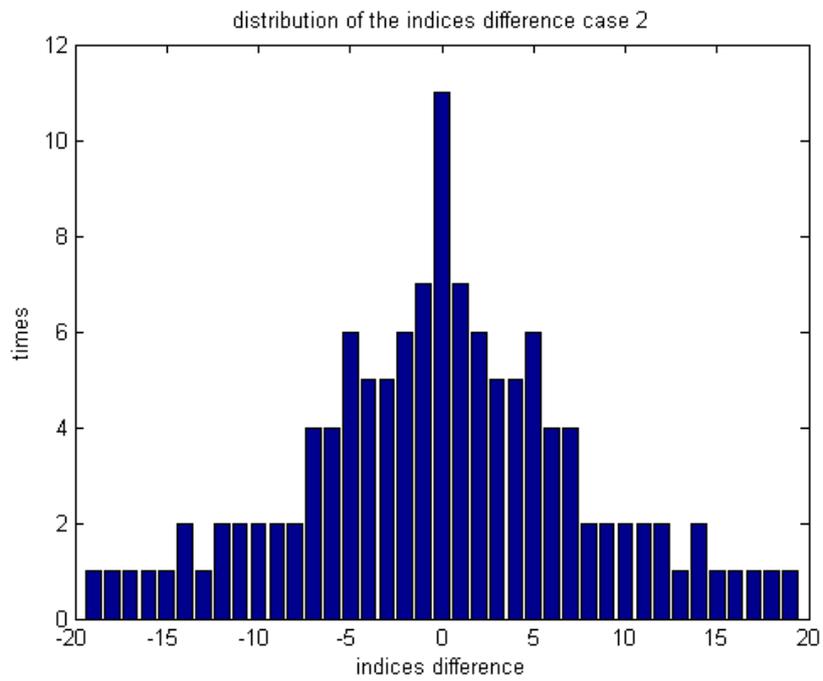


Figure 6.2: Distribution of indices differences in Case 2

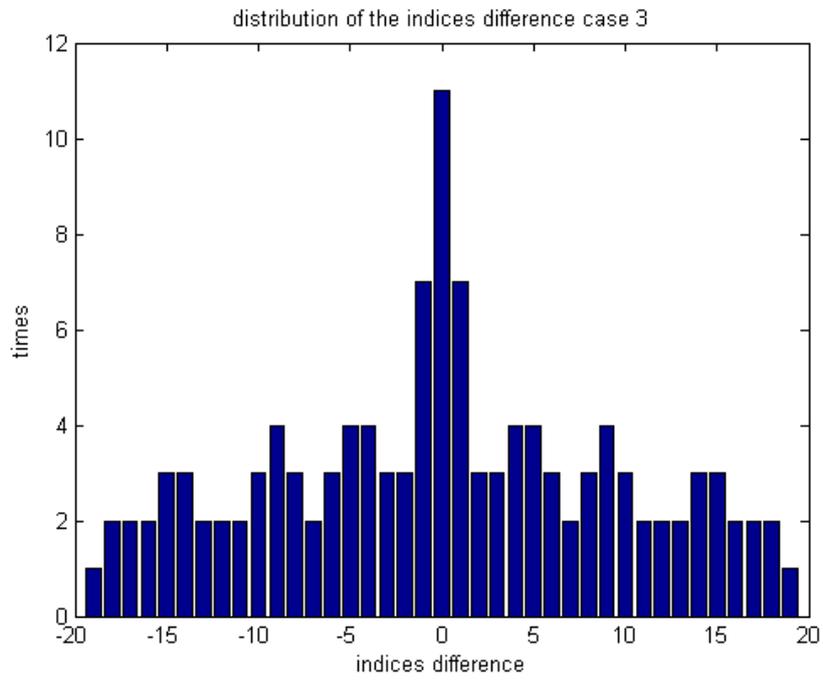


Figure 6.3: Distribution of indices differences in Case 3

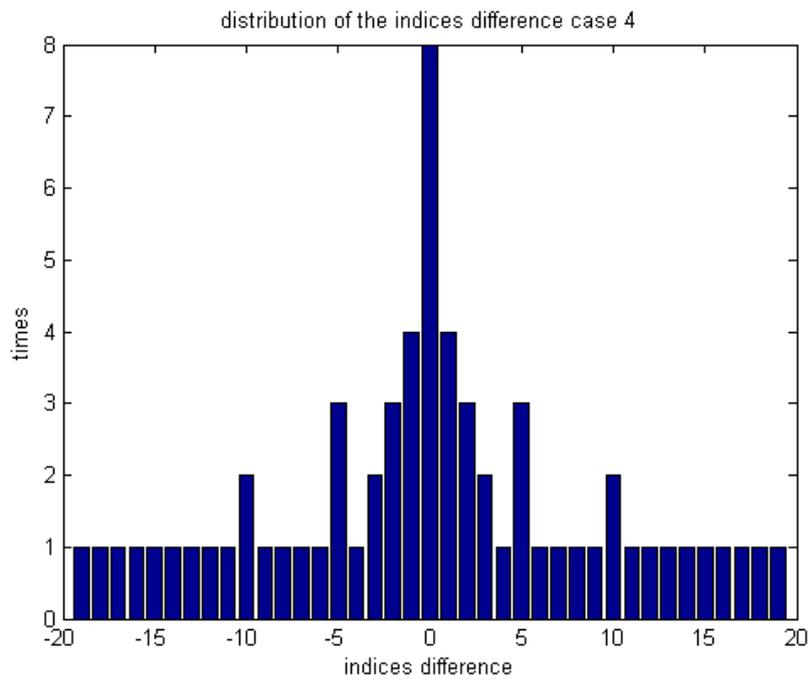


Figure 6.4: Distribution of indices differences in Case 4

By using the aforementioned result of the distribution of the indices differences of the collected samples, we compute the estimated NMSE and the theoretical NMSE for each case, and see how they change with the increased measurements  $K$ . In order to calculate the estimated NMSE, we perform 3000 Monte Carlo simulation runs. The results are shown in Fig. 6.5 and the results in dB scale are shown in Fig. 6.6.

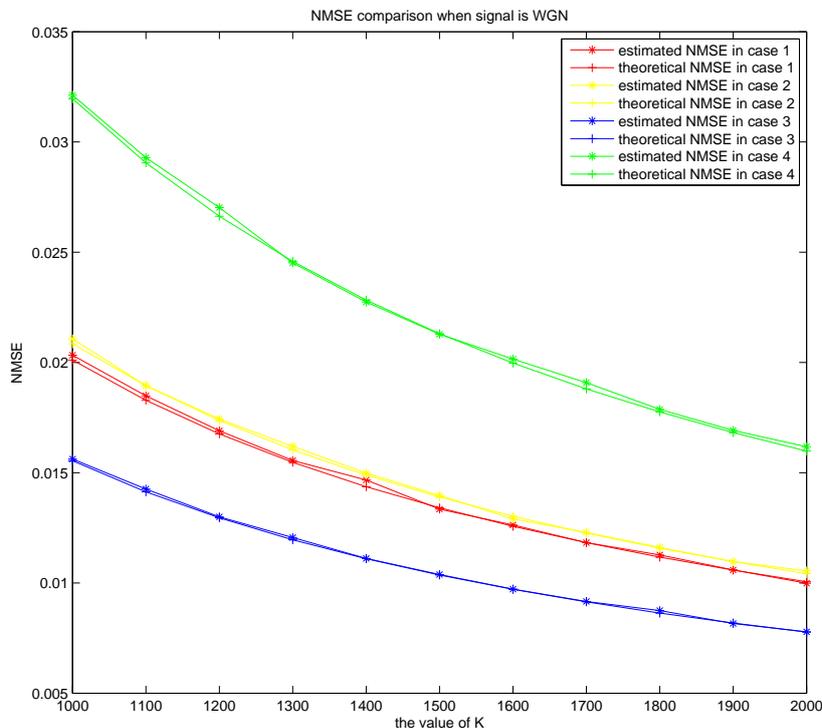


Figure 6.5: Theoretical and simulated NMSE for all the four cases

From the simulation results, we can find that the theoretical NMSE is almost on the top of the NMSE simulated in the Monte Carlo runs. It is clear that with the increasing of the number of measurements, the NMSE will decrease. Moreover, we can draw a conclusion from the results that the NMSE will achieve a small value if the distribution of the indices differences of the collected samples is as equal as possible.

Note that the aforementioned results are for white Gaussian noise signals. Now, we also examine the simulated NMSE computation for a more general Gaussian signal and find some interesting results. First of all, we design four filters to generate the received signal by passing the white Gaussian noise through the filter. The filters are low-pass filter, high-pass filter, bandpass filter and the multi-bandpass filter. We also plot the power spectrum of the filtered white Gaussian noise. We compute the theoretical power spectrum by multiplying the square of the magnitude of the frequency response of the filter with the variance of the noise we used. Then we also perform the compressive power spectrum estimation and label the results as the estimated power spectrum. The

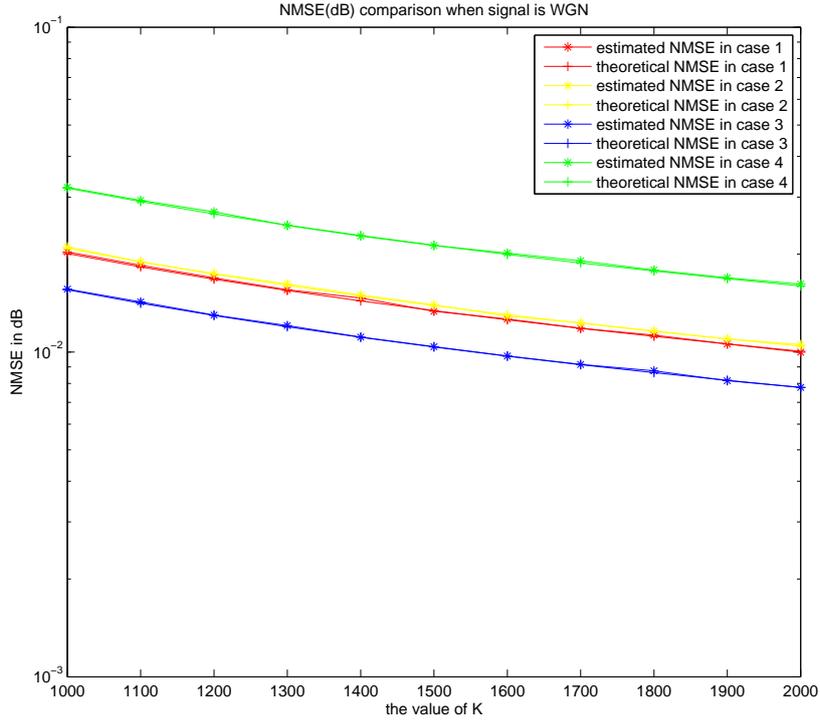


Figure 6.6: Theoretical and simulated NMSE for all the four cases in dB

results for the four different filters are shown in Fig. 6.7, Fig. 6.8, Fig. 6.9 and Fig. 6.10.

The received signal is generated by passing the white noise with a filter using (6.21), where the  $\hat{\mathbf{s}}_x$  is estimated by the compressive power spectrum estimation and the theoretical  $\mathbf{s}_x$  is computed by multiplying the square of the magnitude of the frequency response of the filter with the variance of the noise. After obtaining the filters, we calculate the NMSE of estimated power spectrum with different filters. The results are shown in Fig. 6.11, Fig. 6.13, Fig. 6.15, Fig. 6.17 and the results in dB scale are shown in Fig. 6.12, Fig. 6.14, Fig. 6.16 and Fig. 6.18 for the four types of filters respectively.

From the above results, we can find that even if the received signal is not white Gaussian noise, the NMSE of the estimated power spectrum is also reduced, if the distribution of the indices differences of the collected sample is as equal as possible. Besides, although for such general Gaussian signal, we have no idea about the theoretical NMSE, we also observe that the NMSE of the estimated power spectrum is reduced when the number of measurements  $K$  is increased.

### 6.3 Minimal Compression Rate with certain NMSE

We have known the NMSE of the estimated power spectrum  $\hat{\mathbf{s}}_x$  in (6.23). Then in this section we will investigate an extended problem where we fix the NMSE of the



Figure 6.7: Power spectrum of the WGN passed through low-pass filter

estimated power spectrum  $\hat{\mathbf{s}}_x$ . For this fixed NMSE, we then investigate the minimal compression rate we can achieve. We define the  $N$ -dimensions vector  $\mathbf{v}$  as a selection vector whose elements are 1 and 0. This selection vector illustrates which of  $N$  Nyquist-rate samples in the original vector of samples  $\mathbf{x}$  that have been collected. For example,  $\mathbf{v} = [1, 0, 0, 1, 0]^T$  means that out of 5 Nyquist-rate samples, only the first and the fourth samples have been collected while the others are not collected by the sampling matrix  $\mathbf{C}$ . In this case, matrix  $\mathbf{C}$  is given as

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}. \quad (6.24)$$

Here we define the matrix  $\mathbf{V} = \mathbf{v}\mathbf{v}^T$ . As we know, the  $\text{NMSE}(\hat{\mathbf{s}}_{x,LS}) = \frac{\sigma^4}{K} \sum_{l=0}^{2N-2} \frac{1}{\gamma_{l+1}}$  is given by (6.23) for white Gaussian noise signal, where  $\gamma_{l+1}$  and  $K$  present the number of times the  $l + 1$ -th row of  $\mathbf{I}_{2N-1}$  appears in  $\mathbf{R}_c$  and the number of measurements respectively. It is easy to find that  $\mathbf{C}^T\mathbf{C}$  is a diagonal matrix where its diagonal entries are the same as the elements of  $\mathbf{v}$ , i.e.,  $\mathbf{C}^T\mathbf{C} = \text{diag}[\mathbf{v}]$ . As we have discussed, the  $k$ -th diagonal element of  $(\mathbf{R}_c^T\mathbf{R}_c)$  is  $\gamma_k$ . Then using (2.14), we can rewrite  $(\mathbf{R}_c^T\mathbf{R}_c)$  as

$$(\mathbf{R}_c^T\mathbf{R}_c) = \mathbf{T}^T(\mathbf{C}^T\mathbf{C} \otimes \mathbf{C}^T\mathbf{C})\mathbf{T}. \quad (6.25)$$

Replacing  $\mathbf{C}^T\mathbf{C}$  with  $\text{diag}[\mathbf{v}]$  results in

$$(\mathbf{R}_c^T\mathbf{R}_c) = \mathbf{T}^T(\text{diag}[\mathbf{v}] \otimes \text{diag}[\mathbf{v}])\mathbf{T}. \quad (6.26)$$

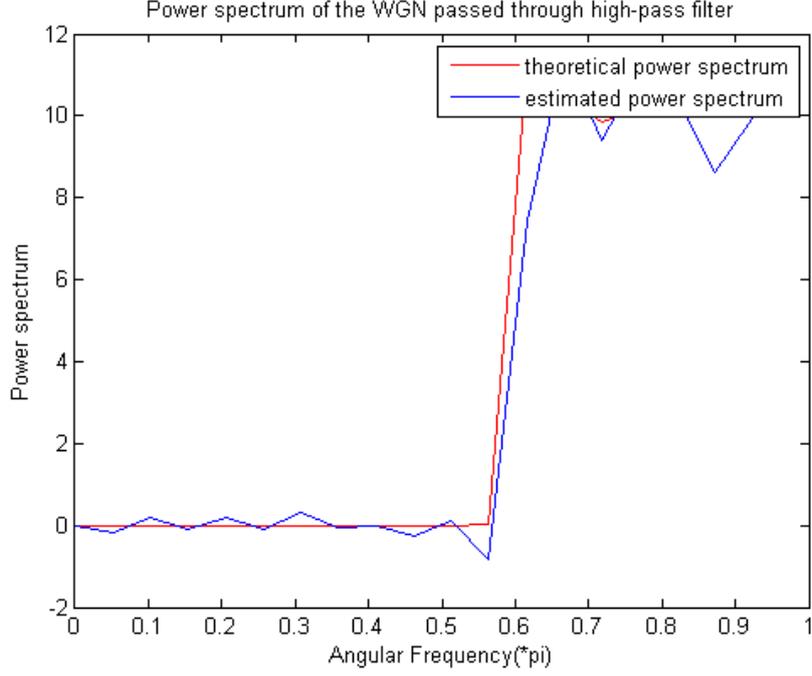


Figure 6.8: Power spectrum of the WGN passed through high-pass filter

Now let's have a look at  $\text{diag}[\mathbf{v}] \otimes \text{diag}[\mathbf{v}]$  in (6.26). It is easy to find that  $\text{diag}[\mathbf{v}] \otimes \text{diag}[\mathbf{v}]$  is an  $N^2 \times N^2$  diagonal matrix, whose  $(n_i - 1)N + n_j$ -th diagonal element is equal to  $v[n_i]v[n_j]$ , where  $n_i, n_j \in \{1, 2, \dots, N\}$ . As we have discussed before, for  $i = 1 - N, \dots, 0, \dots, N - 1$ , the  $((i \bmod (2N - 1)) + 1)$ -th row of  $\mathbf{T}^T$  contains ones only at the  $\{1 + i + (N + 1)n\}_{n=|\min(0,i)|}^{N-1+\min(0,-i)}$ -th elements. Let's have a look at  $1 + i + (N + 1)n$ . It is easy to find that

$$1 + i + (N + 1)n = N((n + 1) - 1) + (n + 1 + i). \quad (6.27)$$

So we can rewrite the  $((i \bmod (2N - 1)) + 1)$ -th diagonal element in  $(\mathbf{R}_c^T \mathbf{R}_c)$  as

$$\gamma_{(i \bmod (2N-1))+1} = \sum_{n=|\min(0,i)|}^{N-1+\min(0,-i)} v[n + 1]v[n + 1 + i]. \quad (6.28)$$

Then, we can rewrite the NMSE of the estimated power spectrum as

$$\frac{\sigma^4}{K} \sum_{l=0}^{2N-2} \frac{1}{\gamma_{l+1}} = \frac{\sigma^4}{K} \sum_{i=1-N}^{N-1} \frac{1}{\sum_{n=|\min(0,i)|}^{N-1+\min(0,-i)} v[n + 1]v[n + 1 + i]}. \quad (6.29)$$

After obtaining the relation between  $\mathbf{v}$  and  $\text{NMSE}(\hat{\mathbf{s}}_{x,LS})$ , we can write the conditioned optimization problem as

$$\min_{\mathbf{v}} \|\mathbf{v}\|_1 \text{ s.t. } \frac{\sigma^4}{K} \sum_{i=1-N}^{N-1} \frac{1}{\sum_{n=|\min(0,i)|}^{N-1+\min(0,-i)} v[n + 1]v[n + 1 + i]} \leq \xi. \quad (6.30)$$

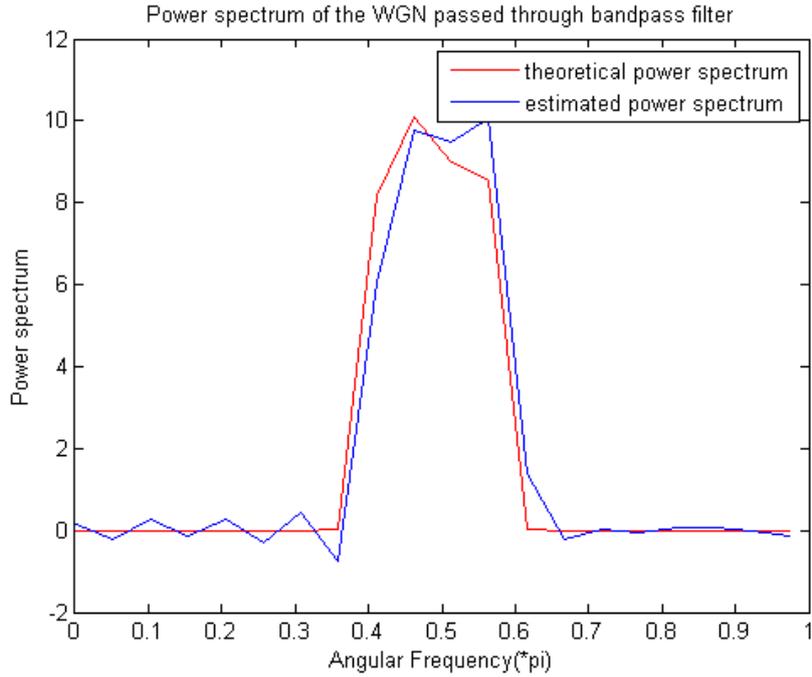


Figure 6.9: Power spectrum of the WGN passed through bandpass filter

However, it does not satisfy the requirements to be considered as a convex problem. Recall that a convex optimization problem is defined as follows [4]. A mathematical optimization problem, or just optimization problem, has the form

$$\begin{aligned} & \text{minimize} && f_0(x) \\ & \text{subject to} && f_i(x) \leq b_i, \quad i = 1, \dots, m. \end{aligned} \quad (6.31)$$

Here the vector  $\mathbf{x} = [x[1], x[2], \dots, x[n]]^T$  is the optimization variable of the problem, the function  $f_0$  is the objective function and the functions  $f_i$  are the constraint functions, and the constants  $b_1, \dots, b_m$  are the bounds, for the constraints. The convex optimization problem is one in which the objective and constraint functions are convex, which means they satisfy the inequality

$$f_i(\alpha x + \beta y) \leq \alpha f_i(x) + \beta f_i(y), \quad (6.32)$$

for all  $x, y \in \mathbf{R}^n$  and all  $\alpha, \beta \in \mathbf{R}$  with  $\alpha + \beta = 1, \alpha \geq 0, \beta \geq 0$ . In our case, the objective function does not satisfy the requirements to be a convex function. In order to solve the problem with convex optimization algorithm, we have to do some relaxation on the objective function.

First, we define the bounds of the vector  $\mathbf{v}$  as  $\mathbf{1}_{N \times 1} \geq \mathbf{v} \geq \mathbf{0}_{N \times 1}$ , where  $\mathbf{1}_{N \times 1}$  is an  $N \times 1$  vector containing ones in all entries and  $\mathbf{0}_{N \times 1}$  is an  $N \times 1$  vector containing zeros in all entries. Now the value of the elements of  $\mathbf{v}$  turned to be real number, not integer  $\{0, 1\}$  anymore. With this relaxation, we can prove that the condition of the

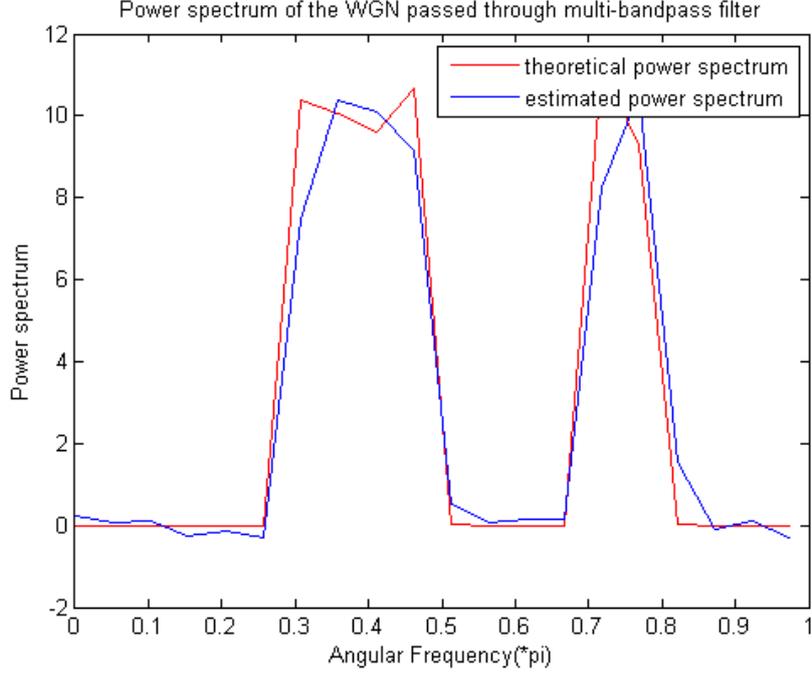


Figure 6.10: Power spectrum of the WGN passed through multi-bandpass filter

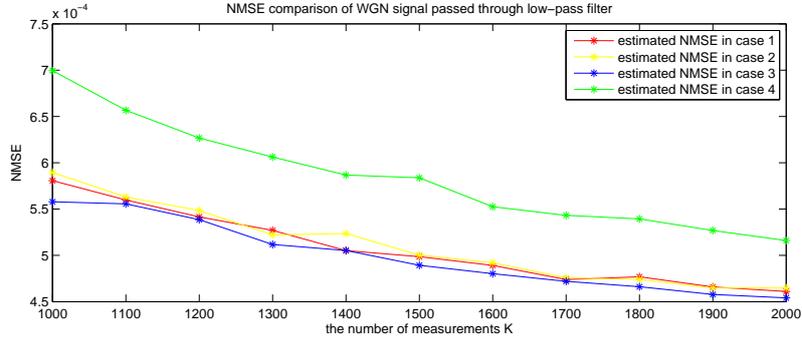


Figure 6.11: NMSE comparison of WGN signal passed through low-pass filter

optimization meets the requirements of convex definition (see appendix). Second, we relax the condition  $\mathbf{V} = \mathbf{v}\mathbf{v}^T$  as in (5.7). Then, we obtain

$$\begin{pmatrix} \mathbf{V} & \mathbf{v} \\ \mathbf{v}^T & 1 \end{pmatrix} \succeq 0. \quad (6.33)$$

After relaxation,  $\mathbf{V}$  can be different from  $\mathbf{v}\mathbf{v}^T$ , so we need to add some condition in addition such as  $\text{diag}(\mathbf{V}, 0) = \mathbf{v}$  and for each  $i = 1 - N, \dots, N - 1$ ,  $\text{diag}(\mathbf{V}, i) \geq 1$  ensuring all autocorrelation lags from  $1 - N$  to  $N - 1$  have been covered.

Note that our goal is to minimize the compression rate which is equivalent to minimizing  $\|\mathbf{v}\|_0$ . However, we do relaxation and minimizing  $\|\mathbf{v}\|_1$ . In fact, we try to

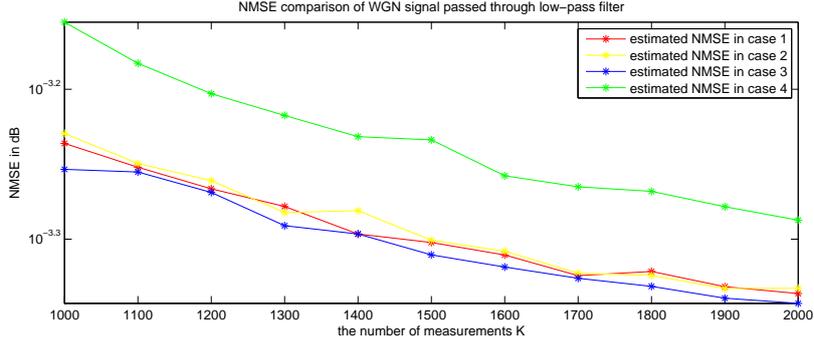


Figure 6.12: NMSE comparison of WGN signal passed through low-pass filter (in dB)

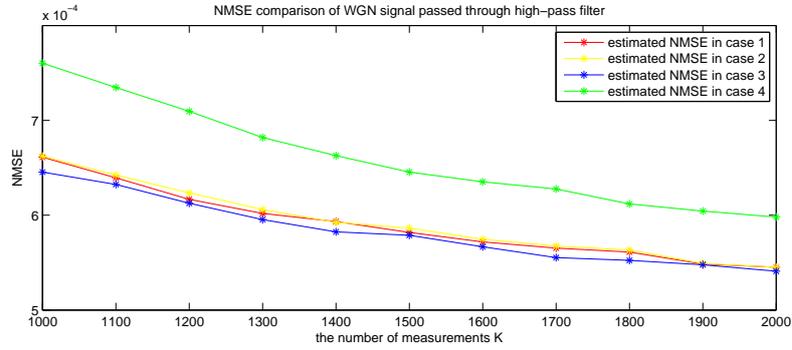


Figure 6.13: NMSE comparison of WGN signal passed through high-pass filter

follow [5] and minimize  $\mathbf{w}^T \mathbf{v}$  where  $\mathbf{w}$  is a  $N \times 1$  weight vector.

The iteration algorithm to find the proper  $\mathbf{v}$  for the sampling matrix is introduced as follows, where  $\mathbf{w} = [w[0], w[1], \dots, w[N-1]]^T$ .

1. Set the iteration count  $l$  to zero and initialize all the elements in weight vector  $\mathbf{w}^{(l)}$ , with  $l = 0$ , to 1.
2. Solve the weighted minimization problem: We want to minimize  $\mathbf{w}^{(l)T} \mathbf{v}$  over the vector  $\mathbf{v}$ , with the constraint that the index differences set  $\Omega_{S_m}$  covers all integers from  $1 - N$  up to  $N - 1$ .
3. Update the weights  $w[i]$  for each  $i = 0, 1, 2, \dots, N - 1$

$$w^{(l+1)}[i] = \frac{1}{|v^{(l)}[i]| + \epsilon} \quad (6.34)$$

where  $\epsilon$  is a small value in order to provide stability and ensure that a zero-valued component in  $\mathbf{v}^{(l)}$  does not strictly prohibit a nonzero estimate at the next step.

4. Terminate on convergence or when  $(l)$  attains a specified maximum number of iterations  $l_{max}$ . Otherwise, increment  $(l)$  and go to step 2.

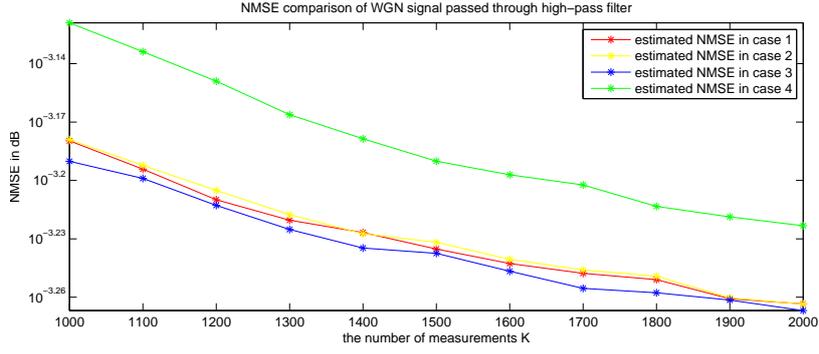


Figure 6.14: NMSE comparison of WGN signal passed through high-pass filter (in dB)

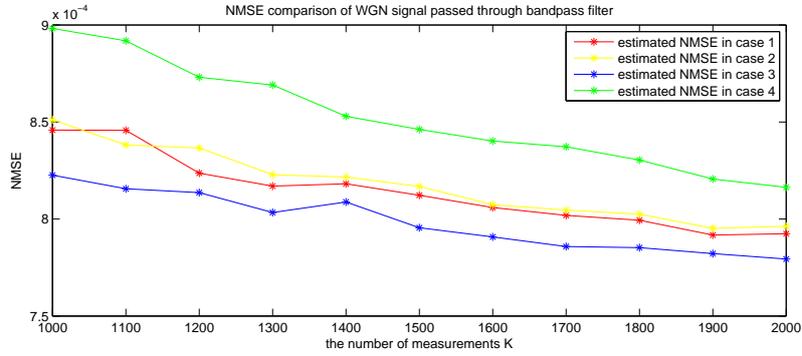


Figure 6.15: NMSE comparison of WGN signal passed through bandpass filter

Now we can write the problem as an optimization problem

$$\begin{aligned}
& \min_{\mathbf{v}} \mathbf{w}^T \mathbf{v} \\
& \text{s.t.} \quad \frac{\sigma^2}{K} \sum_{i=1-N}^{N-1} \frac{1}{\text{sum}(\text{diag}(\mathbf{V}, i))} \leq \xi. \\
& \mathbf{v} \leq \mathbf{1}_{N \times 1}, \\
& \mathbf{v} \geq \mathbf{0}_{N \times 1}, \\
& \text{diag}(\mathbf{V}, 0) = \mathbf{v}, \\
& \begin{pmatrix} \mathbf{V} & \mathbf{v} \\ \mathbf{v}^T & 1 \end{pmatrix} \succeq 0.
\end{aligned} \tag{6.35}$$

Then, we can use the convex optimization to obtain the estimated  $\hat{\mathbf{v}}$ . However, the values of the components of  $\hat{\mathbf{v}}$  are the real value from 0 up to 1. Therefore, we need methods to find the  $\mathbf{v}$  we need.

We use randomized rounding algorithm in [6] to achieve the goal. It is as follows.

1. Generate  $L$  candidate estimates of the form  $v_{n,l} = 1(l = 1, 2, \dots, L)$  with a probability  $\hat{v}_n$  for  $n = 1, 2, \dots, N$ . The  $l$  starts from 1.

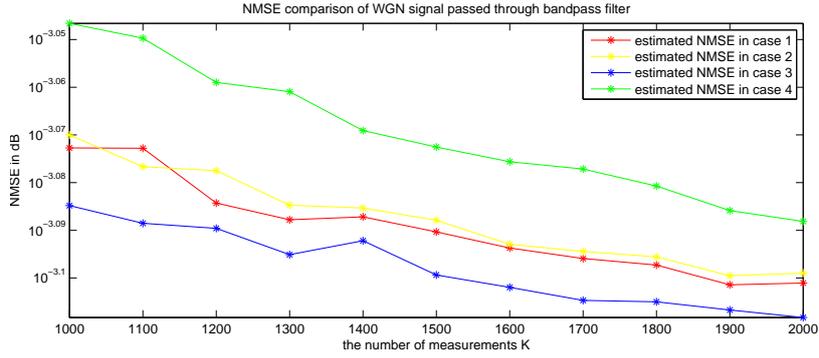


Figure 6.16: NMSE comparison of WGN signal passed through bandpass filter (in dB)

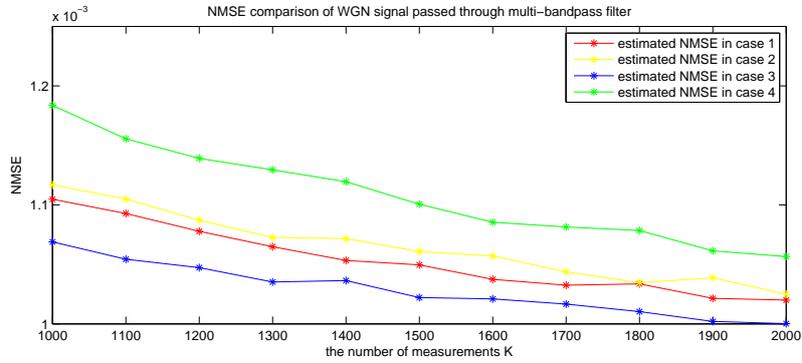


Figure 6.17: NMSE comparison of WGN signal passed through multi-bandpass filter

2. Define  $\mathbf{v}_l = [v_{1,l}, \dots, v_{N,l}]^T$ . Find the  $S_m$  of the  $\mathbf{v}_l$ .
3. If the index differences set  $\Omega_{S_m}$  of  $S_m$  satisfying the condition that  $\Omega_{S_m}$  covers from  $1 - N$  up to  $N - 1$ ,  $\mathbf{v}_l$  is the solution, otherwise go back to step 2 and move to another  $\mathbf{v}_{l+1}$ .

## 6.4 Simulation Study

In our simulation, we define the  $\xi = 0.02$ ,  $\sigma = 1$  and  $K = 10000$ , where  $\xi$  is the certain NMSW we set. We change the number of  $N$  to check whether we can obtain correct results.

First of all, we set  $N = 15$ , and then, we collect 7 samples, whose indices set is  $\{1, 3, 4, 10, 11, 14, 15\}$ . The index starts from 1. The NMSE of the estimated power spectrum with this sampling pattern is 0.0022. The distribution of the correlation lags is showed in Fig 6.19.

Then, we set  $N = 27$ , and then, we collect 10 samples, whose indices set is  $\{1, 2, 3, 13, 17, 19, 21, 24, 26\}$ . The NMSE of the estimated power spectrum with this sampling pattern is 0.0035. The distribution of the correlation lags is showed in Fig 6.20.

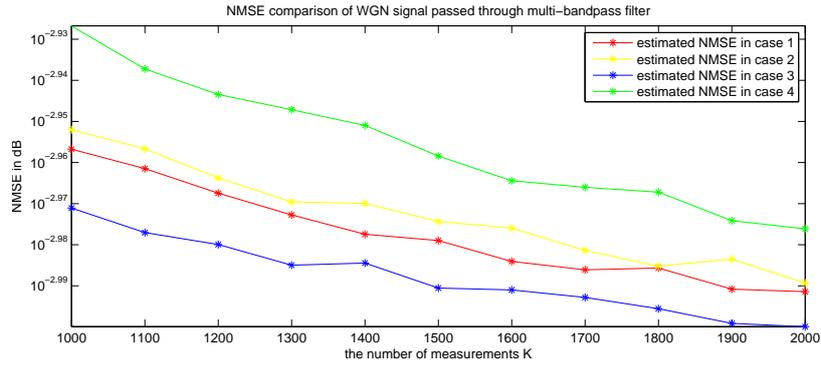


Figure 6.18: NMSE comparison of WGN signal passed through multi-bandpass filter (in dB)

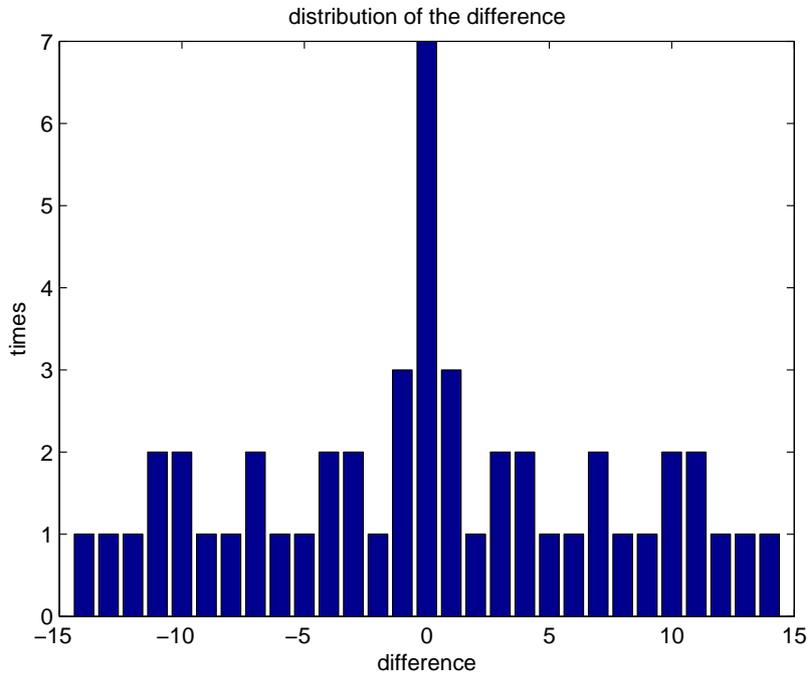


Figure 6.19: The distribution of the correlation lags when  $N = 15$

After that, we set  $N = 36$ , and then, we collect 11 samples, whose indices set is  $\{1, 2, 3, 7, 8, 22, 24, 25, 33, 35, 36\}$ . The NMSE of the estimated power spectrum with this sampling pattern is 0.0054. The distribution of the correlation lags is showed in Fig 6.21.

Last, we set  $N = 58$ , and then, we collect 16 samples, whose indices set is  $\{1, 2, 3, 6, 15, 17, 21, 23, 24, 29, 47, 52, 53, 54, 57, 58\}$ . The NMSE of the estimated power spectrum with this sampling pattern is 0.0071. The distribution of the correlation lags is showed in Fig 6.22.

However, we note that the NMSEs we obtain are much less than the value we set.

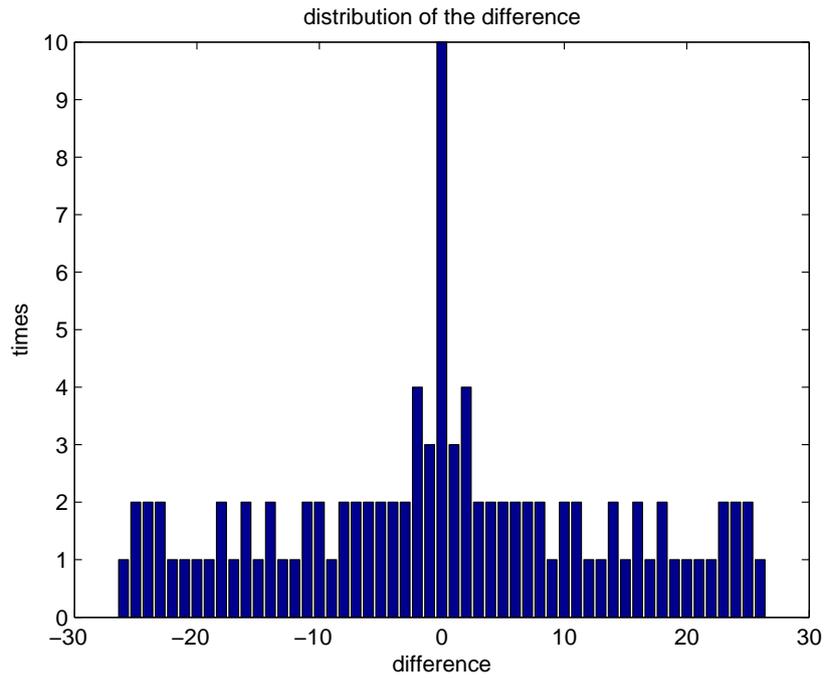


Figure 6.20: The distribution of the correlation lags when  $N = 27$

In other word, the problem is not perfectly solved. We do many relaxation to transform the problem into convex problem, hence, the results may not perfectly agree with the original problem.

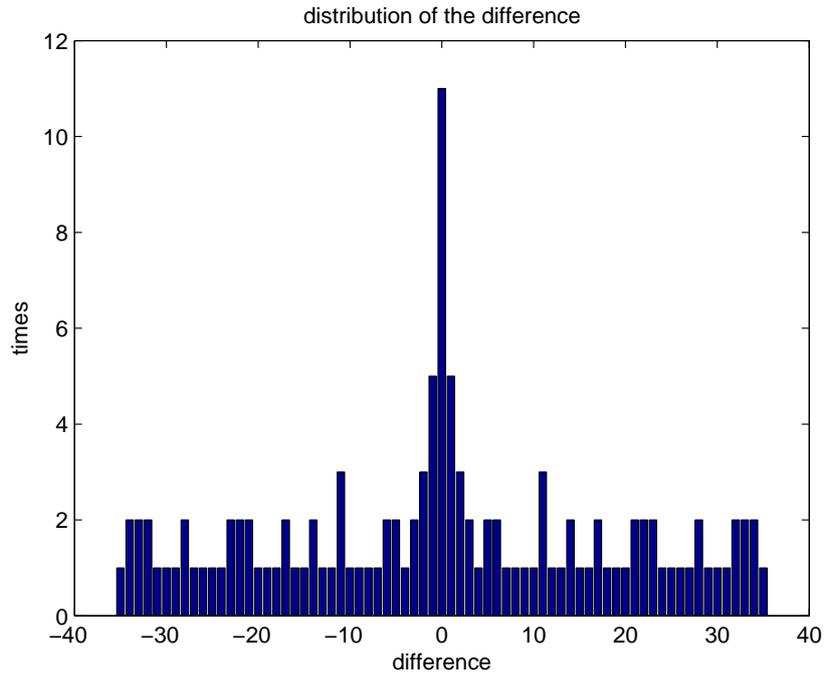


Figure 6.21: The distribution of the correlation lags when  $N = 36$

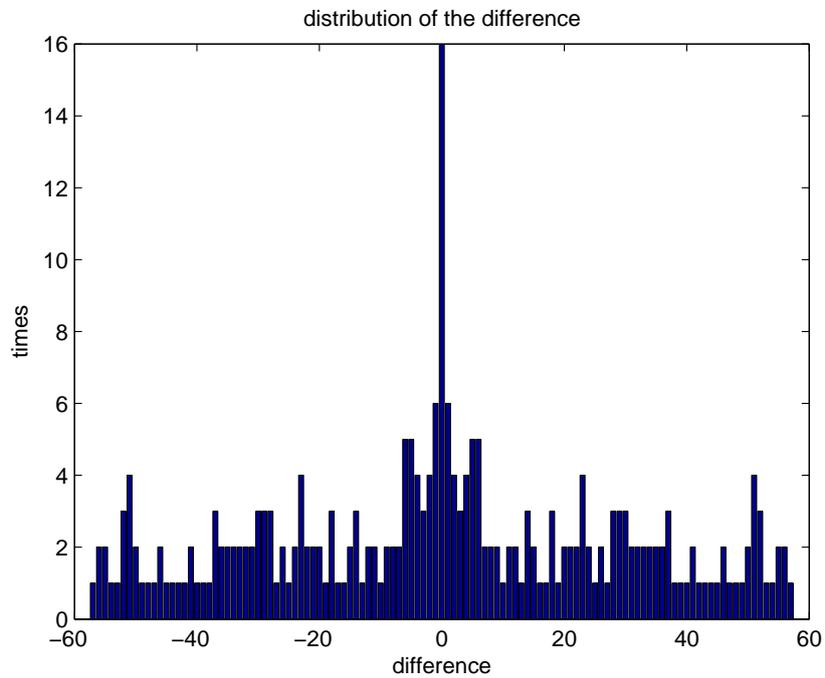


Figure 6.22: The distribution of the correlation lags when  $N = 58$



# Conclusion and Future Works

---

In this chapter we summarize the work in the thesis, draw the final conclusion and suggest directions for the further research.

## 7.1 Conclusion

In the thesis, we first present the introduction and the motivation to the compressive power spectrum sensing, from where we can understand the purpose of the compressive power spectrum sensing. For the demand of spectrum sharing, the information of the power distribution over frequency band is needed. Besides, the wide-band signal indicates a high sampling frequency if we use the uniform sampling according to the Whittaker–Nyquist–Kotelnikov–Shannon sampling theorem. The power consumption of the ADC with a high sampling rate is too huge for current ADCs, therefore we pay the attention on compressive power spectrum estimation.

In the second chapter, we present the background of the compressive power spectrum estimation. Specifically, we make a short introduction to the alternative time domain approach, which is the theoretical foundation of our thesis. In this thesis, the signal we implement as the original signal is always a wide-sense stationary signal. The power spectrum is calculated by performing the Fourier transform of the autocorrelations of the signal. Since we do not need to reconstruct the signal, we do not care about the information in the signal itself. What we want is the information in the autocorrelation of the signal. This allows us to estimate the power spectrum with less samples compared to the samples needed in classical uniform sampling method.

In the third chapter, we implement the parametric method and the non-parametric method on the alternative time domain approach. After comparing the performance of those methods, we can draw some conclusions as follows. When we choose the suitable signal model to estimate the signal, the performance of the parametric method is better than the performance of the non-parametric method. Moreover, in this case, the compression rate for the parametric method can be less than the compression rate for the non-parametric method, because the number of parameters in parametric approach is much less than the number of autocorrelations we need to calculate in non-parametric approach.

In the fourth chapter, we propose the parametric method to implement the compressive power spectrum estimation. Different from the parametric approach in the previous chapter, we estimate the parameters in the model we choose directly without calculating the autocorrelations of the signal first. From the simulations, we can see that, the compression rate can be further decreased, because we do not need to obtain all information of the autocorrelations at all when we still want to keep the accuracy of the power spectrum estimation. However, the methods here is not suitable for all the

cases. In some cases, we could not calculate the parameters, which is needed to make further works on.

In the fifth chapter, we focus on the sampling pattern based on the alternative time domain approach. The sampling pattern ensures the intact information in autocorrelations from lags  $1 - N$  to  $N - 1$ . The optimal sample selection is the minimal sparse ruler. However it is hard to obtain unless adapting brute force procedure. We propose some sub-optimal sparse ruler algorithms and compare them with the current sub-optimal sparse ruler algorithms. From the comparison we can find that the performance of our sub-optimal sparse ruler is better than the coprime sampling and nearly have the same performance compared to the two-level nested array algorithm. When the scale of the samples in a block is small, our cvx based sub-optimal sparse ruler algorithm has the best performance, very close to the performance of optimal sparse ruler. The disadvantage here is that our cvx based sub-optimal sparse ruler algorithm is more complex, which indicates a larger time cost when generating the sampling patterns.

In the sixth chapter, we analyse the performance of the alternative time domain approach and study the cosets selection. We find that the NMSE of the estimated power spectrum is related to the number of times the lags appear due to the cosets selection. The more often the lags appear the less the NMSE is. After obtain the relationship between the cosets selection and the NMSE, we can set the NMSE of the estimated power spectrum to a certain desired value and then find the best cosets selection under this condition.

## 7.2 Suggestion for Further Research

There are several problems remain to be solved in the future.

### 7.2.1 The optimization of the parameters estimation

As we have described in Chapter 4, our methods to estimate the parameters of the AR and ARMA models are not suitable for all the cases. The optimization method in our cases are restricted by the parameters in the models themselves. That means we still need to find an optimization method to estimate the parameters in the models suitable for all the cases.

### 7.2.2 The best solution of the cosets selection

Although we have found the relationship between the cosets selection and the NMSE of the estimated power spectrum, we do not solve the cosets selection problem very well. Because we make the relaxation on the problem to make it into a convex problem, the final result can reach the requirements of the relaxed convex optimization problem, not reach the optimal solution for the original problem. Therefore we still need to solve it in the future.



## A.1 RELAXATION AND CONVEX PROOF OF CHAPTER 6

Problem setting:  $\mathbf{v}$  is an  $N \times 1$  vector, whose element value belongs to  $\{0, 1\}$ , and  $\mathbf{V}$  is a matrix defined by  $\mathbf{V} = \mathbf{v}\mathbf{v}^T$ . We want to turn the constraint function A.1 which appears in Chapter 6, to be a convex function after some relaxation.

$$f(\mathbf{V}) = \sum_{i=1-N}^{N-1} \frac{1}{\text{sum}(\text{diag}(\mathbf{V}, i))} \quad (\text{A.1})$$

It is easy to know that all the elements in matrix  $\mathbf{V}$  are non-negative, which indicates that  $\text{sum}(\text{diag}(\mathbf{V}, i))$  is non-negative for all  $i$  from  $1 - N$  to  $N - 1$ . Note that  $\text{sum}(\text{diag}(\mathbf{V}, i))$  is related to  $\text{diag}(\mathbf{V}, j)$  when  $i$  and  $j$  are different, because all the  $\text{sum}(\text{diag}(\mathbf{V}, i))$  are calculated by the  $\mathbf{v}$ . Therefore we do some relaxation on it. Now we use positive  $p_i$  replacing  $\text{sum}(\text{diag}(\mathbf{V}, i))$  in the function A.2. So we have  $p_i$  for all  $i$  from  $1 - N$  to  $N - 1$ , where all  $p_i$  are independent with each other. Then the function becomes

$$f(\mathbf{p}) = \sum_{i=1-N}^{N-1} \frac{1}{p_i}, \quad (\text{A.2})$$

where  $\mathbf{p}$  is a  $(2N - 1) \times 1$  vector.

According to the definition of convex function, for all scale value  $a$  and  $b$ , where  $a + b = 1$  and  $a \geq 0$ ,  $b \geq 0$ , the function should satisfy the condition  $f(a\mathbf{p} + b\mathbf{q}) \leq af(\mathbf{p}) + bf(\mathbf{q})$  in order for  $f(\cdot)$  to be convex.

In our problem,

$$f(a\mathbf{p} + b\mathbf{q}) = \sum_{i=1-N}^{N-1} \frac{1}{ap_i + bq_i}, \quad (\text{A.3})$$

$$af(\mathbf{p}) = a \sum_{i=1-N}^{N-1} \frac{1}{p_i}, \quad (\text{A.4})$$

$$bf(\mathbf{q}) = b \sum_{i=1-N}^{N-1} \frac{1}{q_i}. \quad (\text{A.5})$$

$$(\text{A.6})$$

Then, take (A.6) into the condition  $f(a\mathbf{p} + b\mathbf{q}) \leq af(\mathbf{p}) + bf(\mathbf{q})$ , our task is to prove if  $\sum_{i=1-N}^{N-1} \frac{1}{ap_i + bq_i} \leq a \sum_{i=1-N}^{N-1} \frac{1}{p_i} + b \sum_{i=1-N}^{N-1} \frac{1}{q_i}$ .

Now the proof is given as follows. First we prove that  $\frac{1}{ap_i+bq_i} \leq a\frac{1}{p_i} + b\frac{1}{q_i}$ . We assume that  $\frac{1}{ap_i+bq_i} > a\frac{1}{p_i} + b\frac{1}{q_i}$ . Note that  $p_i, q_i$  are positive and at most one of  $a$  and  $b$  is zero.

$$\frac{1}{ap_i + bq_i} > a\frac{1}{p_i} + b\frac{1}{q_i} \quad (\text{A.7})$$

$$\frac{1}{ap_i + bq_i} > \frac{aq_i + bp_i}{q_i p_i} \quad (\text{A.8})$$

$$q_i p_i > (ap_i + bq_i)(aq_i + bp_i) \quad (\text{A.9})$$

$$q_i p_i > (a^2 + b^2)p_i q_i + ab(p_i^2 + q_i^2) \quad (\text{A.10})$$

Because  $a + b = 1$ ,  $(a + b)^2 = 1$ ,  $a^2 + b^2 = 1 - 2ab$ . Then we obtain

$$q_i p_i > (a^2 + b^2)p_i q_i + ab(p_i^2 + q_i^2) \quad (\text{A.11})$$

$$q_i p_i > (1 - 2ab)p_i q_i + ab(p_i^2 + q_i^2) \quad (\text{A.12})$$

$$2abq_i p_i > ab(p_i^2 + q_i^2) \quad (\text{A.13})$$

$$2q_i p_i > (p_i^2 + q_i^2) \quad (\text{A.14})$$

$$0 > (q_i - p_i)^2 \quad (\text{A.15})$$

which is impossible. So we have proved that  $\frac{1}{ap_i+bq_i} \leq a\frac{1}{p_i} + b\frac{1}{q_i}$ . Because it is suitable for all  $i$  from  $1 - N$  to  $N - 1$ , we have proved that the condition  $f(a\mathbf{p} + b\mathbf{q}) \leq af(\mathbf{p}) + bf(\mathbf{q})$  is satisfied. Therefore, the problem is convex.

# Bibliography

---

- [1] D.D. Ariananda and G. Leus. Compressive wideband power spectrum estimation. *IEEE Transactions on Signal Processing*, 60(9):4775–4789, 2012.
- [2] D.D. Ariananda and G. Leus. A study on cooperative compressive wideband power spectrum sensing. In *Information Theory in the Benelux and The 2nd Joint WIC/IEEE Symposium on Information Theory and Signal Processing in the Benelux*, pages 102–109, 2012.
- [3] D.D. Ariananda, D. Romero, and G. Leus. Cooperative compressive power spectrum estimation. In *IEEE 8th Sensor Array and Multichannel Signal Processing Workshop (SAM), 2014*, pages 97–100. IEEE, 2014.
- [4] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.
- [5] E.J. Candes, M.B. Wakin, and S.P. Boyd. Enhancing sparsity by reweighted  $l_1$  minimization. *Journal of Fourier analysis and applications*, 14(5-6):877–905, 2008.
- [6] S.P. Chepuri and G. Leus. Sparsity-promoting sensor selection for non-linear measurement models. *IEEE Transactions on Signal Processing*, 63(3):684–698, 2015.
- [7] D. Cohen and Y.C. Eldar. Sub-nyquist sampling for power spectrum sensing in cognitive radios: A unified approach. *IEEE Transactions on Signal Processing*, 62(15):3897–3910, 2014.
- [8] Federal Communications Commission et al. Unlicensed operation in the tv broadcast bands. *ET Docket*, (04-186), 2004.
- [9] Matlab document. Least-squares (model fitting algorithms):trust region algorithm. <http://nl.mathworks.com/help/optim>.
- [10] J. Gallier. The schur complement and symmetric positive semidefinite (and definite) matrices. *December*, 44:1–12, 2010.
- [11] M.H.J. Gruber. Statistical digital signal processing and modeling. *Technometrics*, 39(3):335–336, 1997.
- [12] M.H. Hayes. *Statistical digital signal processing and modeling*. John Wiley & Sons, 2009.
- [13] C. Herley and P.W. Wong. Minimum rate sampling and reconstruction of signals with arbitrary frequency support. *IEEE Transactions on Information Theory*, 45(5):1555–1564, 1999.
- [14] S.J. Kim and G.B. Giannakis. Rate-optimal and reduced-complexity sequential sensing algorithms for cognitive ofdm radios. *EURASIP journal on Advances in Signal Processing*, 2009:2, 2009.

- [15] P. Kolodzy and I. Avoidance. Spectrum policy task force. *Federal Commun. Comm., Washington, DC, Rep. ET Docket*, (02-135), 2002.
- [16] D. Maldonado, B. Le, and A. Hugine. Cognitive radio applications to dynamic spectrum allocation: a discussion and an illustrative example. In *2005 First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, 2005.*, pages 597–600. IEEE, 2005.
- [17] M. Mishali and Y.C. Eldar. Blind multiband signal reconstruction: Compressed sensing for analog signals. *IEEE Transactions on Signal Processing*, 57(3):993–1009, 2009.
- [18] S.M. Mishra and A. Sahai. Cooperative sensing among cognitive radios. In *IEEE International Conference on Communications, 2006. ICC'06.*, volume 4, pages 1658–1663. IEEE, 2006.
- [19] J. Mitola, III and G.Q. Maguire Jr. Cognitive radio: making software radios more personal. *Personal Communications, IEEE*, 6(4):13–18, 1999.
- [20] P. Pal and P.P. Vaidyanathan. Nested arrays: a novel approach to array processing with enhanced degrees of freedom. *IEEE Transactions on Signal Processing*, 58(8):4167–4181, 2010.
- [21] P. Pal and P.P. Vaidyanathan. Coprime sampling and the music algorithm. In *Digital Signal Processing Workshop and IEEE Signal Processing Education Workshop (DSP/SPE), 2011 IEEE*, pages 289–294. IEEE, 2011.
- [22] D. Romero and G. Leus. Compressive covariance sampling. In *Information Theory and Applications Workshop (ITA), 2013*, pages 1–8. IEEE, 2013.
- [23] D. Romero and G. Leus. Wideband spectrum sensing from compressed measurements using spectral prior information. *IEEE Transactions on Signal Processing*, 61(24):6232–6246, 2013.
- [24] A. Sahai, S.M. Mishra, T. Rahul, and K.A. Woyach. Cognitive radios for spectrum sharing. In *IEEE Signal Processing Mag.* Citeseer, 2009.
- [25] C.R. Stevenson, C. Cordeiro, E. Sofer, and G. Chouinard. Functional requirements for the 802.22 wran standard. *IEEE Std*, pages 802–22, 2005.
- [26] P. Stoica and R.L. Moses. *Spectral analysis of signals*. Pearson/Prentice Hall Upper Saddle River, NJ, 2005.
- [27] T. Tandra, S.M. Mishra, and A. Sahai. What is a spectrum hole and what does it take to recognize one? *Proceedings of the IEEE*, 97(5):824–848, 2009.
- [28] R. Venkataramani and Y. Bresler. Perfect reconstruction formulas and bounds on aliasing error in sub-nyquist nonuniform sampling of multiband signals. *IEEE Transactions on Information Theory*, 46(6):2173–2183, 2000.

- [29] wikipedia. Newton's method. [https://en.wikipedia.org/wiki/Newton's\\_method](https://en.wikipedia.org/wiki/Newton's_method). last modified on 25 May 2015.
- [30] Jr. William Green. course materials for 10.34 numerical methods applied to chemical engineering. *MIT OpenCourseWare*, 2006.
- [31] Y Yuan. A review of trust region algorithms for optimization. In *ICIAM*, volume 99, pages 271–282, 2000.
- [32] T. Yucek and H. Arslan. A survey of spectrum sensing algorithms for cognitive radio applications. *Communications Surveys & Tutorials, IEEE*, 11(1):116–130, 2009.
- [33] F. Zeng, C. Li, and Z. Tian. Distributed compressive spectrum sensing in cooperative multihop cognitive networks. *IEEE Journal of Selected Topics in Signal Processing*, 5(1):37–48, 2011.