2012

Implementation of spectral estimation techniques for range-doppler mapping

Praveen Ranganath

IMPLEMENTATION OF SPECTRAL ESTIMATION TECHNIQUES FOR RANGE-DOPPLER MAPPING

by

PRAVEEN RANGANATH

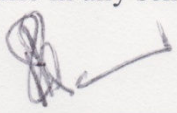
A THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Engineering
in
The Department of Electrical and Computer Engineering
to
The School of Graduate Studies
of
The University of Alabama in Huntsville

HUNTSVILLE, ALABAMA

2012

In presenting this thesis in partial fulfillment of the requirements for a master's degree from The University of Alabama in Huntsville, I agree that the library of this university shall make it freely available for inspection. I further agree that permission for extensive copying for scholarly purposes may be granted by my advisor or, in/her absence, by the Chair of the Department or the Dean of the School of Graduate Studies. It is also understood that due recognition shall be given to me and to The University of Alabama in Huntsville in any scholarly use which may be made of any material in this thesis.



(Student signature)

5/21/2012

(Date)

THESIS APPROVAL FORM

Submitted by Praveen Ranganath in partial fulfillment of the requirements for the degree of Master of Science in Engineering in Electrical Engineering and accepted on behalf of the Faculty of the School of Graduate Studies by the thesis committee.

We, the undersigned members of the Graduate Faculty of The University of Alabama in Huntsville, certify that we have advised and/or supervised the candidate on the work described in this thesis. We further certify that we have reviewed the thesis manuscript and approve it in partial fulfillment of the requirements for the degree of Master of Science in Engineering in Electrical Engineering.

5/30/12 Committee Chair
(Date)

May 21, 2012

6/1/12

6/4/12 Department Chair

06/13/12 College Dean

Rhonda Kay Gaede 7/5/12 Graduate Dean

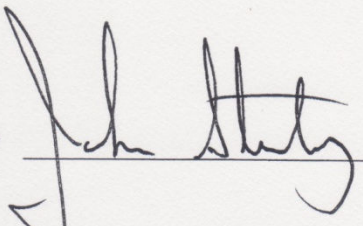
ABSTRACT

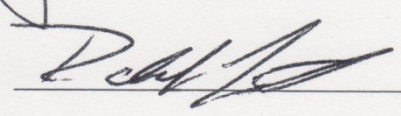
The School of Graduate Studies
The University of Alabama in Huntsville

Degree: Master of Science in Engineering College/Dept: Engineering/Electrical and Computer Engineering

Name of Candidate: Praveen Ranganath
Title: Implementation of Spectral Estimation Techniques for Range-Doppler Mapping

Standard spectral estimators such as a) periodogram and its windowed version b) correlogram and its windowed version (Blackman-Tukey), c) segmenting methods such as Bartlett and Welch, and d) autoregressive (AR) method are studied using a sinusoidal test signal. These methods are then applied to Range-Doppler (RD) mapping algorithm and the resulting Doppler resolutions are compared. Execution time, Bias and Variance of these estimators are also compared. RD map is simulated for a single target in the absence of clutter and scattering. RD map using AR method resulted in best Doppler resolution of 0.9 kHz, but AR method is the slowest in terms of execution time. Segmenting methods are not suitable for RD mapping due to poor resolution. Periodogram is the fastest method with reasonable Doppler resolution. Correlogram with superior Doppler resolution and small execution time would be the best method to choose for RD mapping.

Abstract Approval: Committee Chair  5/30/12

Department Chair  6/4/12

Graduate Dean Rhonda Kay Gaede 7/5/12

ACKNOWLEDGMENTS

The work described in this thesis would not have been possible without the assistance of a number of people who deserve special mention.

First, I would like to thank Dr. John Stensby for his patience and guidance throughout all the stages of the work. I would like to thank Dr. Mark Tillman and Dr. Nagendra Singh for serving as my thesis committee members.

I thank Salmon Library for providing me access to the books and journals that were very useful in completing this work.

Finally, I am thankful to my parents for their support.

TABLE OF CONTENTS

Page

List of Figures viii

List of Tables xi

List of Symbols xii

CHAPTER Page

1 **INTRODUCTION** 1

1.1 Introduction 1

1.2 Motivation 2

1.3 Objectives of the thesis 3

1.4 Chapter organization 4

2 **RADAR BASICS** 5

2.1 Introduction to Radar 5

2.2 Range 11

2.3 Doppler 14

2.4 Signal model 15

2.5 Range Doppler map 17

3 **BACKGROUND IN SPECTRAL ESTIMATION** 21

3.1 Introduction 21

3.2 Spectral densities 24

3.3 Estimator 28

3.4 Periodogram 29

3.5 Correlogram 35

3.6 Bartlett periodogram 38

3.7 Blackman Tukey correlogram 40

3.8 Welch periodogram 42

3.9 Auto-regressive method 45

4 **SIMULATION RESULTS** 48

4.1 Introduction 48

4.2 Simulation parameters 48
4.3 Periodogram estimators 51
4.3.1 Bias and variance of periodogram estimators 51
4.3.2 Range Doppler map using periodogram estimators 57
4.4 Correlogram estimators 62
4.4.1 Bias and variance of correlogram estimators 62
4.4.2 Range Doppler map using correlogram estimators 64
4.5 Autoregressive estimator 66
4.5.1 Bias and variance of AR method 67
4.5.2 Range Doppler map using AR method 68
4.6 Comparison of execution time 69
5 **CONCLUSIONS** 72
6 **APPENDIX A** 74
7 **APPENDIX B** 78
8 **APPENDIX** C 83
9 **REFERENCES** 96

LIST OF FIGURES

Figure Page

2.1 Basic working principle of radar..... 5

2.2 Block diagram showing the subsystems of radar..... 6

2.3 (a) Bistatic configuration, (b) Monostatic configuration. 7

2.4 Pulse width and the PRI..... 8

2.5 (a) Transmitted pulse for 1 PRI, (b) A Pulse burst for $N_P T = 20$ PRIs..... 10

2.6 Range resolution, (a) two unresolved targets, (b) two resolved targets..... 12

2.7 Rectangular pulse and its autocorrelation function..... 13

2.8 13-bit Barker pulse and its autocorrelation function..... 13

2.9 Illustration of the Doppler shift..... 14

2.10 Formation of CPI data matrix 17

2.11 Formation of Range Doppler map from CPI data matrix. 18

2.12 Range-Doppler map for a target using (a) Rectangular pulse, (b) Barker pulse..... 19

2.13 Range resolution obtained using (a) Rectangular pulse, (b) Barker pulse..... 19

2.14 Doppler resolution obtained using (a) Rectangular pulse, (b) Barker pulse..... 20

3.1 1 Hz sinusoid and its approximate spectrum 26

3.2 (a) 1 Hz sinusoid and its band-limited white Gaussian noise corrupted version …...27

3.3 Autocorrelation function and PSD of the corrupted version of the sinusoid in Fig. 3.2..... 27

3.4 Algorithm for the periodogram 29

3.5 Periodogram PSD estimate for N=16 and different M point DFTs as indicated in respective figures 31

3.6 Effects of the signal length N on the periodogram spectral estimate for M=1024 point DFT as indicated in the respective figures. 33

3.7 Algorithm for windowed periodogram 34

3.8 Algorithm for correlogram. 36

3.9 Effect of signal length N for Correlogram estimate, as indicated in the respective figures 37

3.10 Algorithm for Bartlett periodogram. 39

3.11 Bartlett periodogram estimate for (a) N=64, (b) N=512. 40

3.12 Algorithm for Blackman-Tukey method. 41

3.13 Blackman-Tukey periodogram estimate for (a) N=32, (b) N=512. 41

3.14 Welch periodogram estimate for (a) N=32, (b) N=512. 43

3.15 Algorithm for Welch periodogram. 44

3.16 AR YW, p=10 estimate for (a) N=32, (b) N=512. 47

4.1 (a) 13-Bit code for 1 PRI, (b)Real part of received signal modulated by frequency f_d with a delayed by time T_d , (c) Close up view of Fig. 4.1b 50

4.2 Comparison of noise-free PSD and periodogram for (a) N=32, N=512. 51

4.3 Bias and variance of the periodogram estimate for (a) N=16, (b) N=512 52

4.4 Comparison of noise free PSD and windowed periodogram using (a) Hamming window, (b) Hanning window, (c) Bartlett window, (d) Gaussian window. 53

4.5 Bias and variance of the PSD estimate of N=512 sequence using (a) Hamming window, (b) Hanning window, (c) Bartlett window, (d) Gaussian window. 54

4.6 Comparison of noise-free PSD and mean of estimated PSD using Bartlett periodogram for (a) N=32, (b) N=512 55

4.7 Bias and variance of Bartlett periodogram estimate for (a) N=32, (b) N=512.....55

4.8 Comparison of noise-free PSD and mean of the estimated PSD using Welch method for (a) N=32, (b) N=512 56

4.9 Bias and variance of the Welch estimate for (a) N=32, (b) N=512. 57

4.10 Periodogram (a) Range Doppler map, (b) Doppler resolution. 58

4.11 Gauss windowed Periodogram (a) Range Doppler map, (b) Doppler resolution. 59

4.12 Bartlett Periodogram (a) Range Doppler map, (b) Doppler resolution. 60

4.13 Gauss windowed Welch periodogram (a) Range Doppler map, (b) Doppler resolution. 61

4.14 Comparison of noise-free PSD and mean of estimated PSD using the correlogram for (a) N=32, (b) N=512. 62

4.15 Bias and variance of the correlogram estimate for (a) N=32, (b) N=512. 63

4.16 Comparison of true PSD and mean of the estimated PSD using Blackman-Tukey method for (a) N=32, (b) N=512. 63

4.17 Bias and variance of the Blackman –Tukey estimate for (a) N=32, (b) N=512. 64

4.18 Correlogram (a) Range Doppler map, (b) Doppler resolution..... 65

4.19 Blackman-Tukey correlogram (a) Range Doppler map, (b) Doppler resolution..... 66

4.20 Comparison of true PSD and mean of the estimated PSD using AR YW, p=10 for (a) N= 32, (b) N=512..... 67

4.21 Bias and variance of the Yule-Walker p=10 estimate for (a) N=32, (b) N=512..... 67

4.22 AR YW, p=10 (a) Range Doppler map, (b) Doppler resolution..... 68

4.23 Execution time of the spectral estimators for 1 column of range-Doppler map.....70

LIST OF TABLES

Table Page

4.1 Radar parameters for the simulated data..... 49

4.2 Comparison of execution times of spectral estimators for one column and for the entire Range-Doppler map 69

LIST OF SYMBOLS

M	Number of points in a DFT
N	Length of the observation signal
τ	Pulse width of the transmitted wave
t	Continuous time
f	Frequency
$x_p(t)$	pulse of length τ
N_P	Number of pulses in the burst
T	Pulse Repetition Interval
c	Speed of light in Air
R	Range to the target
Δt	Two way time taken for the wave to travel from radar to target and back
ΔR	Range resolution
f_d	Doppler Shift
v_r	Radial velocity of the target
λ	Wavelength of a wave
$P_F(t)$	Modulating Pulse
f_0	Carrier frequency
T_d	Time delay
$n(t)$	Noise term in the received signal
$E(f)$	Energy spectral density
$P(f)$	Power spectral density
$X(f)$	Fourier transform of a signal x[n]
R_0	Initial range of the target
n	Discrete time index
m	DFT index

Chapter 1

INTRODUCTION

1.1 Introduction

The radar signal processing, which has evolved as an important branch of engineering is essentially the union of electromagnetics and signal processing. We are reminded of the rotary antenna when the term RADAR is encountered. The term was coined by the U.S. Navy and is in use since the 1940's.

The radar essentially uses the fundamental electromagnetic device, the antenna. The transmitter antenna sends out a series of electromagnetic (EM) waves, most commonly the microwaves, to illuminate the area of interest. The microwaves used in the radar have all-time and all-weather capability, which are the prominent feature in radars that enable them to operate anytime and anywhere. Based on the dielectric and conductive properties of the scatterer, some amount of the transmitted signal is absorbed, and the rest is scattered back, which is further attenuated by the clutter entities during the transit. This signal is picked up by the receiver antenna. Then, it is signal processed into a comprehensible format.

One of the key components for processing the radar data is the spectral estimation. Here, the power spectral density is computed to characterize the frequency content of the signal. Since the radar data predominantly relies on the phase history of the scatterers, spectral estimation plays a major role in revealing the Doppler shifts that are used in the computation of the target velocity.

There are two important qualities of an estimator, bias and consistence. Bias is the difference of the expected value of the estimate and that of the actual value of the parameter. Consistency means how the variance of the estimate varies as the number of measurements keeps increasing. A good estimator must be unbiased, meaning bias equals zero, and must be consistent, that is the variance of the estimate decreases to zero as more measurement becomes available.

1.2 Motivation

A significant amount of interest has been developed by researchers belonging to various fields to apply spectral analysis on time series [7] - [11]. But, the recent works [4], [5] were focused on the spectral estimation for a signal with three sinusoids buried in white Gaussian noise. Sadate [4] emphasized the non parametric methods, the Welch periodogram and its modified versions. Garner [5] focused more on autoregressive spectral estimators which is a class of parametric spectral estimators. But however, in both [4] and [5], the application was limited to a sum of sinusoids, and the bias and variance were not detailed. As a result, in this thesis, range-Doppler mapping was chosen. In addition, the bias and variance of the estimators are examined, in order to understand the behavior of the spectral estimators.

1.3 Objectives of the thesis

The main goal of this thesis is to test the most commonly used non parametric spectral estimation techniques, and use them to investigate the Doppler resolution of range- Doppler map for a simple target detecting radar. Doppler resolution is compared for non- parametric spectral methods such as a) periodogram, b) correlogram, c) averaged periodogram and d) modified correlogram and e) autoregressive method using Yule-Walker algorithm. Averaged periodogram methods include Bartlett and Welch periodogram, and Blackman-Tukey method is the modified correlogram that will be used in this work. The bias and variance of the estimators are studied, which vary with the different estimators and increasing observation lengths.

The FFT is a conventional means of spectral analysis performed for range-Doppler mapping in most radars [1]. In order to examine the range-Doppler map resolution as mentioned before, we utilized the simulated data for an aircraft target at a range R with a Doppler frequency f_d. For simulation, we use MATLAB on a personal computer. The received signal containing the target information is matched filtered with the transmitted signal. The spectral estimation techniques mentioned earlier are applied to the output of the matched filter to get the range-Doppler map. The main parameters in the range-Doppler map being the range resolution and the Doppler resolution are compared.

Using MATLAB programs, it will be demonstrated that the Barker coded pulse yields better range resolution than the rectangular pulse. Also, the Doppler resolution is compared using the spectral estimation techniques, and the bias and variance of the estimators are studied.

1.4 Chapter organization

In Chapter 2, theoretical background of radars and the signal model of radar waveforms are presented. In Chapter 3, a background of the non-parametric spectral estimation and the autoregressive method is discussed. This chapter illustrates the differences in the approaches used by the FFT based non-parametric methods and the autoregressive method. In Chapter 4, we discuss the simulation results of range-Doppler map resolution and statistical characteristics of the spectral estimators of Chapter 3. Further, to understand the behavior of the estimators in a better manner, the bias, variance and the performance in terms of execution time are investigated. In Chapter 5, we see that the AR estimator is the best estimator in terms of resolution and the slowest in terms of execution time. Chapter 5 concludes by comparing the performance of the estimators discussed in Chapter 4 based on visual perception of the computer generated results, along with the possible developments to this work.

Chapter 2

RADAR BASICS

2.1 Introduction to Radar

Radar is a system of electrical devices that transmits EM waves, in particular the radio waves and microwaves, towards an area of interest and receives the waves reflected by the objects in that region. Fig. 2.1 shows the basic working of radar.

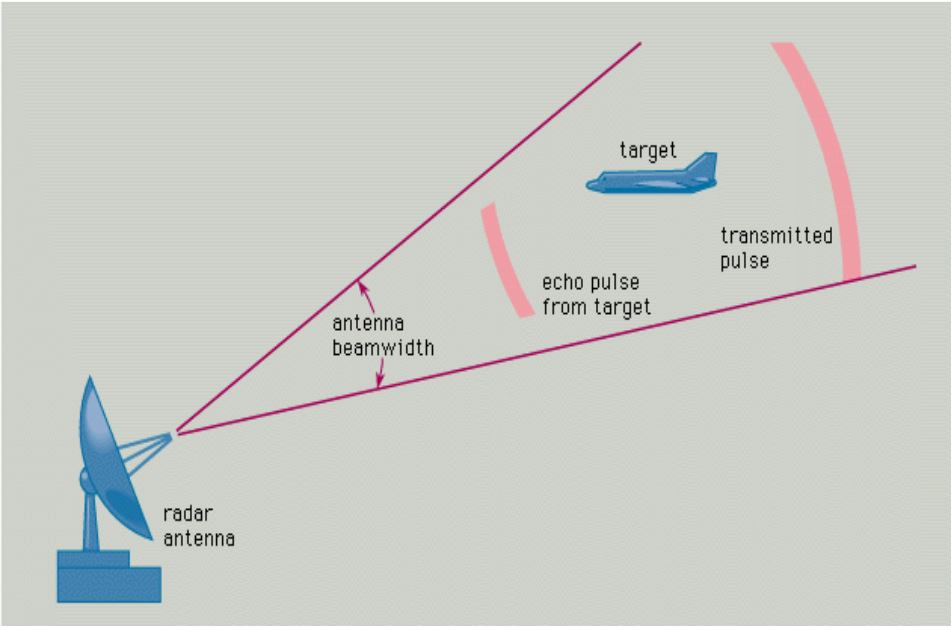


Fig. 2.1: Basic working principle of Radar

The transmitted signal, when incident on any surface, is reflected back. The reflected signals from the targets are called echoes, while the signals reflected from unwanted surfaces are collectively called clutter. The strength and quality of both the transmitted and echo signals are affected by noise, clutter and the atmosphere. Common subsystems in radar are shown in Fig. 2.2.

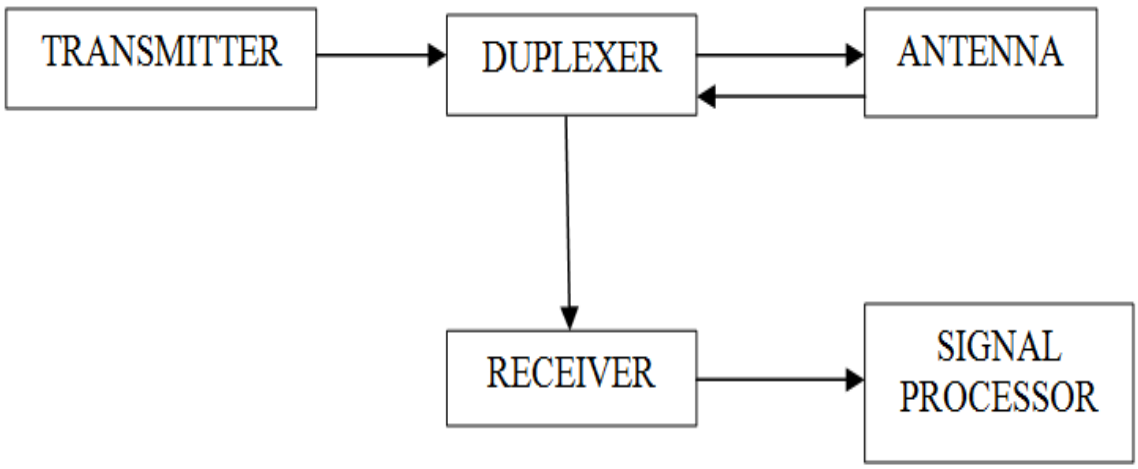


Fig. 2.2: Block diagram showing the subsystems of Radar

The circuit connecting transmitter and the receiver to the antenna at all times is called the duplexer. Since the transmitted signal is of very high power, the duplexer maintains electrical isolation between the transmitter and the receiver, in order to protect the sensitive components of the receiver that respond to signal power levels in the order of milliwatts to nanowatts [1]. The main parts of the receiver consist of: filters, those minimize the effect of unwanted signals, amplifiers, to improve the signal characteristics, local oscillators and detectors, to synchronize with the carrier frequency.

A radar system is non-coherent if it detects only the amplitude of the returning signal. Though, the non-coherent radars are primitive, yet they are still used in some applications where it is just sufficient to display the position of the target in a ground map background, and where it is known that the signal from the target can overcome the effects of clutter. A coherent radar system measures both amplitude and phase of the received signal. The presence of phase enables the system to track the motion characteristics of the changing target. To detect the phase change, coherent systems use a reference sinusoid at the transmitting and the receiving ends.

Depending on the antenna configuration, radars can be categorized into Monostatic or Bistatic. In the Monostatic configuration, the transmitting and the receiving antennas are located close to one another. But, in contrary the Bistatic configuration employs two physically separate antennas located at a sufficient separation to perform the transmission and reception operations as in Fig. 2.3.

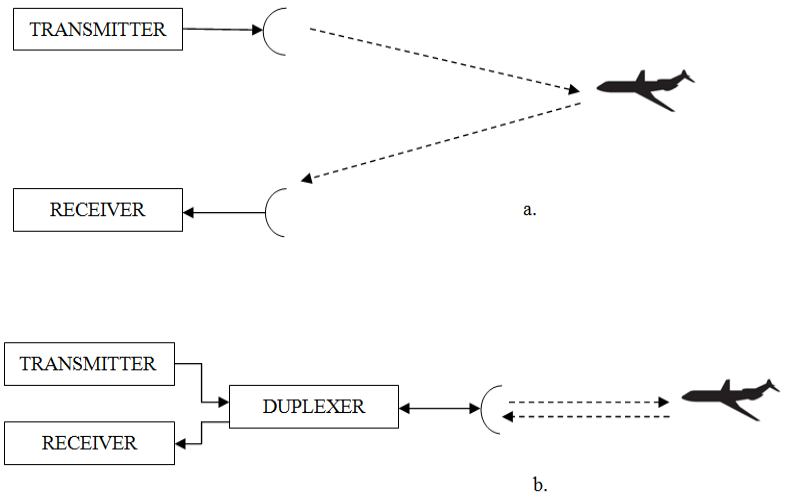


Fig. 2.3: (a) Bistatic configuration, (b) Monostatic configuration

Based on the waveform of the transmitted wave, radars can be divided into two classes, continuous wave (CW) radars and pulsed radars. In the CW radars, the transmitter operates all the time, while the pulsed radars emit a sequence of finite duration pulses at regular intervals.

In pulsed radars, the transmitted signal is ON for a duration τ called the pulse width or the ON time. The periodic interval between the two transmitted pulses is called Pulse Repetition Interval (PRI). The time interval between the end of ON time and the start of next pulse is the listening time as shown in Fig. 2.4.

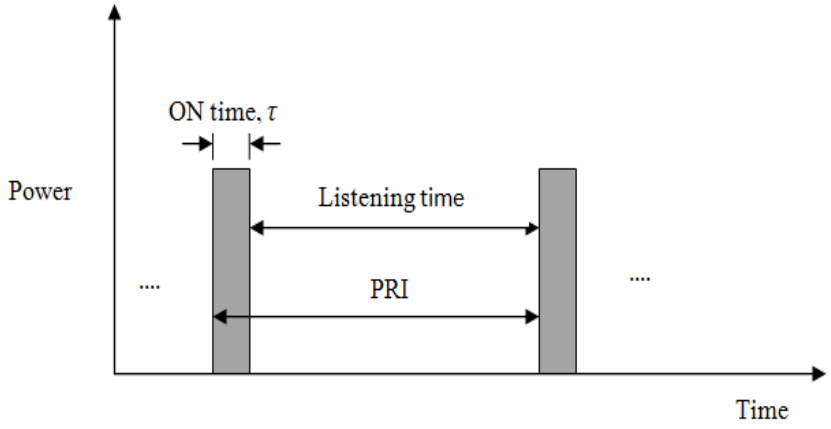


Fig. 2.4: Pulse width and the *PRI*

The pulse repetition frequency (PRF) is the number of transmit/receive cycles, completed in one second by the radar, given by

$$PRF = \frac{1}{PRI} \ . \qquad (2.1)$$

Better frequency resolution requires longer observation time, since frequency and time have inverse relationship. One way to meet this requirement is to send a pulse burst waveform instead of a single pulse. A pulse burst waveform is defined [1] by

$$x(t) = \sum_{m=0}^{N_P-1} x_p(t - mT), \tag{2.2}$$

where $x_p(t)$ is the single pulse of length τ, N_P is the number of pulses in the burst, and T is the pulse repetition interval.

The total duration of the pulse train, which is $N_P T$, is known as the Coherent Processing Interval (CPI). Fig. 2.5 below shows a single pulse of τ =1μs, T=20μs and a pulse burst for N_P=20. Hence the pulse burst in Fig. 2.5b has a CPI of 400 μsec. Operations that are applied to data from a single pulse are referred to as fast time because of shorter time scale which is one PRI, or in other words higher sample rates. Operations performed on a pulse burst involve larger time scale of one CPI. Such operations are referred to as slow time. The slow time processing that will be used in this thesis is Doppler processing.

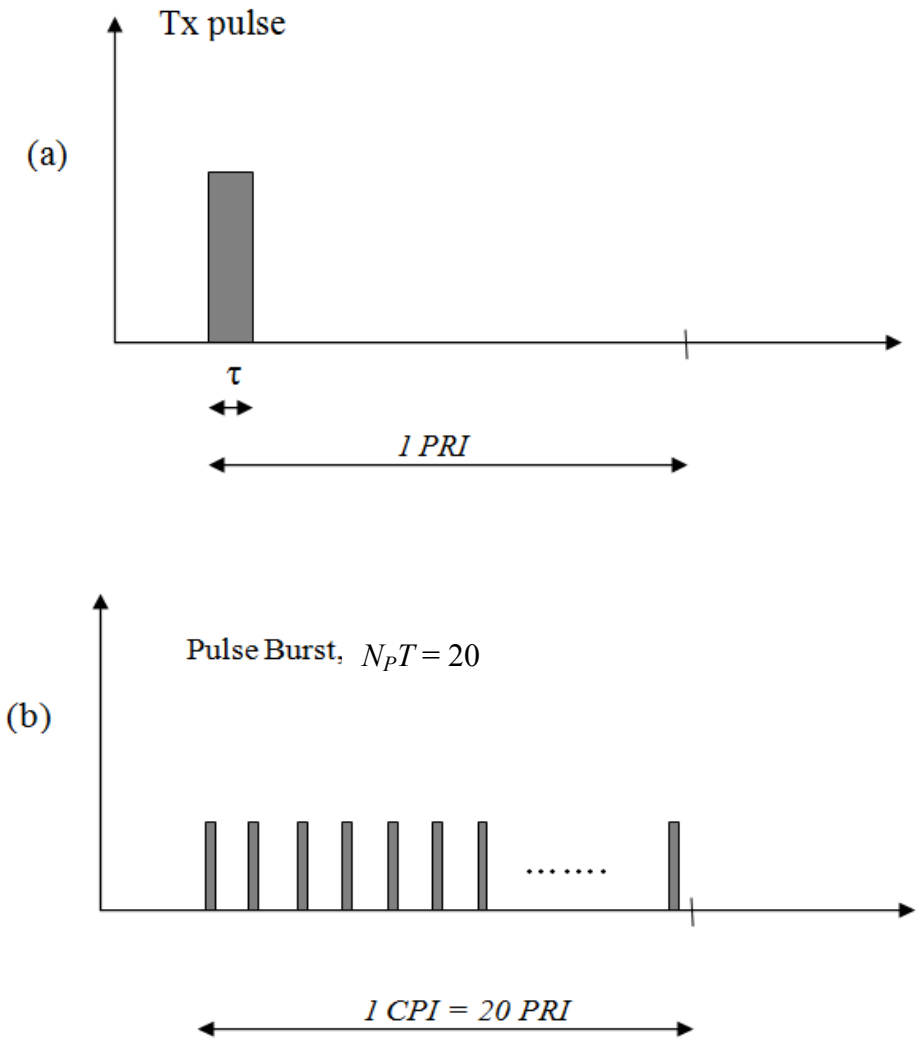


Fig. 2.5: (a) Transmitted pulse for 1 PRI, (b) A pulse burst for $N_P T = 20$ PRIs

2.2 Range

The distance between the radar and the target is the range, usually denoted by R, is given by

$$R = \frac{c\Delta t}{2}. \tag{2.3}$$

Here, c is the speed of light in meters per second, Δt is the time in seconds for the two-way travel. The maximum unambiguous range is the maximum distance, which the radar can detect the range with certainty.

Resolution is defined [1] as the ability of the radar to discern multiple targets that are spaced closely in range, Doppler frequency or angle. To understand the idea behind resolution, we will consider only the range resolution here. The minimum distance to successfully resolve two closely spaced targets is called the range resolution, ΔR. The minimum value of ΔR is $c\tau/2$ [6]. This is illustrated in Fig. 2.6.

As we can see from Fig. 2.6a, two targets are spaced very closely so that their combined return appears as a single target to the system, whereas in Fig. 2.6b, the targets are spaced well enough to be resolved. Thus, shorter the pulse width, better is the range resolution.

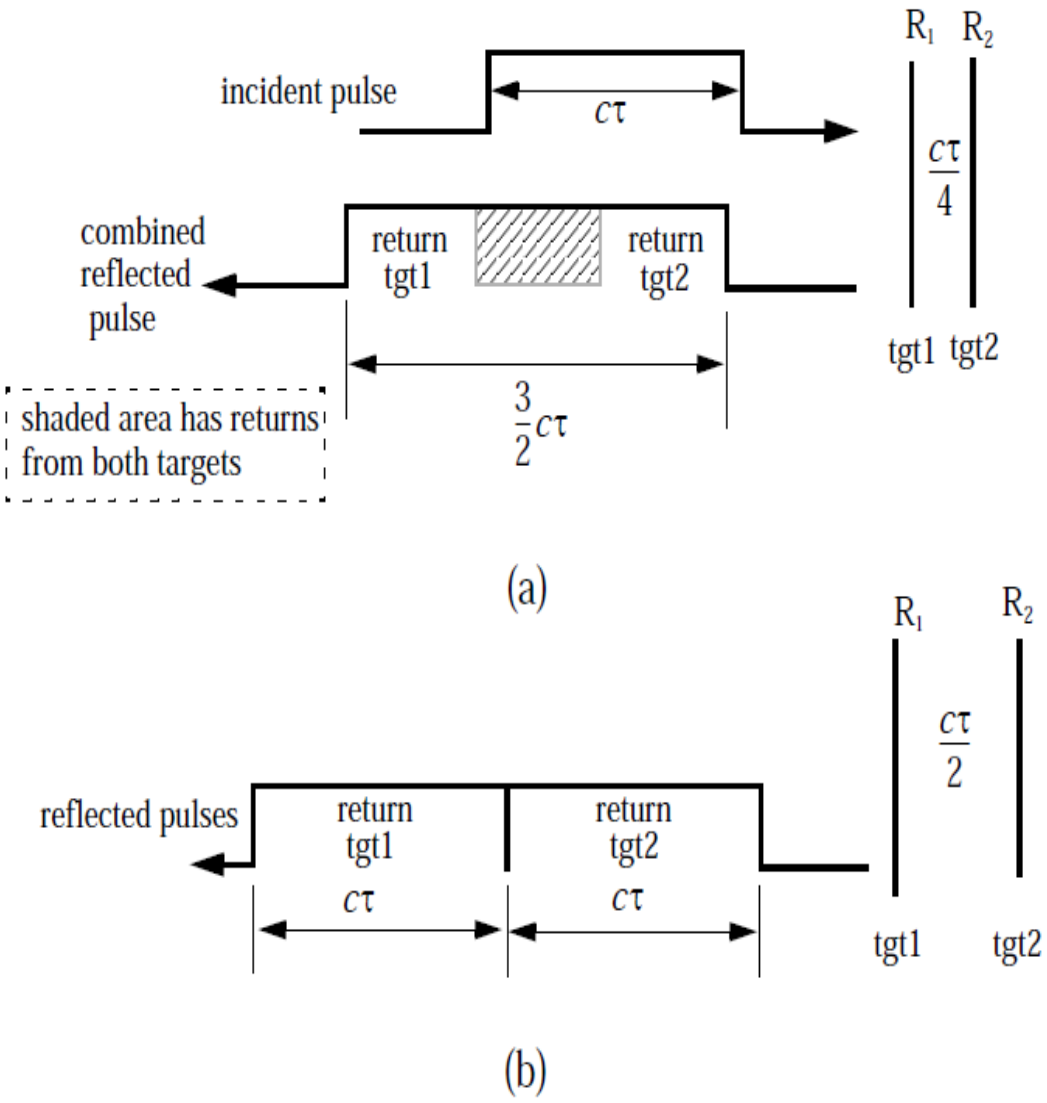


Fig. 2.6: Range resolution, (a) two unresolved targets, (b) two resolved targets

But, a short pulse increases the resolution at the cost of reduced transmitted power. Hence, to have a trade off, pulse compression is employed which improves range resolution without significantly reducing the transmitted power. Biphase coding is one of the pulse compression techniques that have only two phases: zero and 180 degrees, whereas, a polyphase-coding has more than two phases. Barker Code is one of the

commonly used Biphase coding schemes employed in compressing the radar waveforms [1]. Fig. 2.7 and 2.8 show the waveform and their autocorrelation function for the rectangular pulse and 13-bit Barker coded pulse, respectively.

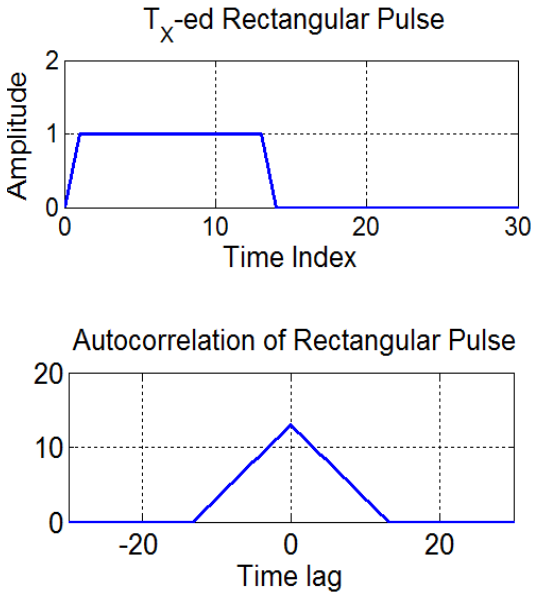


Fig. 2.7: Rectangular pulse and its autocorrelation function

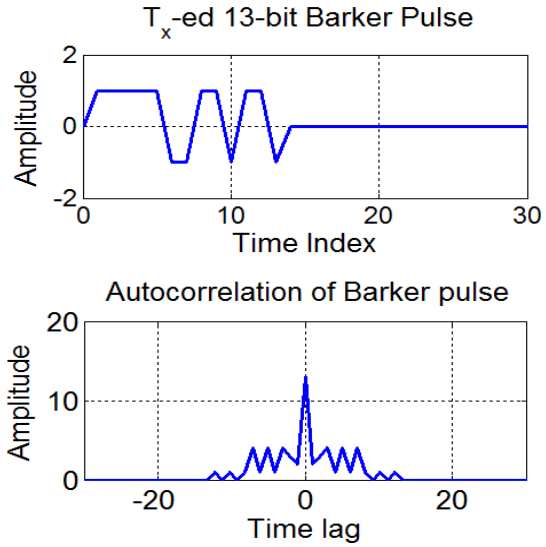


Fig. 2.8: 13-bit Barker pulse and its autocorrelation function

Due to the sub-pulses present in the Barker pulse, the autocorrelation function for the Barker pulse is narrower than the autocorrelation function of the rectangular pulse.

2.3 Doppler

When there is a relative motion between the radar and the target, the received wave has a different frequency $f + f_d$ from that of the transmitted wave f. This difference f_d is called the Doppler frequency shift, or simply Doppler, is given by

$$f_d = \frac{2v_r}{\lambda}. \tag{2.4}$$

In Eq. 2.4, v_r is the radial component of the target velocity, and λ is the wavelength of the transmitted wave. When f_d is positive, it means that the target is approaching the radar, while a receding target gives a negative f_d as shown in Fig. 2.9.

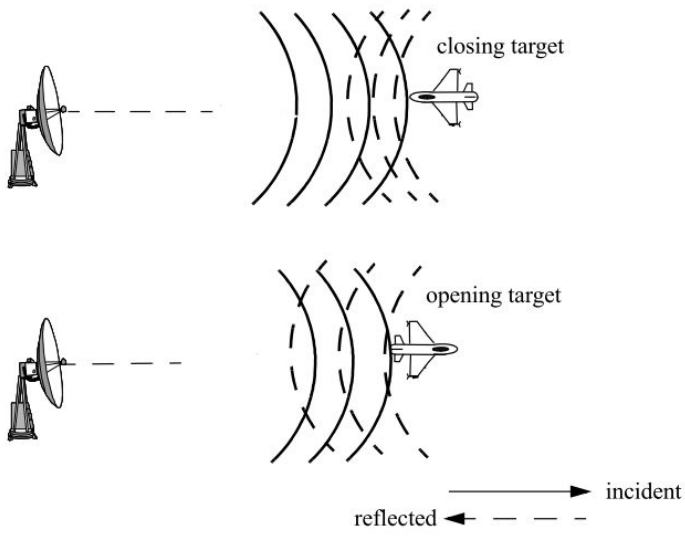


Fig. 2.9: Illustration of the Doppler shift

The Doppler shift is an important parameter in radar measurements, since it has the capability to detect a moving target and distinguishes the clutters. For example, in military radars, returns from the clutter will have a low Doppler shift, whereas the moving target return will have a higher Doppler shift.

2.4 Signal model

We consider a simplified mathematical model, where the effects of scattering and clutter are ignored. Modulated with a pulse $P_F(t)$, the transmitted signal of the radar is a complex sinusoid expressed as

$$x_{Tx}(t) = A_t P_F(t) \exp(j2\pi f_0 t), \tag{2.4}$$

where A_t is the amplitude of the transmitted signal, and f_0 is the carrier frequency. The received signal, reflected by the target, is modeled as

$$x_{Rx}(t) = A_r P_F(t - T_d) \exp(j2\pi f_0 (t - T_d)). \tag{2.5}$$

In Eq. 2.5, A_r is the amplitude of the received signal, and the time delay is given by

$$T_d = \frac{2R}{c} = \frac{2(R_0 - v_r t)}{c}, \tag{2.6}$$

where R_0 is the initial range at some time t_0.

Substituting for T_d in the first term exponential function of Eq. 2.5, we get

$$x_{Rx}(t) = A_r P_F(t - T_d) \exp\left(j2\pi f_0 \left(t - \frac{2(R_0 - v_r t)}{c} \right) \right) . \tag{2.7}$$

$$x_{Rx}(t) = A_r P_F(t - T_d) \exp\left(j\left(2\pi f_0 t - \frac{4\pi f_0 R_0}{c} + \frac{4\pi f_0 v_r t}{c} \right) \right). \tag{2.8}$$

In the exponential function of Eq. 2.8, the first term is simply the transmitted frequency, the second term is a constant, and the third term contains the Doppler information.

Eq. 2.8 is heterodyned to remove the first term containing the carrier frequency f_0, and low pass filtered to remove the second term to obtain

$$x'_{Rx}(t) = A_r P_F(t - T_d) \exp\left(j2\pi f_d t \right) , \tag{2.9}$$

where,

$$f_d = \frac{2v_r}{\lambda} . \tag{2.10}$$

To maximize the Signal-to-noise ratio (SNR) [1], usually the received signal in Eq. 2.9 is matched filtered with the transmitted pulse $P_F(t)$. Matched filtering is the operation of correlating the received signal with a copy of the transmitted signal [1].

2.5 Range Doppler map

The maximum unambiguous range is usually divided into sub ranges called range cells or range bins. Since there are N_P Barker coded pulses in a CPI, the received signal of each pulse is match filtered with the copy of transmitted signal. The resulting filtered signal of each received pulse is sampled and stored in a row of range bins. The N_P matched filtered rows form a CPI data matrix as shown in Fig. 2.10.

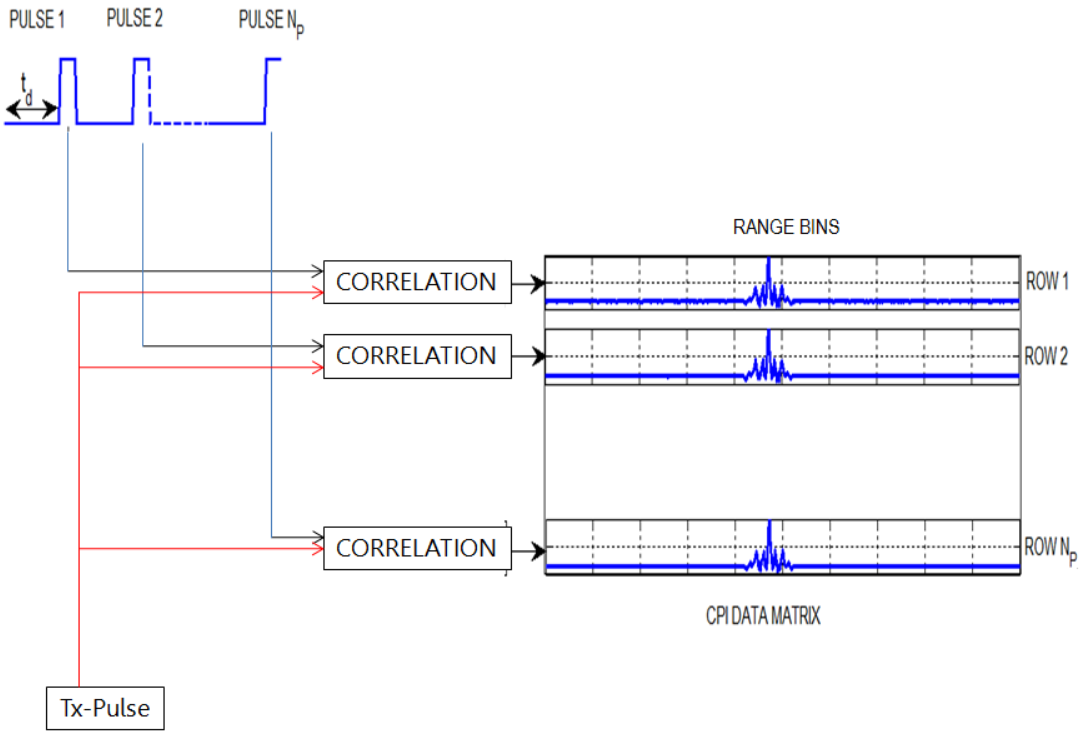


Fig. 2.10: Formation of CPI data matrix

To locate the position of the target, the Range-Doppler map is required. It is a visual representation of the range versus Doppler. The Doppler axis shows the Doppler shift f_d,

resulting due to the velocity of the target. The range axis shows the distance of the target from the Radar. This is illustrated in Fig. 2.11.

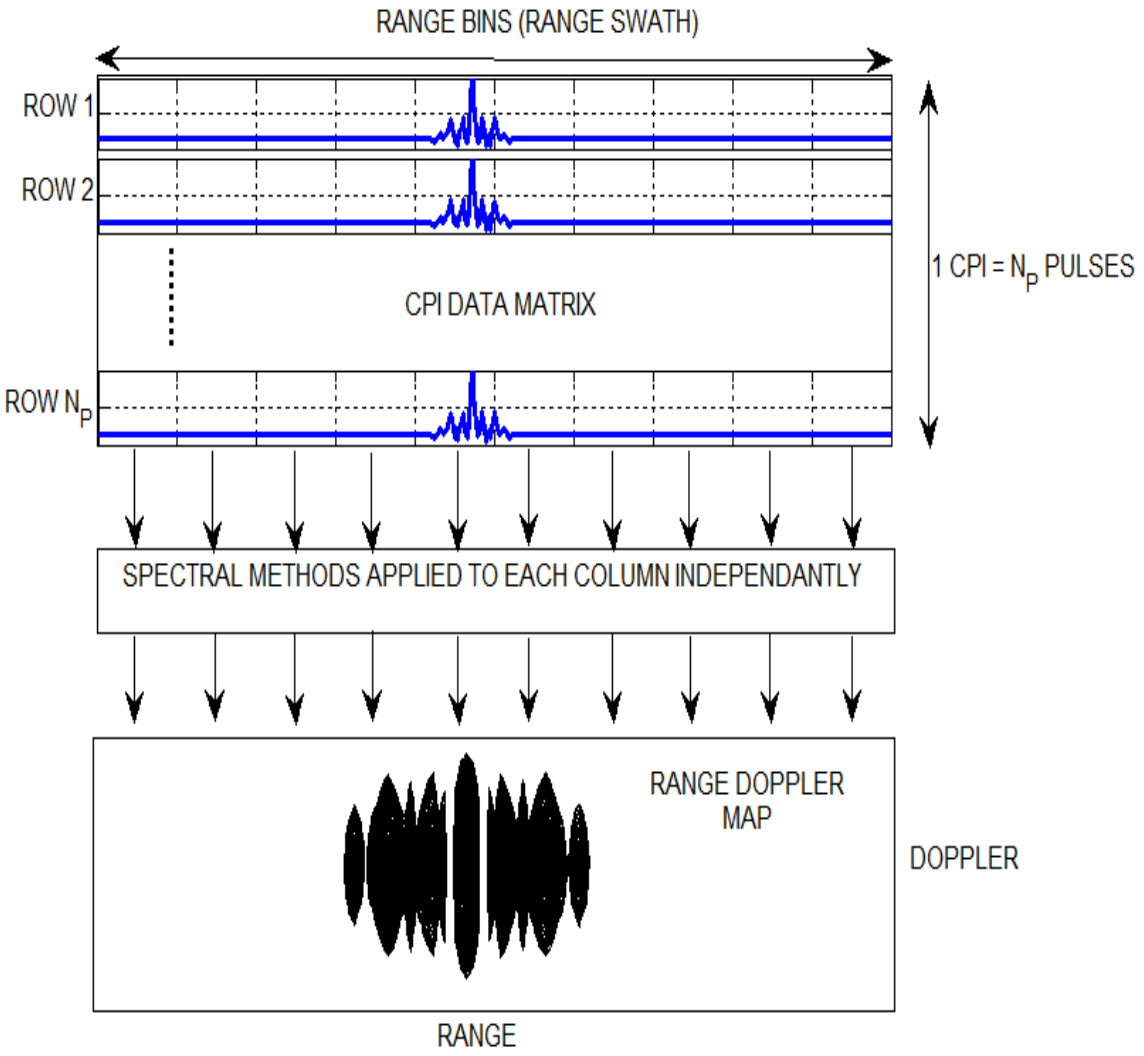


Fig. 2.11: Formation of range Doppler map from CPI data matrix

Fig. 2.12a and 2.12b shows the Range-Doppler map resulting from a rectangular pulse and a Barker pulse for target at a range 8.1 km with a Doppler shift of 25.5 kHz. The actual range and the Doppler are not clear due to the width of the main lobe.

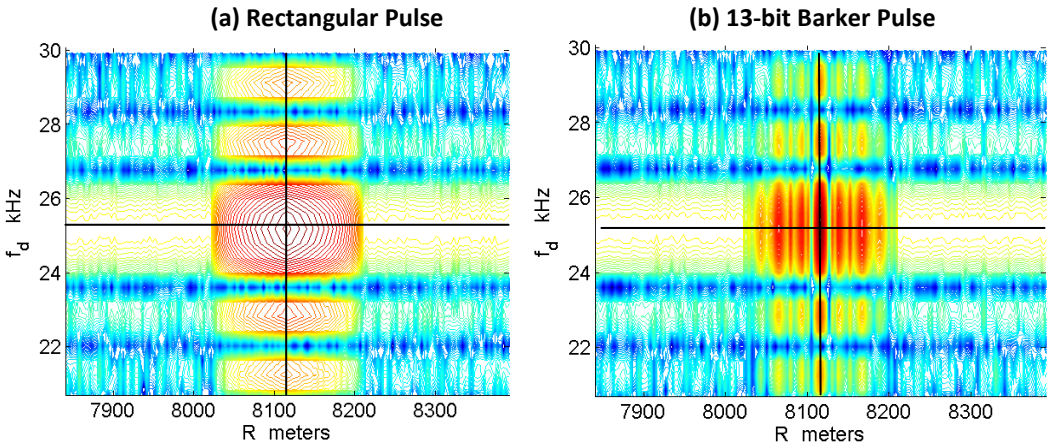


Fig. 2.12: Range-Doppler map for a target using (a) Rectangular pulse, (b) Barker pulse

To compare the range resolution, cut-plot from the range Doppler map in Fig. 2.12 is obtained along the Doppler frequency 25.5 kHz as shown in Fig. 2.13.

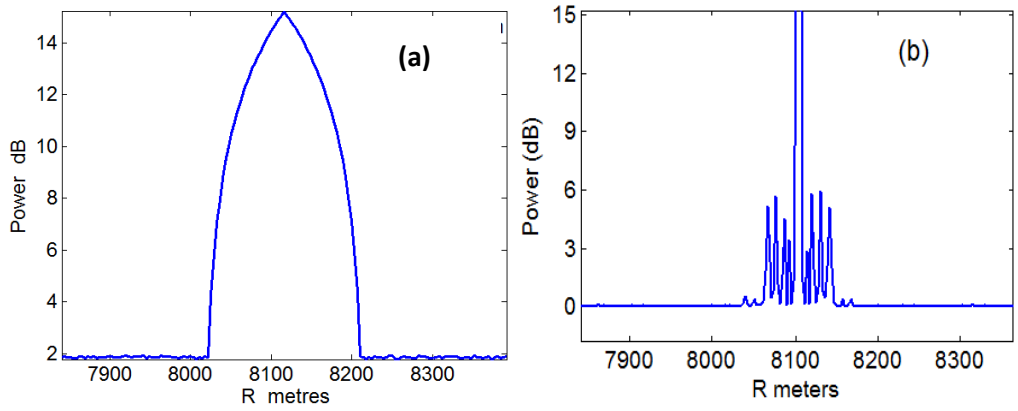


Fig. 2.13: Range resolution obtained using (a) rectangular pulse, (b) Barker pulse

By visually inspecting the width of the range resolution in Fig. 2.13, we find that the rectangular pulse has less resolution of 180 meters, while the range resolution provided by the Barker code is 20 meters.

For comparing the Doppler resolution, a cut plot along the range 8.1km is obtained from the Range Doppler map as in Fig. 2.14.

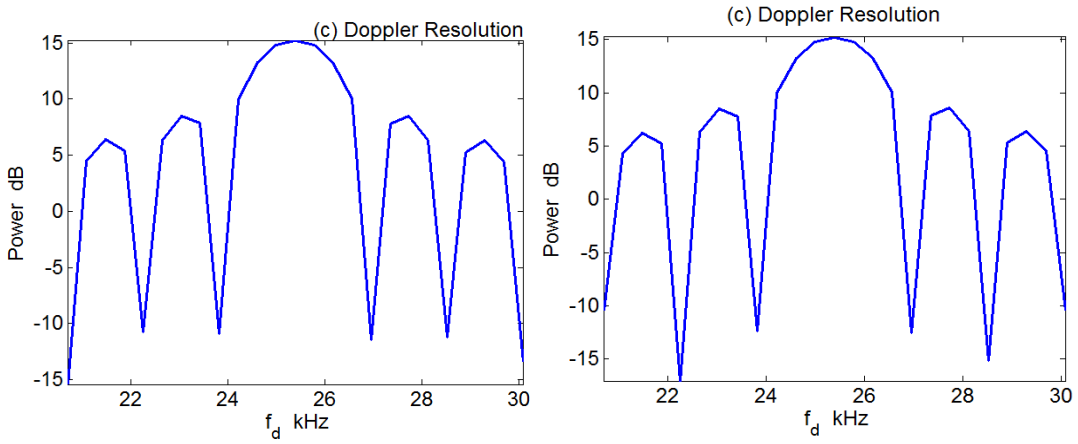


Fig. 2.14: Doppler resolution obtained using (a) rectangular pulse, (b) Barker pulse

From Fig. 2.13, it is clear that the use of Barker code provides a better resolution in range. In this thesis, we use the 13-bit Barker pulse due to its higher resolution when compared to the rectangular pulse. This suggests that, the pulse compression helps in improving the range resolution, but not the Doppler resolution, as in Fig. 2.14. The Doppler resolution can be improved by using Spectral estimation techniques which are discussed in the next chapter.

Chapter 3

BACKGROUND IN SPECTRAL ESTIMATION

3.1 Introduction

Digital Signal Processing (DSP) has many operations as its components. To name a few, we have filtering, convolution, windowing, transformations, power calculations, statistics, etc. One of the most important aspects of the DSP is the Spectral Estimation, which comprises of the most or all of the above techniques applied to finite length real world signals. Spectral estimation is a signal processing technique of studying the power distribution of a signal in the frequency domain. In other words, Spectral estimation is all about estimating the distribution of the Power Spectral Density (PSD) of a time-domain signal. Spectral estimation is used for both deterministic and random signals.

Spectrum estimation, in a more general term called the spectral analysis, is a topic of interest and importance in many diverse fields such as in radar, imaging, biomedical diagnostics, speech processing, image processing, seismology, and source location in sensor arrays, astronomy.

Spectrum estimation considers the problem of obtaining and analyzing the spectral content of a time domain signal through a finite set of measurements by utilizing either

parametric or non-parametric methods. Parametric methods are the modern approaches to the spectrum estimation. Here, the signal is assumed to be generated by using a known model, and by knowing the transfer function of the model, the required parameters are estimated. Non-parametric methods are the FFT based classical approaches in which there are no assumptions made regarding the signal model.

There are two ways of performing spectrum analysis. First, it can be performed on the entire signal. Alternatively, a signal can be split into smaller segments, and then spectrum analysis may be applied to these segments.

In this chapter, for a sinusoidal signal, we test the most commonly-used non-parametric spectral estimation techniques such as periodogram, windowed periodogram, correlogram, Welch, Bartlett and Blackman-Tukey and parametric method such as auto-regressive (AR) method. In the AR method, we calculate the filter coefficients using Yule-Walker equations.

The non parametric methods are the classical approaches which formed the foundation of spectral estimation since the 1890's founded by Arthur Schuster [3]. It began with the periodogram, followed by the correlogram and the modified versions of the same. The basis for these techniques is the Fourier transform. Some of the direct methods use Fourier transform directly on the observed data to obtain the PSD estimate, while others use autocorrelation function and then compute its Fourier transform.

The periodograms and correlograms are simple and natural estimates with a reasonable resolution. But unfortunately, in many applications, the bias and variance are

so high, and this makes the PSD obtained to be unreliable. To avoid this problem, the windowed periodogram, Blackman-Tukey method, Bartlett method, and Welch method were developed. There are no assumptions made regarding the signal model, yet the non parametric methods are still used extensively when there is no information available about the behavior of the signal.

Incorporating the information about the signal behavior into the spectral estimation algorithm could lead to a more accurate and higher resolution in frequency. This idea is the basis of parametric spectral estimation methods. With a parametric approach, a spectral estimation model is selected based on experimental results indicating that particular model suits well. In this thesis we use the Yule-Walker method, to solve the AR model coefficients.

3.2 Spectral densities

This section provides a brief background and the importance of the applicability of the energy and power spectral densities. Here, all the signals under consideration are discrete time signals unless otherwise specified.

Consider a time-domain energy signal $x[n]$, with n being the time index. The finite quantity $E(f)$ in Eq. 3.1 is called the Energy Spectral Density (ESD), which is the magnitude squared value of $X(f)$, where $X(f)$ is the discrete-time Fourier transform of the signal $x[n]$.

$$E(f) = |X(f)|^2 = \left| \sum_{n=-\infty}^{\infty} x[n] e^{-j2\pi fn} \right|^2 \tag{3.1}$$

The ESD is a plot of the signal energy as a function of frequency. But, ESD is applicable only for those signals that have finite energy. The properties possessed by the stationary stochastic signals are that these signals are infinite in length and have infinite energy. So, the Fourier transform cannot be applied to infinite length signals directly to analyze the energy, which led to another quantity of interest called the Power Spectral Density (PSD) [3].

The PSD of a discrete time power signal $x[n]$ is the discrete-time Fourier transform of the autocorrelation function, and it is defined by

$$P_X(f) = \sum_{n=-\infty}^{\infty} r_X(n) e^{-j2\pi fn}. \tag{3.2}$$

Here, $r_X(k)$ is the autocorrelation function of $x[n]$ with a lag *k,* is given by

$$r_X(k) = E[x(n)x(n-k)]. \tag{3.3}$$

The autocorrelation function is the measure of self similarity of a signal. It enables us to determine the degree of identicalness of a signal. Mathematically, it is the expectation of the product of a signal with its delayed version as given in Eq. 3.3 [3].

A deterministic signal has its power concentrated in narrow bands of frequencies, while a purely random signal has its power spread in a wide range of frequencies [3].

In practice, we deal with finite length signals that are obtained from one or few realizations of stochastic processes. Therefore, we compute the Discrete Fourier Transform (DFT) of such signals. The DFT has a finite number of points on the spectrum. The expression to compute the PSD estimate of $x(n)$ is given by [3]

$$X(m) = \sum_{n=0}^{N-1} x(n) \exp(-j\frac{2\pi}{N} mn), \tag{3.4}$$

where m=0,1,…,N-1, are the points on the spectrum.

In Fig 3.1, the power is concentrated in a narrow frequency band around 1 Hz which is the actual frequency of the sinusoid under consideration.

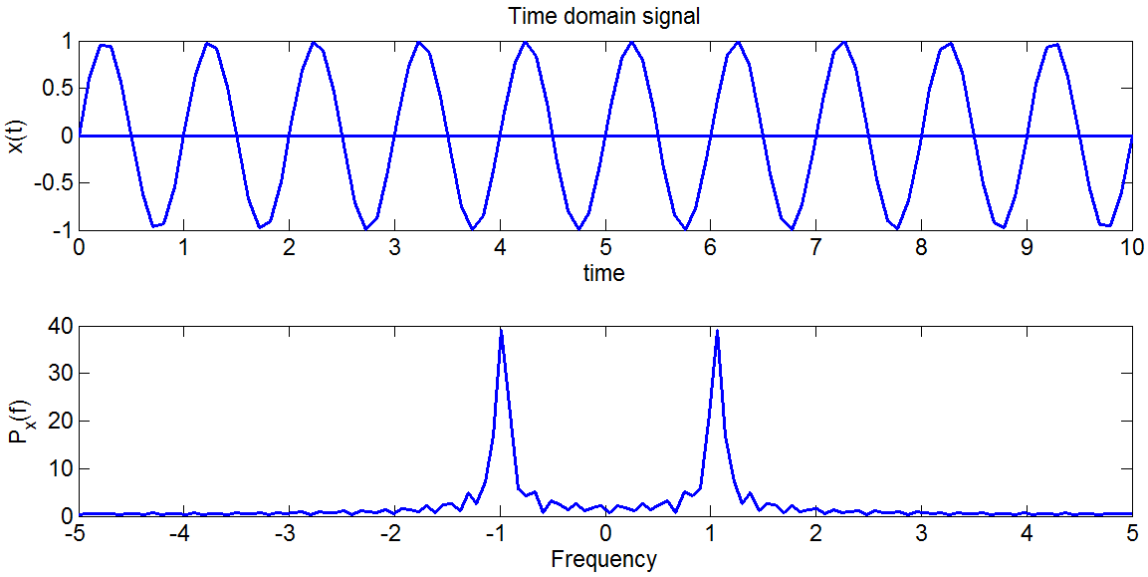


Fig. 3.1: 1 Hz sinusoid and its approximate spectrum

The ability of the PSD in analyzing a signal buried in a white noise is demonstrated as in Fig. 3.2. In Fig. 3.2a, the actual sinusoid x (t) and its white noise corrupted version y (t) are shown. In Fig. 3.2b, the autocorrelation function of y (t) and its PSD are shown. Even though a considerable level of noise is present in the signal, by obtaining the PSD, we can notice the actual frequency component of the signal is 1 Hz. The ripples in the spectrum of Fig. 3.2b are due to the white noise.

The autocorrelation function and PSD of the corrupted version of the sinusoid in Fig. 3.2 is shown in Fig. 3.3.

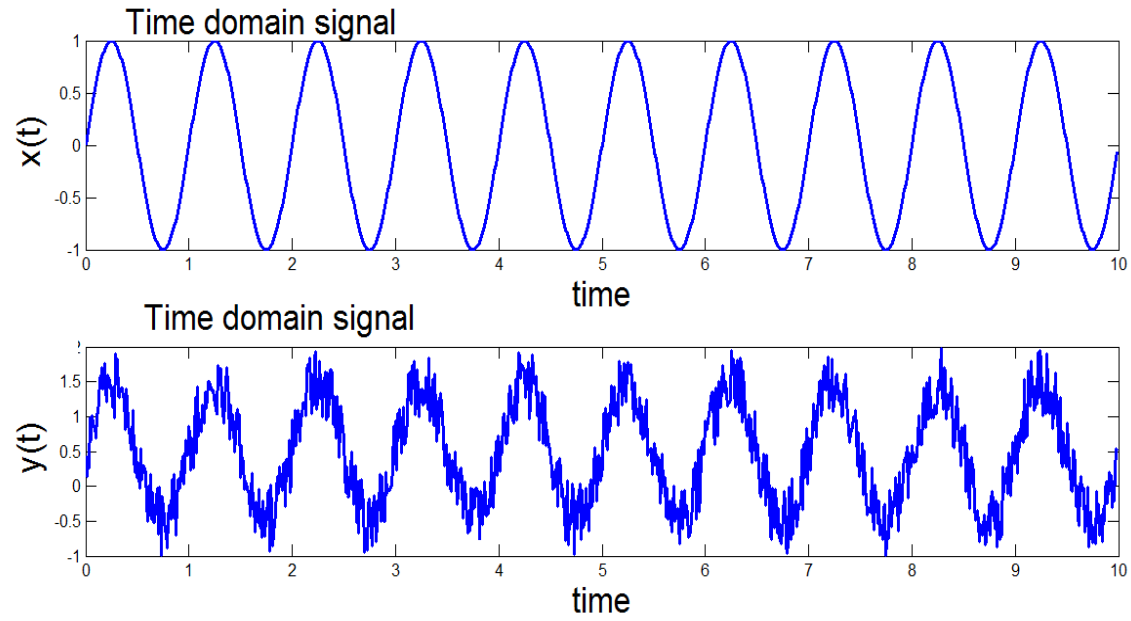


Fig. 3.2: 1 Hertz sinusoid and its band-limited white Gaussian noise corrupted version

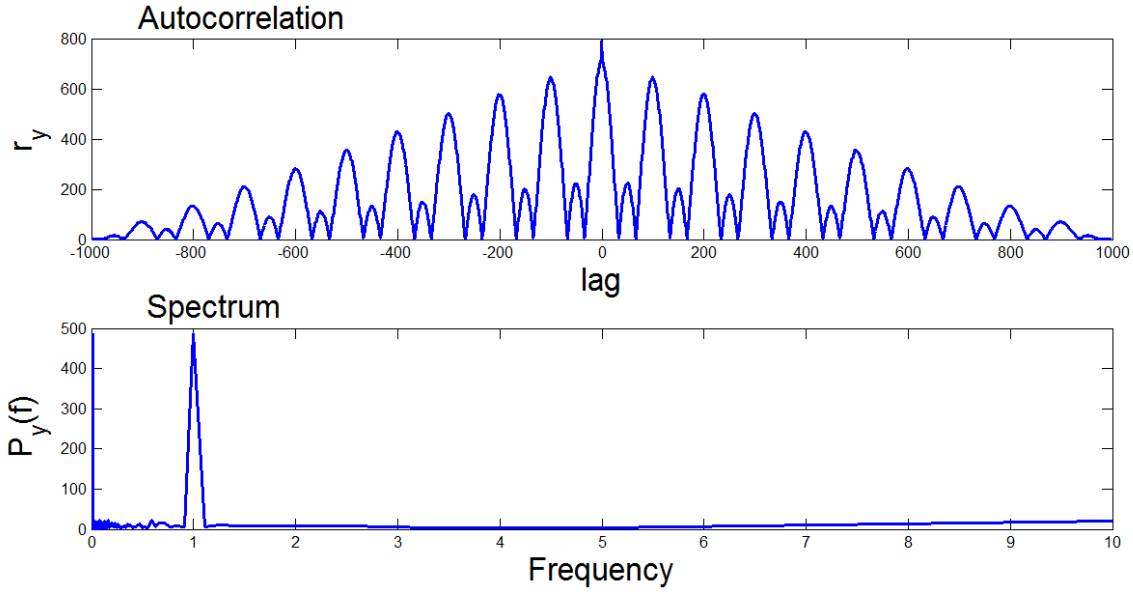


Fig. 3.3: Autocorrelation function and PSD of the corrupted version of the sinusoid in Fig. 3.2

3.3 Estimator

The purpose of estimation is to obtain the best possible value of the parameter to be estimated from the given set of observations. The filter or algorithm which does the estimation is called the estimator. The input to the estimator is a set of noisy, or incomplete observations, and by using a probabilistic or a model based signal approach, the unknown parameter is estimated. The accuracy of the estimator mainly depends on the length of the observations available and the model selected for estimating.

The important qualities of an estimator are bias, variance and consistency. Consider a scalar quantity a and its estimate $\hat{a}$. Then, the bias is defined as the difference of the expected value of the estimate $E[\hat{a}]$ and the true value a. The variance of an estimator is defined as the expected value of the squared value of the difference of the estimate and its mean value [10].

$$Bias = E[\hat{a}] - a \tag{3.5}$$

$$Variance = E[(\hat{a} - E[\hat{a}])^2] \tag{3.6}$$

An ideal estimator should have zero bias and zero variance. But in practice, a good estimator will have a small bias and small variance [12], [3]. Consistency means that, as the number of observations *N* grows larger and approaches infinity, the estimate will tend to equal the true value.

3.4 Periodogram

Arthur Schuster is the inventor of the periodogram [3]. He used this technique to find the hidden frequencies or periodicities in a signal, hence the name periodogram. The presence of a dominant sinusoidal frequency, f, in a time signal results in a sharp peak in the frequency domain centered at f. This fact is utilized in computing the periodogram estimate. Mathematically, the PSD estimate of a discrete signal $x[n]$ of length N using the periodogram is given by Eq. 3.6 [3] and Fig. 3.4.

$$\hat{P}_{periodogram}(m) = \frac{1}{N}\left|\sum_{n=0}^{N-1} x[n]\exp(-j\frac{2\pi}{N}mn)\right|^2 \tag{3.7}$$

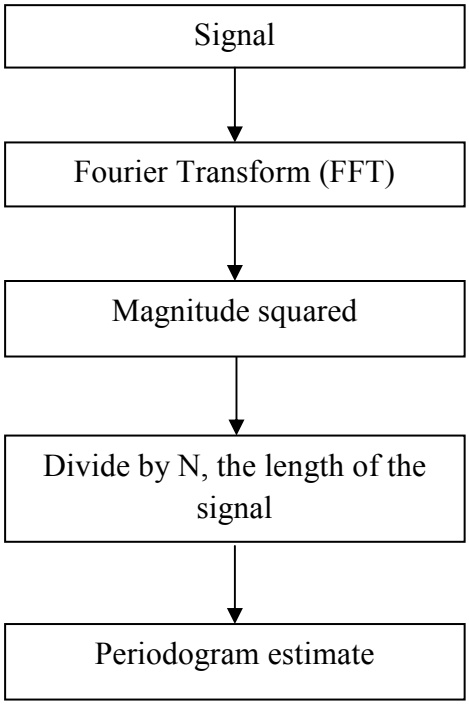


Fig. 3.4: Algorithm for the periodogram

To study the various estimators in this chapter, we make use of a band-limited white Gaussian noise (Appendix A) contaminated signal of the form $y = x + w$.

Here, $x = \cos(2\pi f_1 n) + \cos(2\pi f_2 n) + \cos(2\pi f_3 n) + \cos(2\pi f_4 n)$ is the true signal containing four sinusoids of relative frequencies: $f_1 = 0.05,\ f_2 = 0.3,\ f_3 = 0.31,\ f_4 = 0.4,$ and w is the white Gaussian noise (Appendix A). Fig. 3.5 demonstrates the effect of different DFT lengths *M*, on the periodogram for 10 simulations with independent sets of random data. For efficient and fast computation of the DFT, the original sequence is zero padded to a length with the power of 2, and a fast algorithm [15] called the Fast Fourier Transform (FFT) is used. The signal of length *N* is zero-padded to form a sequence of length *M*, and then an *M*-point DFT is applied to the resulting sequence for computing the periodogram. This process of padding the sequence with zeros replaces *N* with *M* in Eq. 3.6, and in the subsequent techniques to follow. But however, the mathematical expressions for spectral estimation techniques are represented as an *N*-point DFT rather than an *M*-point DFT.

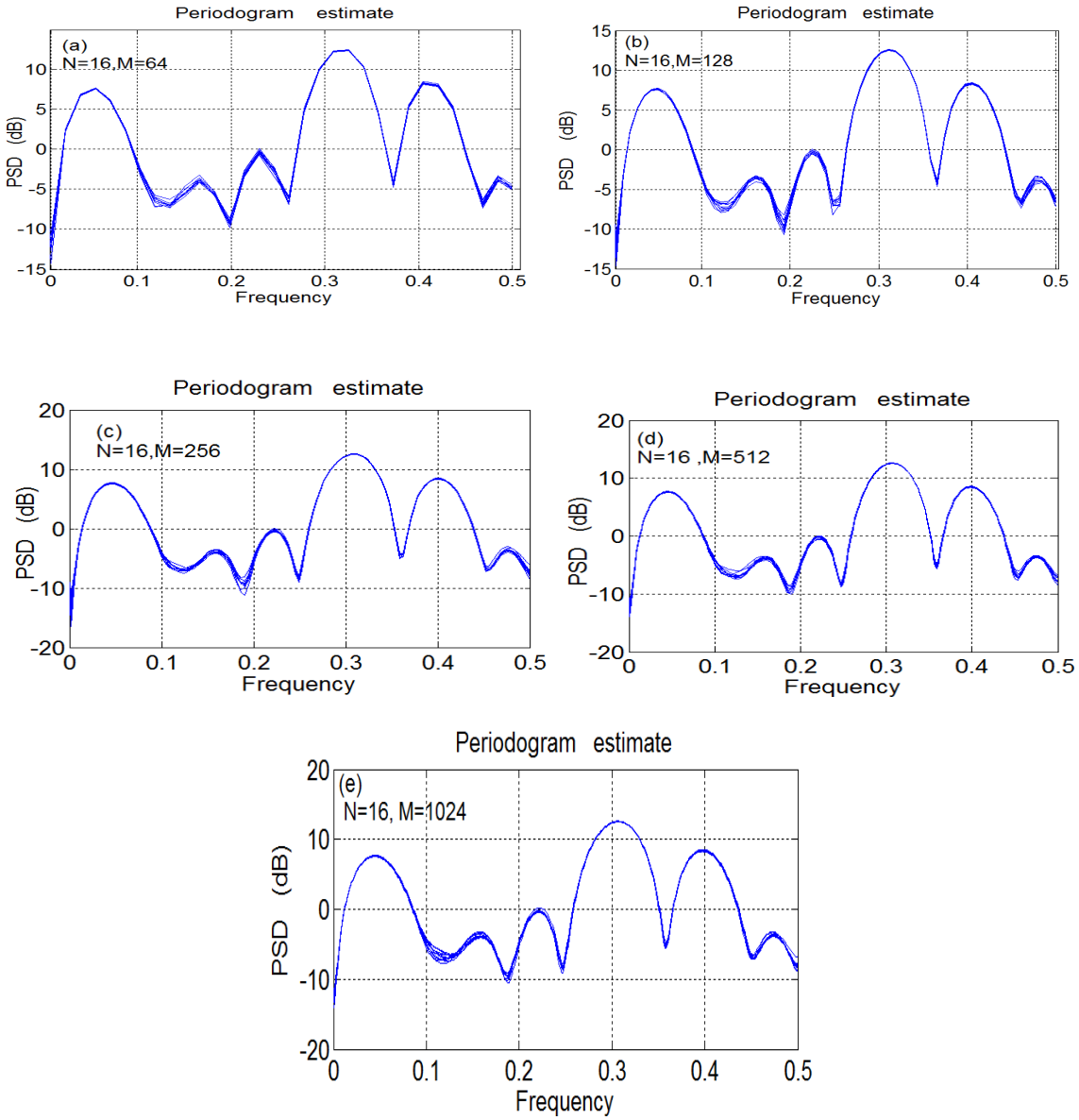


Figure 3.5: Periodogram PSD estimate for *N*=16 and different *M* point DFTs as indicated in respective figures

From Fig. 3.5, it is evident that, larger the number of points in the DFT, the smoother is the spectrum. Therefore, we use *M*=1024 point DFT in the subsequent figures. It is important to note that the large value of *M* does not provide better resolution,

but rather gives a smoother curve. In Fig. 3.5, the frequencies $f_2 = 0.3$ and $f_3 = 0.31$ are not resolved and appear as a single lobe due to the small length of the sequence.

We shall now investigate the effect of larger sequence length on the process of periodogram estimation as in Fig. 3.6.

By visual inspection, we notice in Fig. 3.6, the peaks in periodogram estimate become sharper and get resolved as the length of the data sequence increases. In Fig. 3.6a –Fig. 3.6c, the frequencies $f_2 = 0.3$ and $f_3 = 0.31$ are not resolved, but in Fig. 3.8d these frequencies are partially resolved indicating that there are two frequency components. In Fig. 3.6e- Fig. 3.6g, the peaks are completely resolved. The conclusion of Fig. 3.6 is that larger observation length yields better resolution which is given by $0.89/N$ [13].

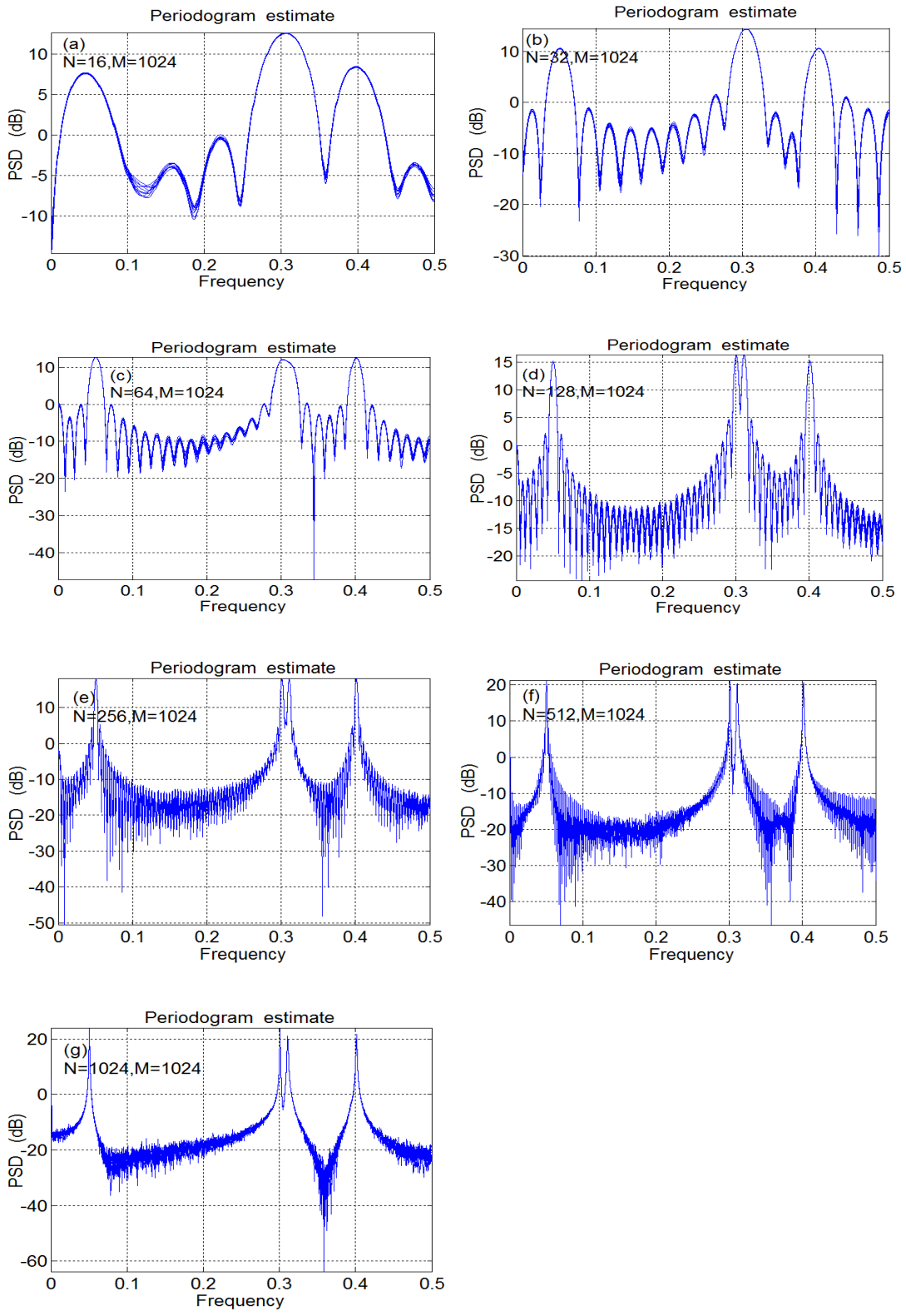


Fig. 3.6: Effects of the signal length *N* on the periodogram spectral estimate for M=1024 point DFT as indicated in the respective Figures.

The conventional periodogram is not consistent, since its variance does not converge to zero [2]. This is due to the fact that there is an abrupt truncation of the sequence. To mitigate this effect, windows are used to gradually truncate the sequence in time domain and reduce the effect of sidelobes in the frequency domain. Convergence of the variance can be improved by applying windows, described in Appendix B, to the periodogram as in Eq. 3.7. The term $w[n]$ is the window function, and U is its power. The algorithm for the windowed periodogram is almost similar to that of the natural periodogram shown in Fig. 3.7.

$$\hat{P}_{windowed}(m) = \frac{1}{NU}\left|\sum_{n=0}^{N-1} w[n]x[n]\exp(-j\frac{2\pi}{N}mn)\right|^2 \tag{3.8}$$

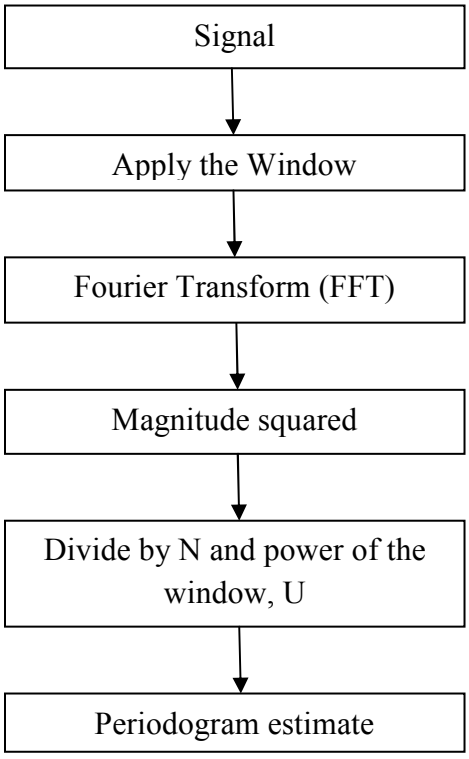


Fig. 3.7: Algorithm for windowed periodogram

3.5 Correlogram

The correlogram is another non-parametric technique well recognized as the periodogram. Since the PSD and the autocorrelation function are Fourier transform pairs, they contain exactly the same information. The correlogram uses the estimate of the autocorrelation function and then its Fourier transform is computed to estimate the PSD.

The correlogram in its mathematical form is described by [2].

$$\widehat{P}_{correlogram}(m) = \sum_{-(N-1)}^{(N-1)} \hat{r}(k) \exp(-j\frac{2\pi}{N} mn), \tag{3.9}$$

where k is the lag of $\hat{r}(k)$, the estimate of the autocorrelation function.

$\hat{r}(k)$ is calculated in two ways,

Biased Estimate: $$\hat{r}(k) = \frac{1}{N}\sum_{i=k}^{N-1} x(i)x^*(i-k) \quad , k \geq 0, \tag{3.10}$$

Unbiased estimate: $$\hat{r}(k) = \frac{1}{N-k}\sum_{i=k}^{N-1} x(i)x^*(i-k) \quad , \; k \geq 0, \tag{3.11}$$

For both cases, $$\hat{r}(k) = \hat{r}^*(-k) \quad , k < 0. \tag{3.12}$$

The autocorrelation estimate given by the biased estimate in Eq. 3.9 is used more often in the computation of the correlogram [2]. The algorithm for computing the PSD estimate using the correlogram is given in Fig. 3.8, followed by analysis of the correlogram for varying signal lengths as in Fig. 3.9.

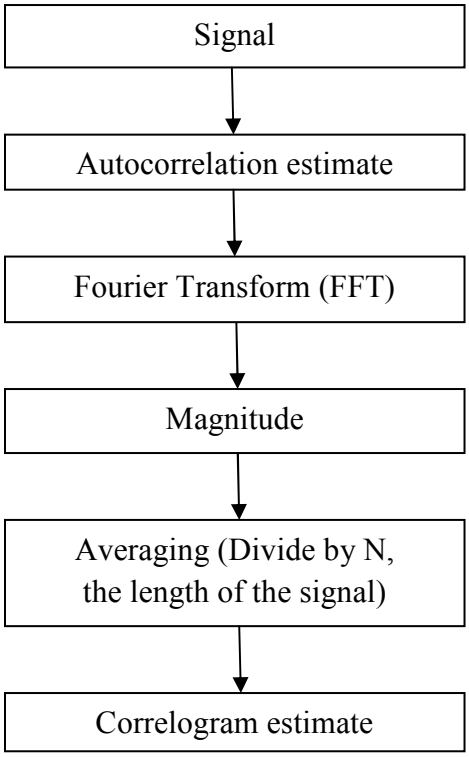


Fig. 3.8: Algorithm for correlogram

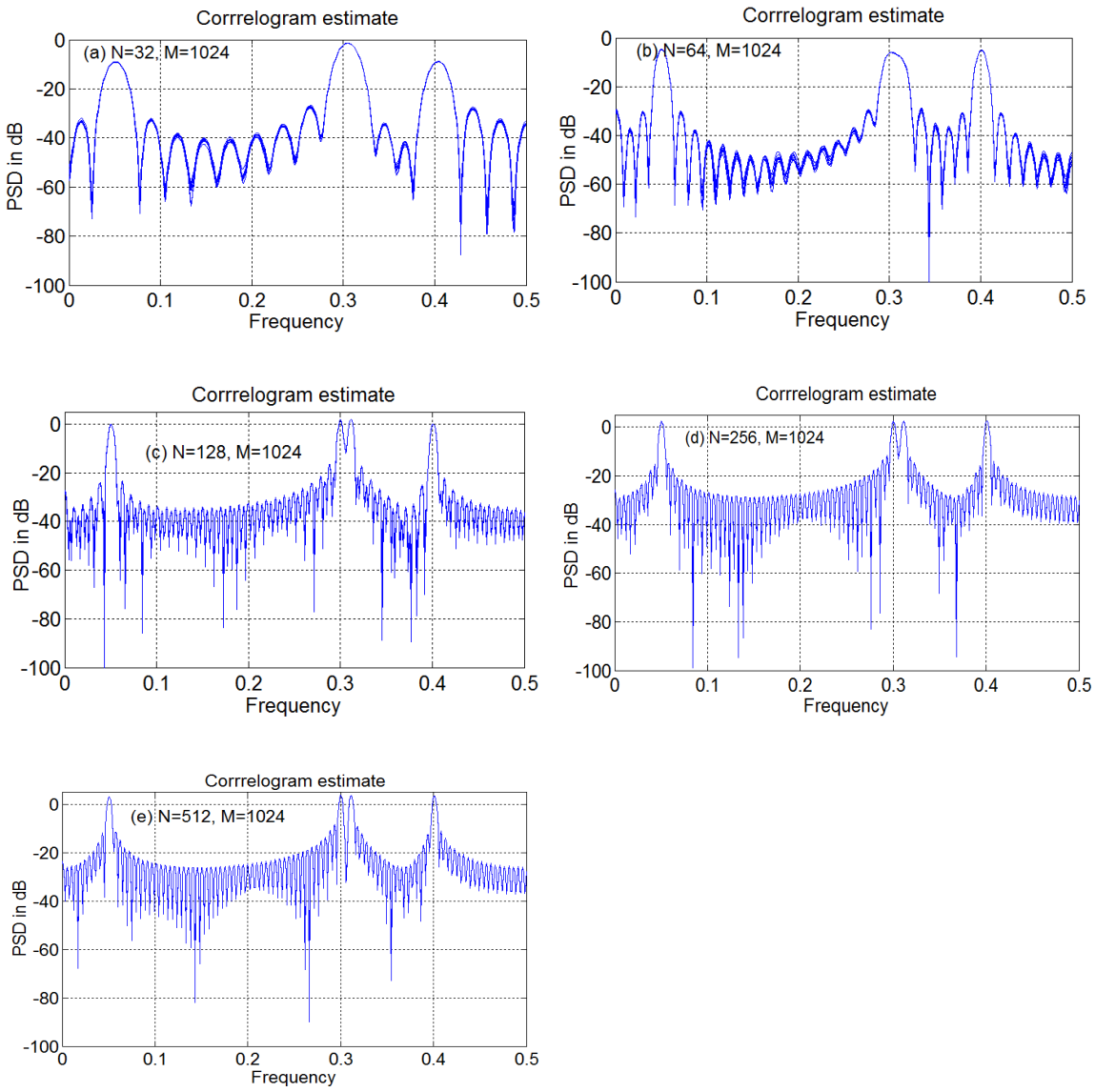


Fig. 3.9: Effect of signal length *N* for Correlogram estimate, as indicated in the respective figures.

In the sections to follow, we present the modified periodogram and correlogram methods that adopt different schemes of averaging.

3.6 Bartlett periodogram

This is a modified periodogram method in which a different averaging scheme is used. The signal is split into segments, and the periodogram is computed for each individual segment. The individual periodograms are then averaged. Mathematically, the Bartlett periodogram is given as in Eq. 3.7, and its algorithm is shown in Fig. 3.10.

$$\hat{P}_{Bartlett}(m) = \frac{1}{K}\sum_{k=0}^{K-1}\hat{P}_{segment}^{(k)}(m) \tag{3.13}$$

In Eq. 3.7, K is the number of segments, and

$$\hat{P}_{segment}^{(k)}(m) = \frac{1}{L}\left|\sum_{i=0}^{L-1}x_k[i]\exp(-j\frac{2\pi}{N}mi)\right|^2 \tag{3.14}$$

is the periodogram estimate for the k-th segment of length L. To achieve an equal number of segments, the last segment is either zero padded to length L or discarded.

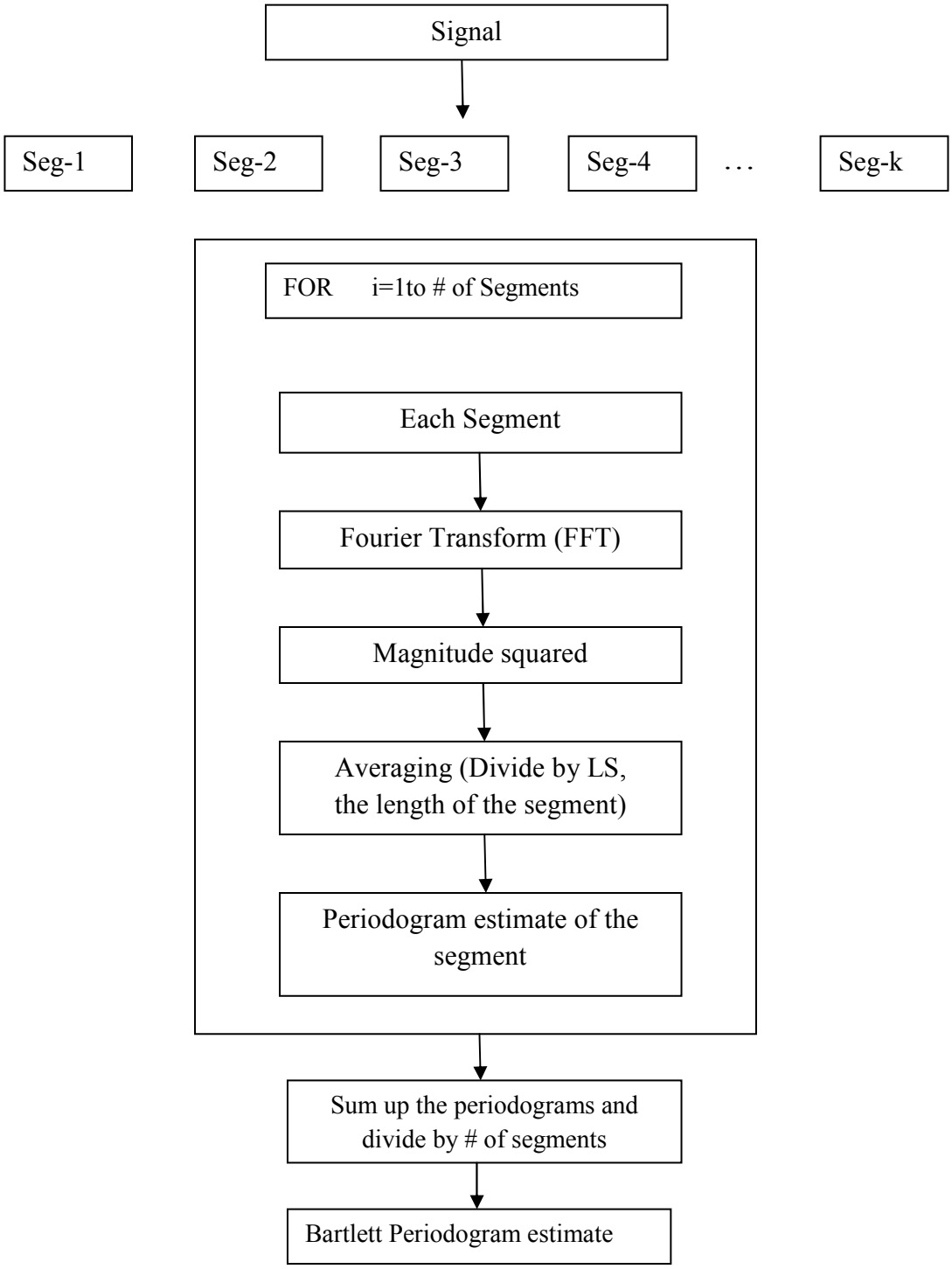


Fig. 3.10: Algorithm for Bartlett periodogram

Fig. 3.11 shows the Bartlett periodogram estimate for the length of the sequences N =64 and N=512. The PSD estimate is better for N =64 than N=512. This shows that the Bartlett periodogram is well suited for short sequences. Moreover the resolution is poor since the frequencies 0.3 and 0.31 are not resolved.

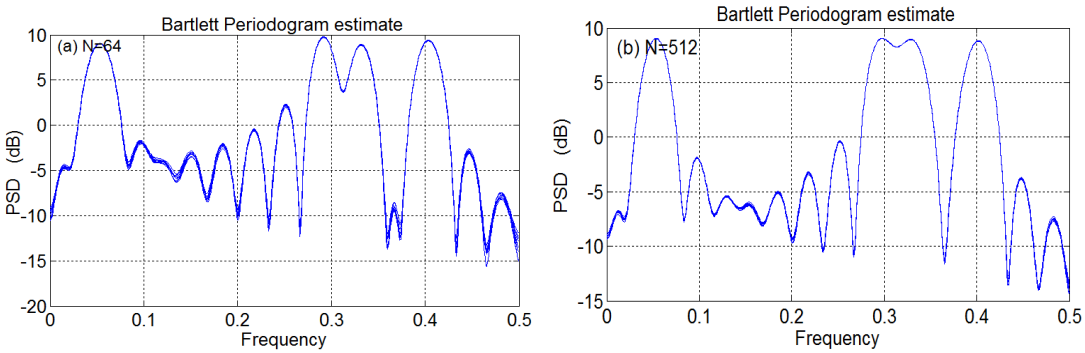


Fig. 3.11: Bartlett periodogram estimate for (a) N=64, (b) N=512

3.7 Blackman Tukey correlogram

This is a modified correlogram method. Here the PSD estimate is obtained by computing the Fourier transform of the windowed autocorrelation estimate. The window is used in order to truncate the autocorrelation estimate gradually, and to reduce the sidelobe levels at either ends of the autocorrelation estimate. Bartlett window is the commonly used window in computing the Blackman Tukey estimate [2]. Mathematically, the Blackman-Tukey is shown in Eq. 3.9, and its algorithm follows it in Fig. 3.12.

$$\widehat{P}_{BT}(m) = \sum_{k=-(N-1)}^{(N-1)} w(k)\hat{r}(k)\exp(-j\frac{2\pi}{N}mn) \qquad (3.15)$$

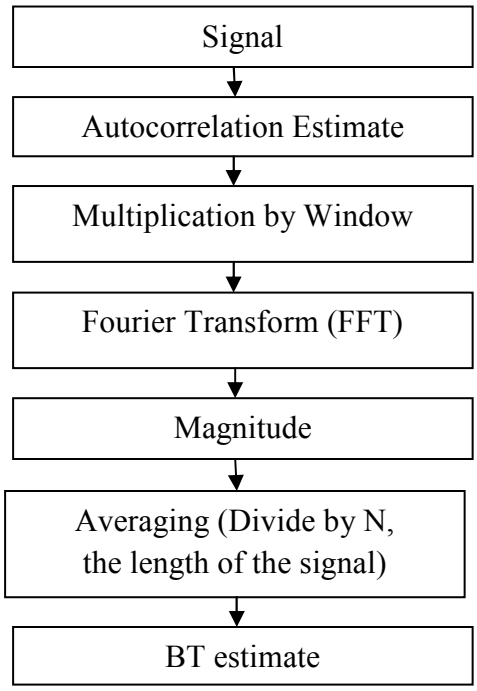


Fig. 3.12: Algorithm for Blackman-Tukey method

Fig. 3.13 shows the Blackman-Tukey periodogram estimate for N=32 and N=512. We see that even though N=512 has sharper peaks than the N=32, it does not resolve the peaks completely.

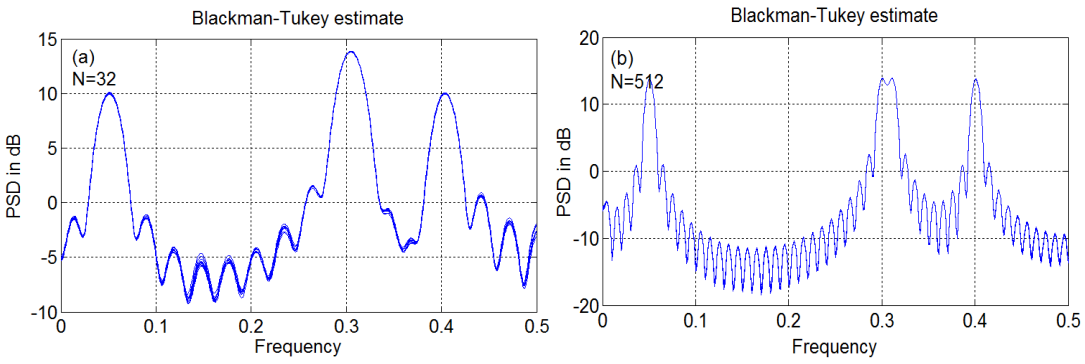


Fig. 3.13: Blackman-Tukey periodogram estimate for (a) N=32, (b) N=512

3.8 Welch periodogram

The Bartlett periodogram was modified to produce a better estimate. Now we consider the Welch periodogram named after P. D. Welch [2]. In this method, the original sequence is split into length L subsequences or segments. The adjacent segments are allowed to overlap and a window is applied to each segment. The periodogram is computed for each of the windowed segments and then averaged to obtain the Welch periodogram estimate. The overlap factor is D. When $D=L$, there is no overlap which is nothing but the Bartlett periodogram. But $D=L/2$ is the 50% overlap used very often since it yields better estimates [1], [2]. The Welch periodogram estimate is given by the equation

$$\hat{P}_{Welch}(m) = \frac{1}{N_{seg}} \sum_{k=0}^{N_{seg}-1} \hat{P}_{segment}^{(k)}(m). \tag{3.16}$$

$\hat{P}_{segment}^{(k)}(f)$ is the periodogram for the kth windowed segment defined [2] as

$$\hat{P}_{segment}^{(k)}(m) = \frac{1}{WL} \left| \sum_{i=0}^{L-1} w(i) x_k(i) \exp(-j \frac{2\pi}{N} mi) \right|^2, \tag{3.17}$$

where, W is the power of the window function given by

$$W = \frac{1}{L} \sum_{i=0}^{L-1} w^2(i). \tag{3.18}$$

The finite length sequences exhibit the property of sidelobe leakage in the spectrum. To overcome this problem, windows are used to reduce the spectral leakage and provide smoothing caused due the discontinuities while truncating. The algorithm for computing Welch periodogram is given in Fig. 3.15 and the Welch periodogram estimate for N=32 and N=512 are demonstrated in Fig. 3.14.

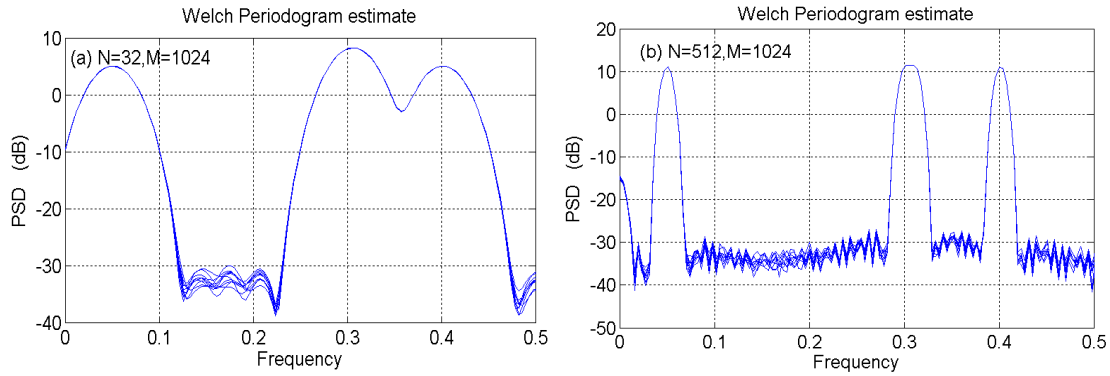


Fig. 3.14: Welch periodogram estimate for (a) N=32, (b) N=512

From Fig. 3.14, conclusion can be made that the Welch periodogram performs better for large sequences unlike the Bartlett periodogram, due to the overlapping of segments.

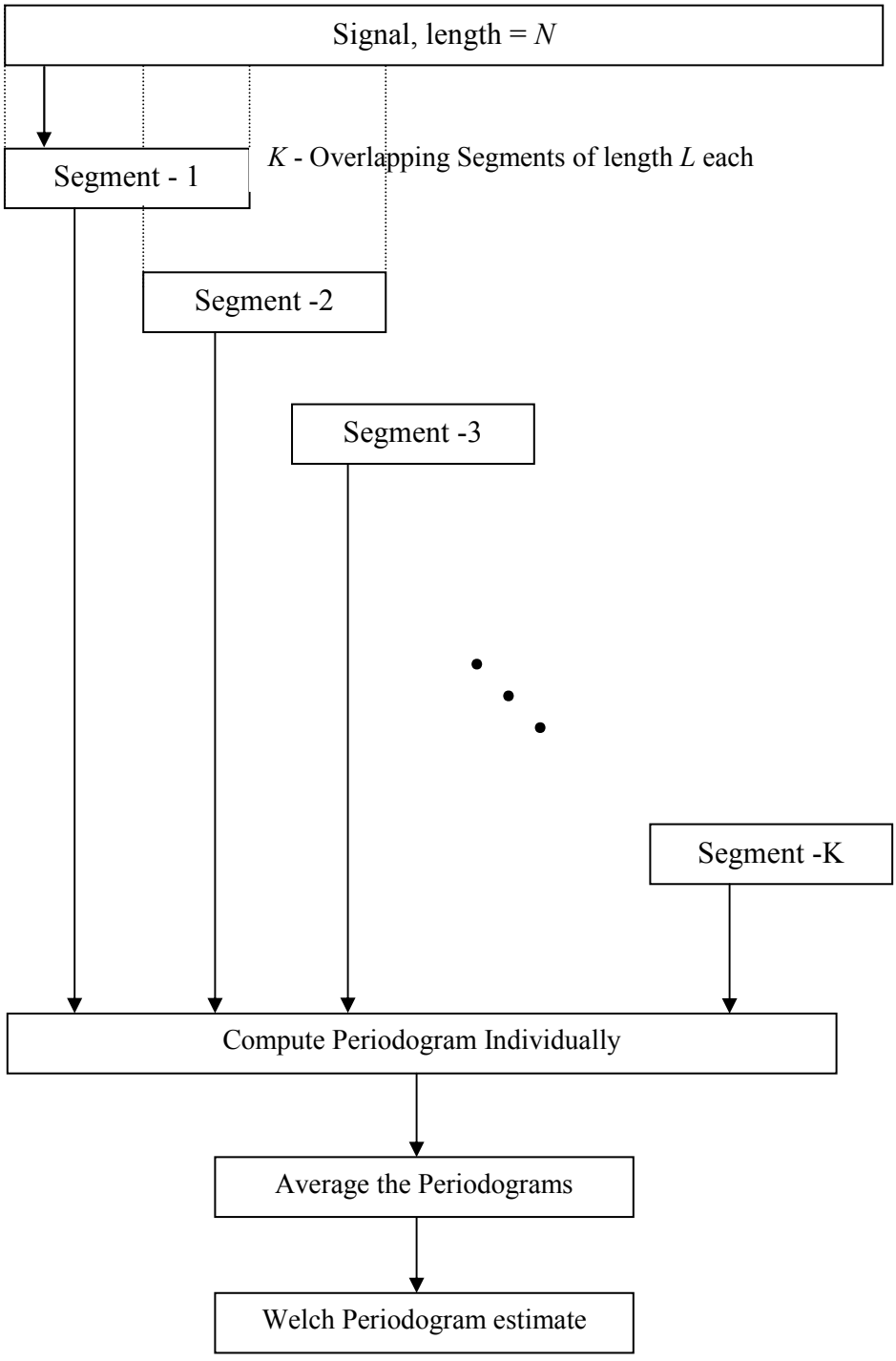


Fig. 3.15: Algorithm for Welch periodogram

3.9 Auto-regressive method

Parametric methods use an alternative approach to the nonparametric methods by modeling the signal as a transfer function with the input as white Gaussian noise process. The commonly used parametric methods are Auto Regressive (AR), Moving Average (MA), and Autoregressive Moving Average (ARMA) methods.

AR model is an IIR filter used to predict the current value of a signal based on its previous value. Mathematically, the AR model can be written as

$$x[n] = \sum_{i=1}^{p} a[i]x[n-i] \ , \tag{3.19}$$

where $x[n]$ is the signal under investigation, p is the filter order which is usually less than the signal length *N*, and $a[i]$ are the AR coefficients. Eq. 3.13 means that the current value of the signal can be predicted by a linear sum of previous values weighted by AR coefficients. Given $x[n]$, the AR coefficients $a[i]$ can be calculated by using many methods, in our case the Yule-Walker method.

The Yule-Walker (YW) method is one of the techniques to compute the AR coefficients. For a finite length sequence, it is possible to compute the autocorrelation for a lag less than the signal length *N*. As a result, we can compute the estimates of the AR coefficients using the YW equations as given [2] by

$$\hat{r}[k] = \sum_{l=1}^{p} \hat{a}[l]\hat{r}[k-l] \qquad \text{for } k \geq 1, \tag{3.20}$$

$$\hat{r}[k] = \sum_{l=1}^{p} \hat{a}[l]\hat{r}[-l] + \sigma^2 \quad \text{for } k = 0, \tag{3.21}$$

and

$$\hat{r}[k] = \hat{r}^{*}[-k] \quad \text{for } k < 0. \tag{3.22}$$

The estimates of the filter coefficients order p are then solved by the YW set of equations [2] as

$$\begin{bmatrix} 1 \\ \hat{a}[1] \\ \vdots \\ \hat{a}[p] \end{bmatrix} = \begin{bmatrix} \hat{r}[0] & \hat{r}[-1] & \dots & \hat{r}[-p] \\ \hat{r}[1] & \hat{r}[0] & \dots & \hat{r}[-p+1] \\ \vdots & \vdots & \ddots & \vdots \\ \hat{r}[p] & \hat{r}[p-1] & \cdots & \hat{r}[0] \end{bmatrix}^{-1} \begin{bmatrix} \sigma^2 \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{3.23}$$

For a non-finite length sequence, the true autocorrelation function can be computed and the AR estimates are replaced by the true coefficients $a[n]$.

The PSD estimate of the AR process in Eq. 3.13 is given [3] by

$$\hat{P}_{AR}(m) = \sigma^2 \left| \frac{1}{1 + \sum_{n=1}^{p} \hat{a}[n] \exp(-j\frac{2\pi}{N} mn)} \right|^2, \tag{3.24}$$

where, $\hat{a}[n]$ are the estimated filter coefficients, σ^2 and is the noise variance.

Fig. 3.16 shows the AR estimate using the YW method of model order p=10. Note that the YW method produces sharp peaks at the frequencies of the signal. But, the YW does not resolve the very close frequencies 0.3 and 0.31.

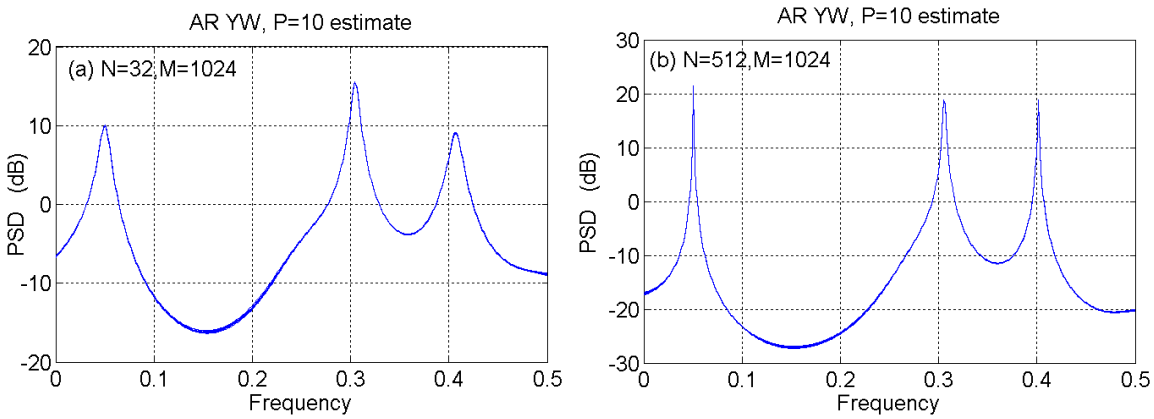


Fig. 3.16: AR YW, P=10 estimate for (a) N=32, (b) N=512

In this chapter, we have provided the background of the spectral estimation techniques used. The following chapter will present the results of the bias and variance of the estimators prior to the results of the Range-Doppler mapping.

Chapter 4

SIMULATION RESULTS

4.1 Introduction

In Chapter 2, we discussed about the formation of range-Doppler map and how spectral estimation techniques are applied to the same. In this chapter, the formation of range-Doppler map, obtained from the usage of spectral estimation techniques as in Fig. 2.11, is presented. The results shown in this thesis are obtained from a series of computer algorithms in MATLAB. Due to the powerful capabilities of performing complex operations and graphical plots, MATLAB was the software package chosen for presenting the results.

4.2 Simulation parameters

In this thesis, we assume an airplane target at a range R is moving towards the radar with a radial velocity v_r. Hence, the Doppler shift f_d is a time varying function of radial velocity. But however, if we consider only the data observed for 1 CPI, then Doppler shift can be treated as a constant. This assumption can be justified with an example as follows. For the Radar parameters such as 1 $PRI = 20\,\mu s$ and $N_P = 32$, the dwell time which is the length of 1 CPI is given by 0.64 ms. Even a supersonic jet cannot travel more than a meter in this time. Hence, for a given CPI, the assumed target will have a

fixed Doppler frequency f_d and a fixed range R. Table 4.1 shows the parameters used for simulation of an aircraft target in this thesis.

Table 4.1: Radar parameters for the simulated data

PARAMETER	NOTATION	VALUE
RF CARRIER FREQUENCY	f_0	20 GHz
PULSE	$P_F(t)$	13-BIT BARKER CODE (Fig. 2.10)
PULSE REPETITION FREQUENCY	*PRF*	50 kHz
PULSE REPETITION INTERVAL	*PRI*	20 μs
NUMBER OF PULSES IN PULSE BURST	N_P	32
COHERENT PROCESSING INTERVAL	*CPI*	32x20 μs = 0.64 *ms*
PULSE WIDTH	τ	1 μs
SUB-PULSE WIDTH	τ_B	1/13 μs
TARGET VELOCITY	v_r	250 *m/s*
DOPPLER SHIFT	f_d	31.66 kHz
TIME DELAY	T_d	33.33 μs
SNR	SNR	10 dB
RANGE TO THE TARGET	*R*	5000 *m*

Fig. 4.1a shows the 13-bit Barker pulse, which is the unmodulated transmitted wave, and Fig. 4.1b is the received wave with a time delay T_d. Fig. 4.1c is the closer view of Fig. 4.1b that demonstrates the amplitude level changed due to the complex sinusoid in Eq.2.4. The amplitudes in Fig. 4.1 are chosen arbitrarily for demonstration purposes. The typical voltage levels of the transmitted signal and the received signals are usually in kilovolts and microvolts, respectively [1].

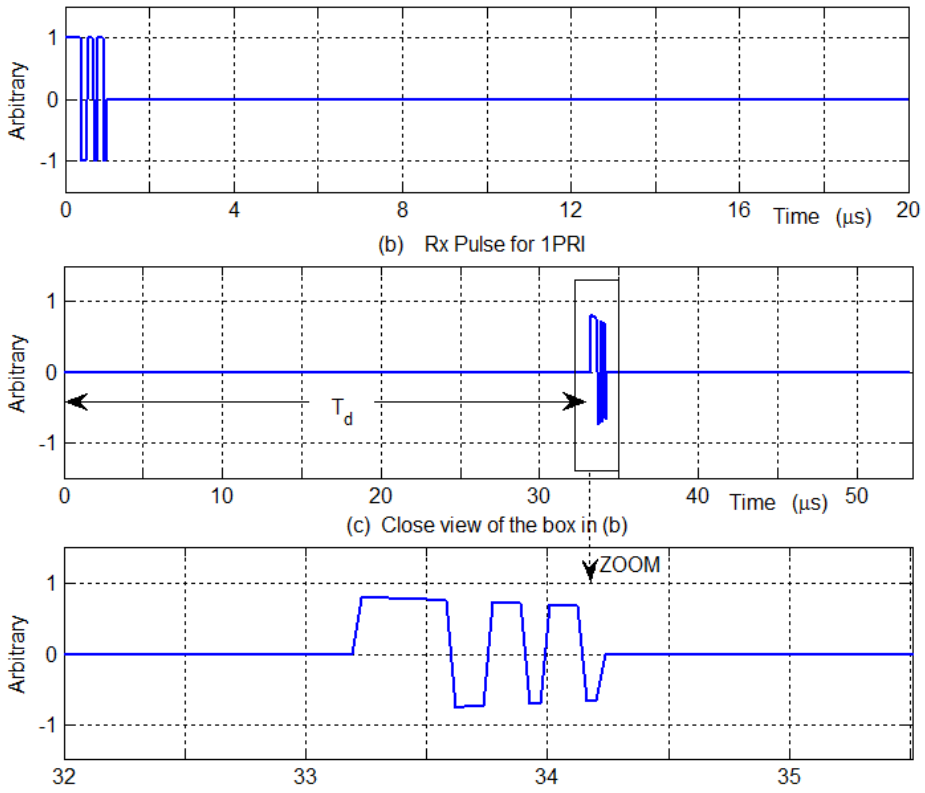


Fig. 4.1: (a) 13-Bit code for 1 PRI, (b) Real part of received signal modulated by frequency f_d with a delayed by time T_d , (c) Close up view of Fig. 4.1 (b)

In the following sections, based on the procedures mentioned in Chapter 2, and by using the parameters in Table 4.1, range-Doppler maps are simulated using the spectral estimation techniques mentioned in Chapter 3.

4.3 Periodogram estimators

The periodogram estimators used in thesis are the un-windowed periodogram, windowed periodogram, Bartlett periodogram, and the Welch periodogram as in sections 3.4, 3.5, 3.7, and 3.9 respectively.

4.3.1 Bias and variance of periodogram estimators

The bias and the variance of an estimator are computed as per Eq. 3.4 and Eq. 3.5, respectively.

In Fig. 4.2, the PSD estimate becomes more accurate as the length N increases. The frequencies 0.3 and 0.31 are well resolved, and the peaks become sharper as in Fig. 4.3b. The sidelobes are also suppressed to 40dB from the maximum peak.

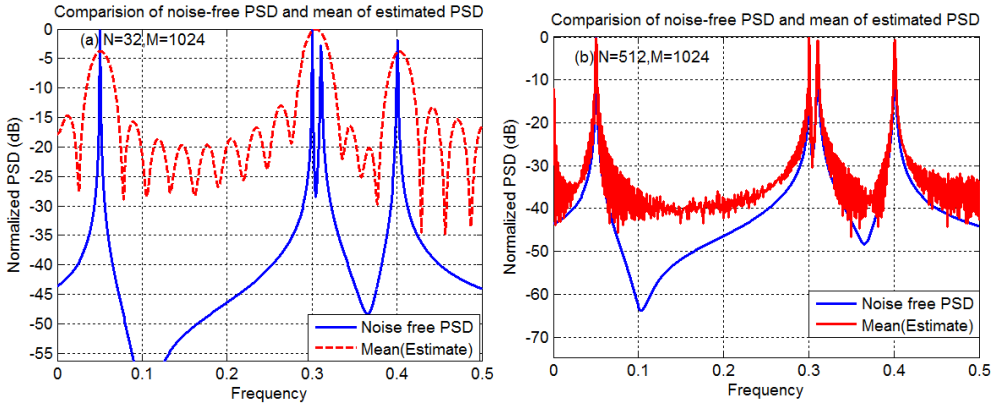


Fig. 4.2: Comparison of noise-free PSD and periodogram for (a) N=32, N=512

We now analyze the bias and variance of the periodogram, as in Fig. 4.3. From Fig. 4.3, we can see that the bias of the periodogram is large. On the other hand, the variance reduces from N=16 to N=512. It is evident from Fig. 4.3, that as N increases the bias also increases and variance decreases. This indicates that the periodogram is a

consistent, but a biased estimate. The variance is up scaled to 1000 times for visual purposes.

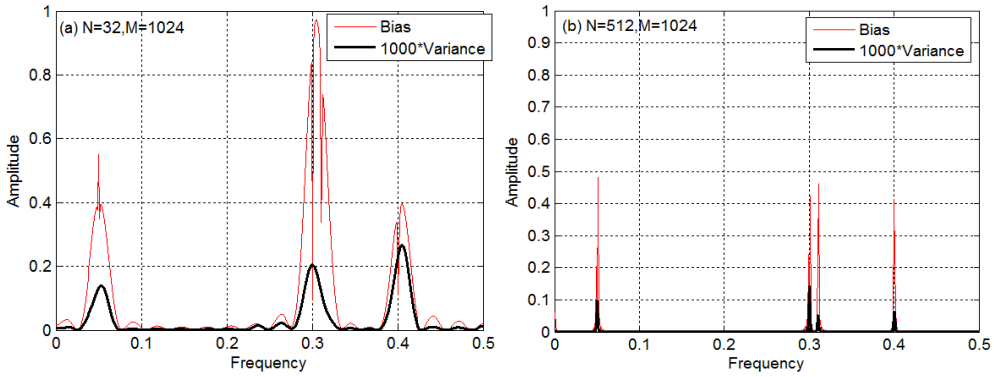


Fig. 4.3: Bias and variance of the periodogram estimate for (a) N=16, (b) N=512

Fig. 4.4 demonstrates the effects of using windowed periodogram on the PSD estimate for the signal length N=512, and FFT points M=1024. The peaks are sharp and sidelobe levels have significantly reduced by using the windows in accordance with the discussions in Chapter 3.

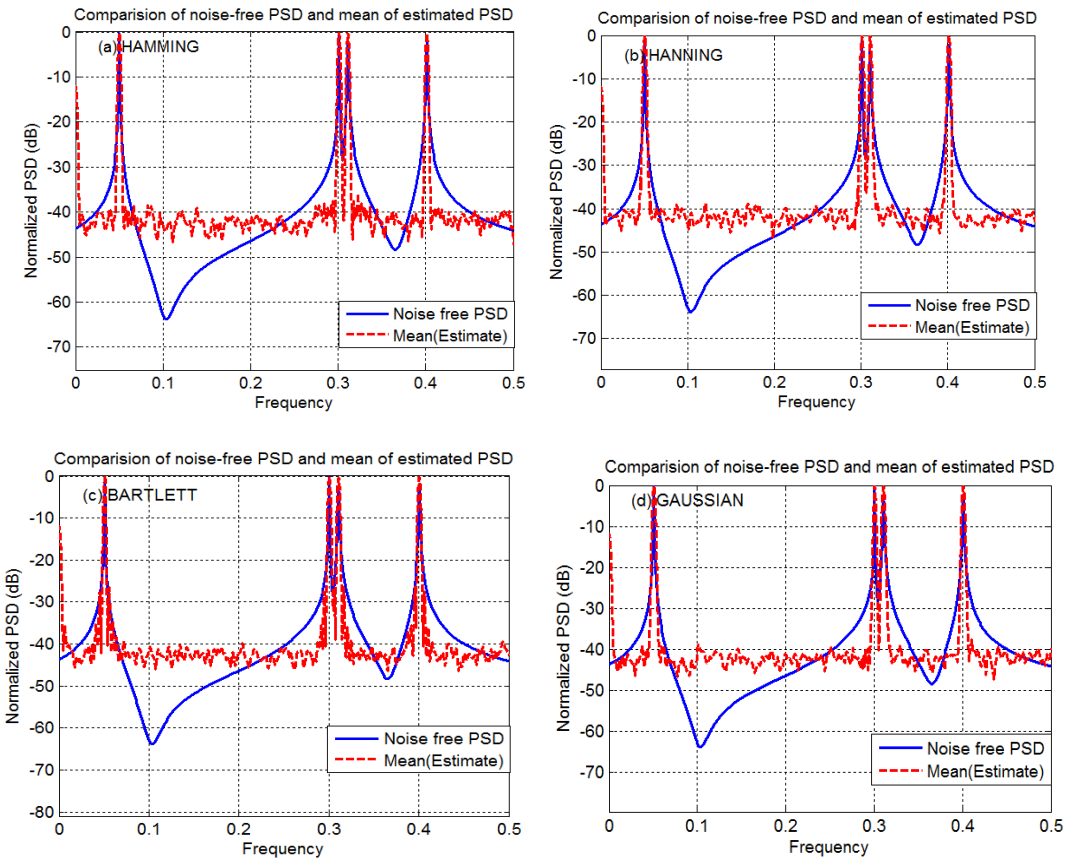


Fig. 4.4: Comparison of noise free PSD and windowed periodogram using (a) Hamming window, (b) Hanning window, (c) Bartlett window, (d) Gaussian window

As discussed in Chapter 3, Fig. 4.5 proves that the usage of windows reduces the problem of variance.

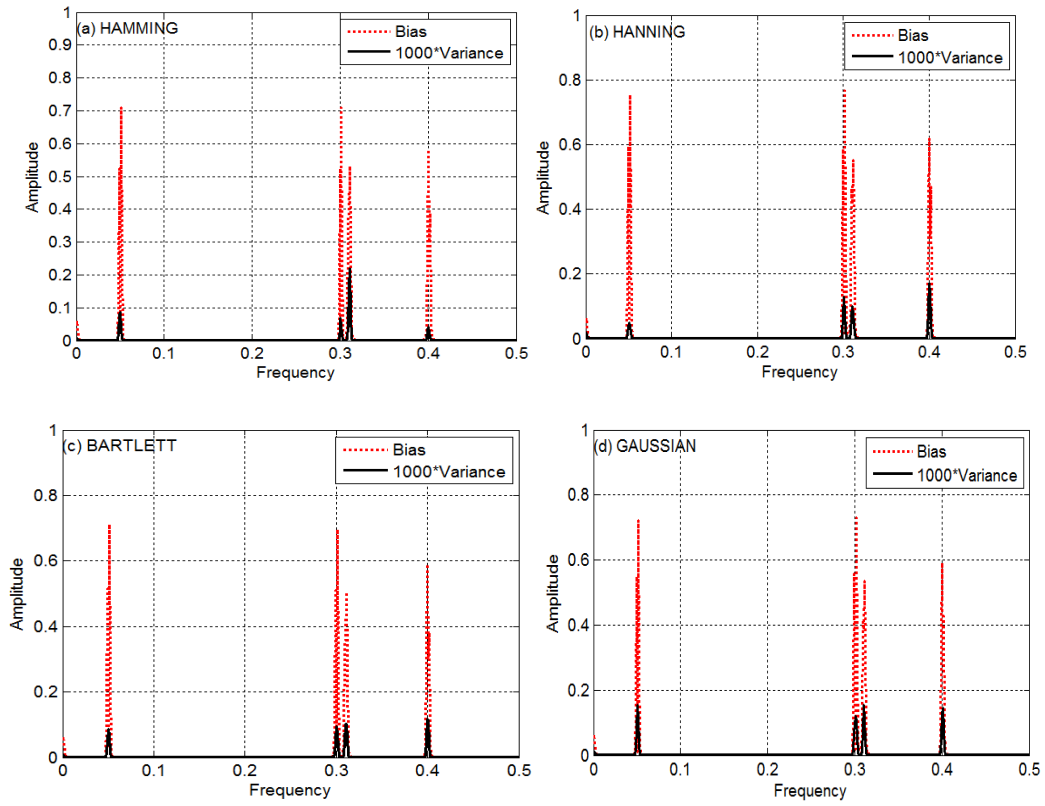


Fig. 4.5: Bias and variance of the PSD estimate of $N = 512$ sequence using (a) Hamming window, (b) Hanning window, (c) Bartlett window, (d) Gaussian window

In Fig. 4.5, we see that the bias of the periodogram resulting from all the four windows is almost the same. But, the variances are unequal. Even though, Hanning window exhibits similar properties as the hamming window, the variance of Hanning periodogram is slightly lower than that of the hamming periodogram. But however, the variance of Bartlett and the Gaussian window are the least among the four windows.

The next class of periodogram is the Bartlett periodogram that employs segmentation of the original sequence and computing periodogram of individual segments, followed by averaging the periodograms. The Bartlett periodogram estimates in Fig. 4.6, are obtained by averaging the periodogram of two segments.

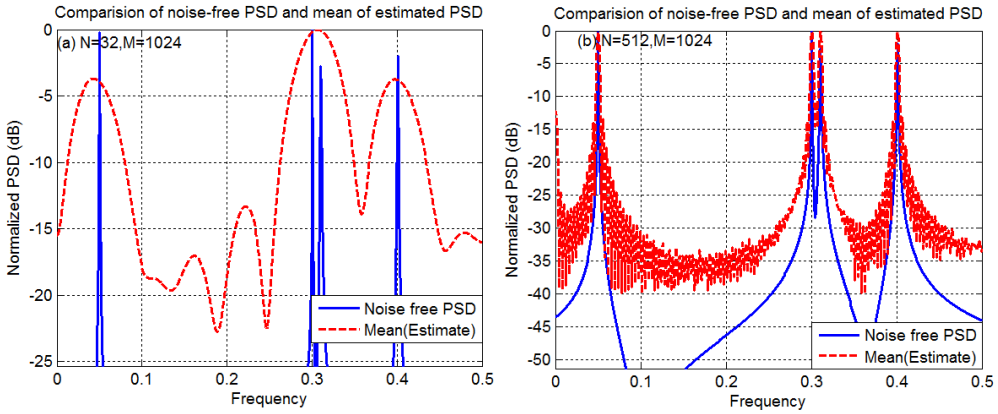


Fig. 4.6: Comparison of noise-free PSD and mean of estimated PSD using Bartlett periodogram for (a) N=32, (b) N=512

From Fig.4.7, we see that the resolution of the Bartlett periodogram decreases as the length of the signal increases, so does the variance. As N increases, the bias also increases, yet the bias of the Bartlett periodogram is better than the un-windowed periodogram.

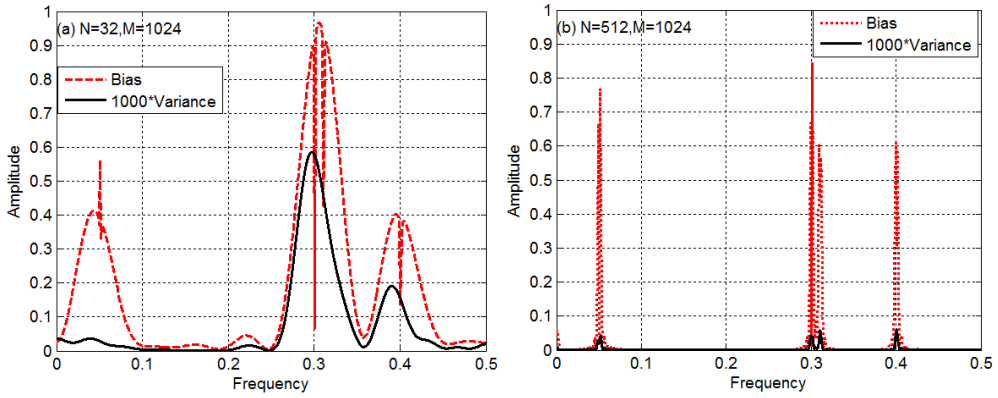


Fig. 4.7: Bias and variance of the Bartlett periodogram estimate for (a) N=32, (b) N=512

The generalized version of the Bartlett periodogram is the Welch periodogram which incorporates overlapping of the windowed subsequences. As seen from Fig. 4.8, the Welch periodogram is not a good estimator for two closely spaced frequencies. Welch periodogram does not provide good resolution due to averaging the windowed sequences and we also note that the close frequencies get smeared. In the case of windowed periodograms or the Bartlett periodogram, overlapping is not performed. Hence they offer better resolution than the Welch periodogram. However, the resolution of the Welch periodogram increases as the number of observations increases unlike in the case of Bartlett periodogram.

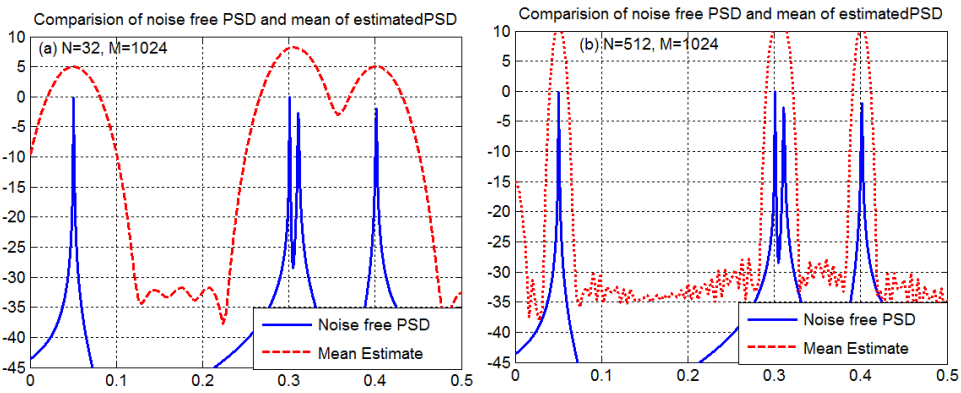


Fig. 4.8: Comparison of noise-free PSD and mean of the estimated PSD using Welch method for (a) N=32, (b) N=512

It is evident, from Fig. 4.9, that the bias of the Welch periodogram is very high for large N. On the other hand, variance decreases as N increases. This shows that the Welch periodogram is not asymptotically unbiased, but a consistent estimate. The variance is almost zero due to averaging the overlapped segments.

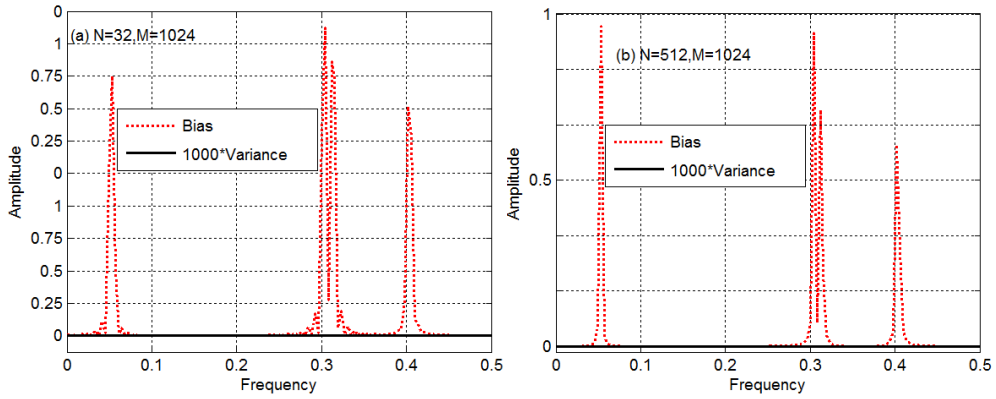


Fig. 4.9: Bias and variance of the Welch estimate for (a) *N*=32, (b) *N*=512

4.3.2 Range Doppler map using periodogram estimators

The periodogram estimators used in thesis are the un-windowed periodogram, windowed periodogram, Bartlett periodogram and the Welch periodogram as in sections 3.4, 3.5, 3.7, and 3.9, respectively.

The range-Doppler matrix of the simulated target is obtained by taking the Fourier Transform of the CPI data matrix. Magnitude of the range-Doppler matrix is the range-Doppler map which is presented in the form of contour map. First, we consider the periodogram method used to generate the range-Doppler map and the resolution in Doppler as in Fig. 4.10.

The red color in Fig. 4.10a indicates the peaks present in the range-Doppler map and the blue regions represent the valleys. The Doppler resolution is observed by taking a cut-plot along the *R*=5000 meters as in Fig. 4.10b. We notice that there are multiple sidelobes in the Doppler resolution plot. This illustrates the inability of the periodogram technique to suppress the sidelobes. Next, we consider the windowed periodogram.

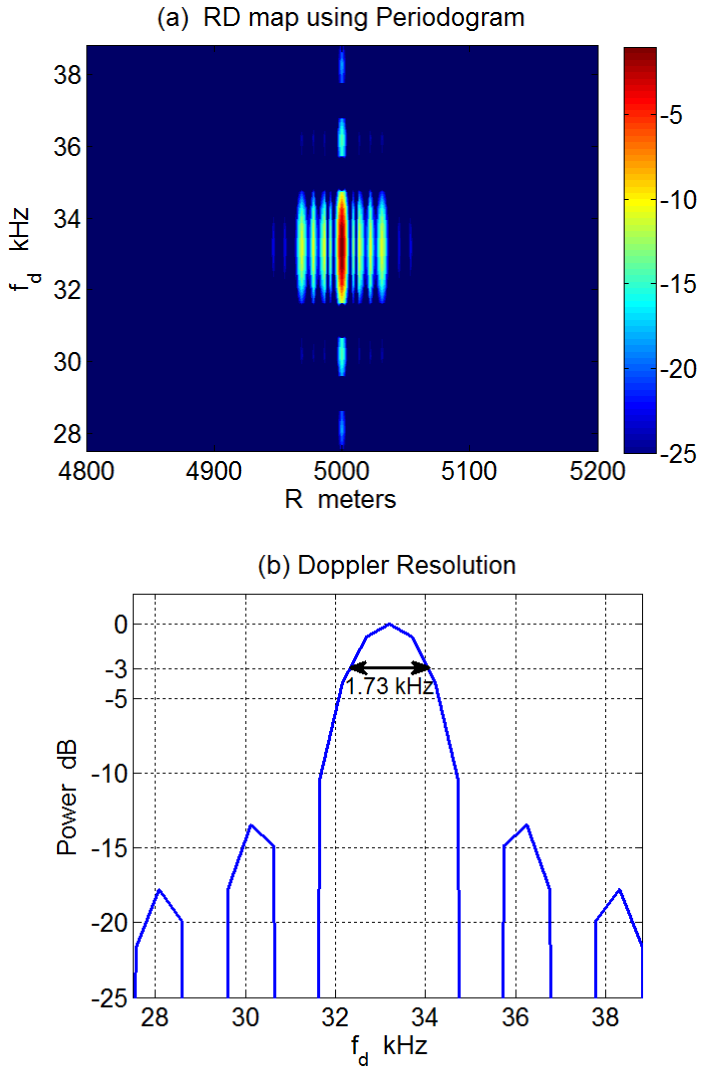


Fig. 4.10: Periodogram (a) Range Doppler map, (b) Doppler resolution

Fig. 4.11a shows the range-Doppler map obtained by using a Gaussian windowed periodogram. Only the presence of main lobes is noticed without any sidelobes. It is true that by using a window, sidelobe can be reduced. But however, the width of the mainlobe increases with the usage of the window. The Doppler resolution in Fig. 4.11b is 2.82 kHz, which is much wider than the periodogram (1.73 kHz). This demonstrates that the resolution decreases with the use of windows, and sidelobes are absent.

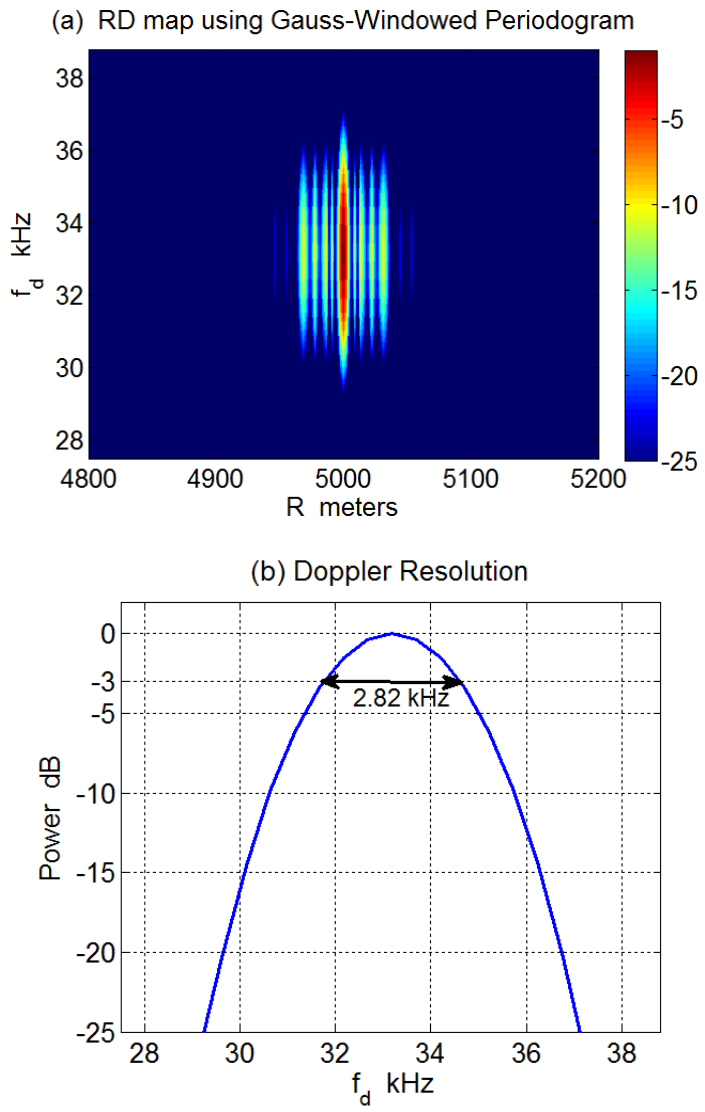


Fig. 4.11: Gauss windowed Periodogram (a) Range Doppler map, (b) Doppler resolution

The range-Doppler map in Fig. 4.12a appears to be better than the periodogram. But however, the Doppler resolution in Fig. 4.12b has a main lobe width of 5.76 kHz due to the averaging. This enormous main lobe width is caused due to sectioning of the available record of data and then averaging the periodograms of the sections. In spite of absence of the sidelobes, the large Doppler width is a drawback of the Bartlett

periodogram. These results indicate that the Bartlett periodogram is not a suitable technique for range-Doppler mapping.

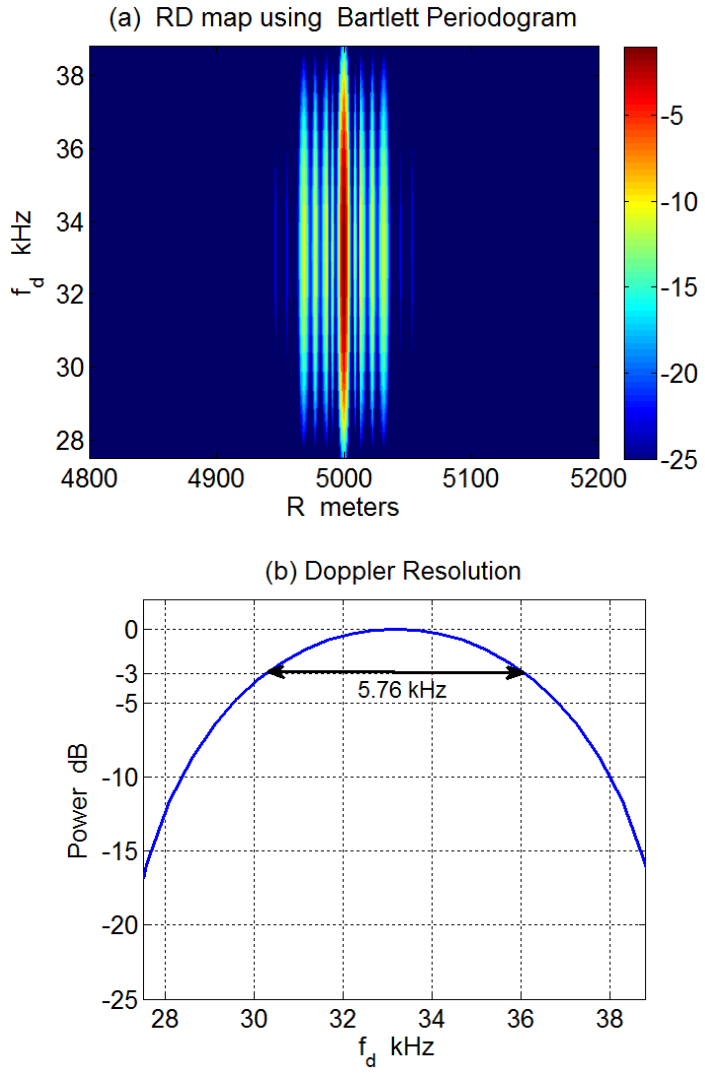


Fig. 4.12: Bartlett Periodogram (a) Range Doppler map, (b) Doppler resolution

As seen from the previous discussions, the Welch periodogram is a special case of Bartlett periodogram with windowed overlapping segments. Fig. 4.13 was obtained by applying a Gauss window to the individual segments. Although the Doppler resolution is not very high, this method has the advantage that unlike the Bartlett periodogram where

the segments are not allowed to overlap, there are more segments created using this approach, allowing a better estimate than the Bartlett approach.

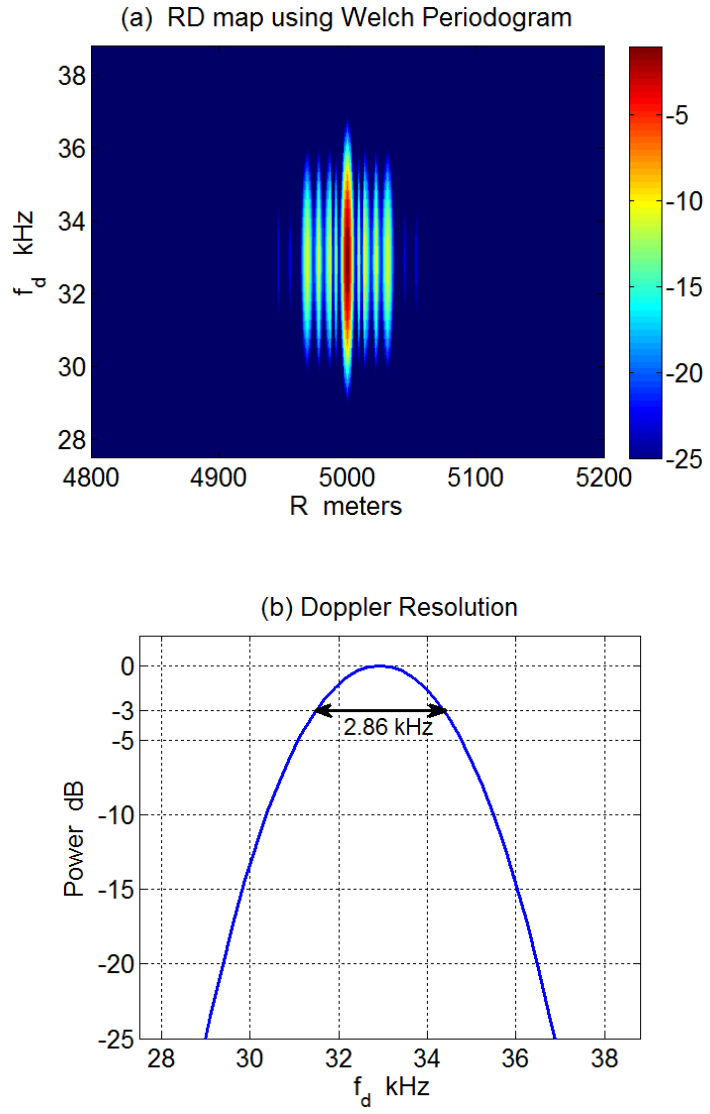


Fig. 4.13: Gauss windowed Welch Periodogram (a) Range Doppler map, (b) Doppler resolution

The Doppler resolution resulting from the Welch periodogram is 2.86 kHz, which is wider than the periodogram, but half the width of Bartlett periodogram.

4.4 Correlogram estimators

The correlogram estimators rely on the estimate of the autocorrelation function where the lag is no more than the signal length *N*. The un-windowed correlogram and the Blackman-Tukey correlogram methods are discussed in this section.

4.4.1 Bias and variance of correlogram estimators

Fig. 4.14 demonstrates the effect of the signal length on the bias and variance of the Correlogram estimate. It is clearly seen from Fig. 4.14 that the correlogram performs better for large sequence than for a short sequence. The frequencies 0.3 and 0.31 are well resolved for *N*=512, unlike in the case *N*=32.

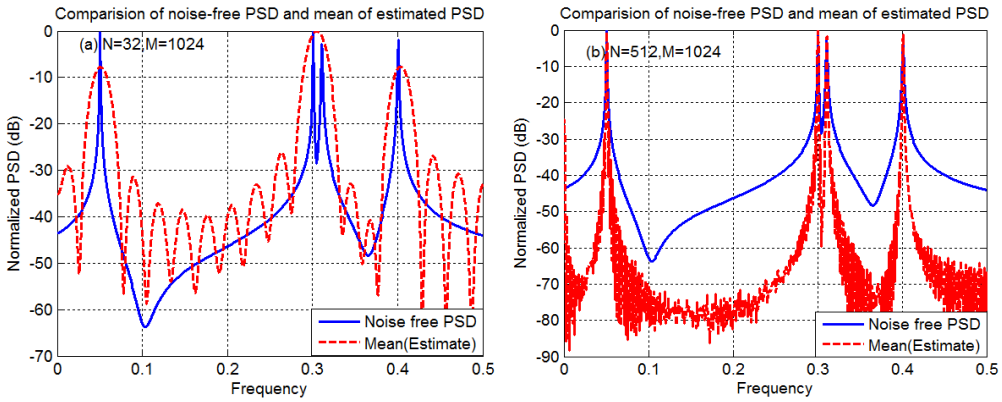


Fig. 4.14: Comparison of noise-free PSD and mean of estimated PSD using the Correlogram for (a) *N*=32, (b) *N*=512

From Fig. 4.15, we see that the bias and variance of the correlogram decreases as the length of the signal increases. The bias and variance of the correlogram are much lesser than the periodogram.

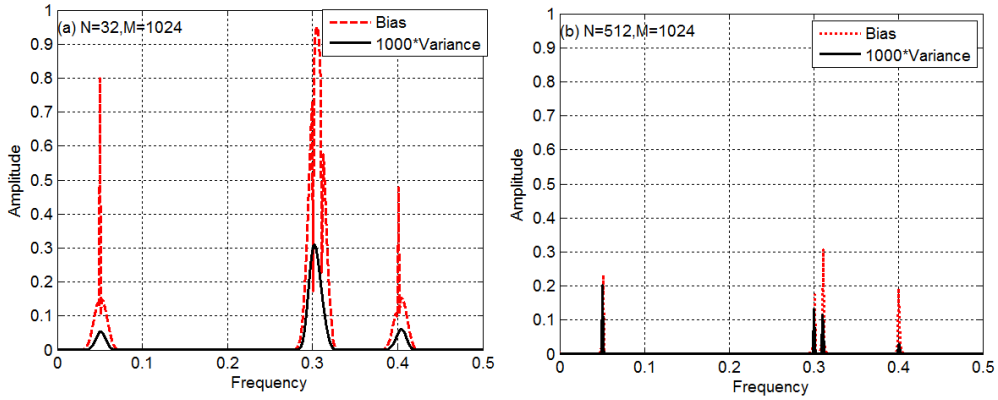


Fig. 4.15: Bias and variance of the correlogram estimate for (a) *N*=32, (b) *N*=512

Blackman-Tukey is the windowed correlogram. The most preferred window used in this method is the Bartlett window [2]. Fig. 4.16 shows the noise free PSD and the mean of the Blackman-Tukey correlogram for *N*=32 and *N*=512.

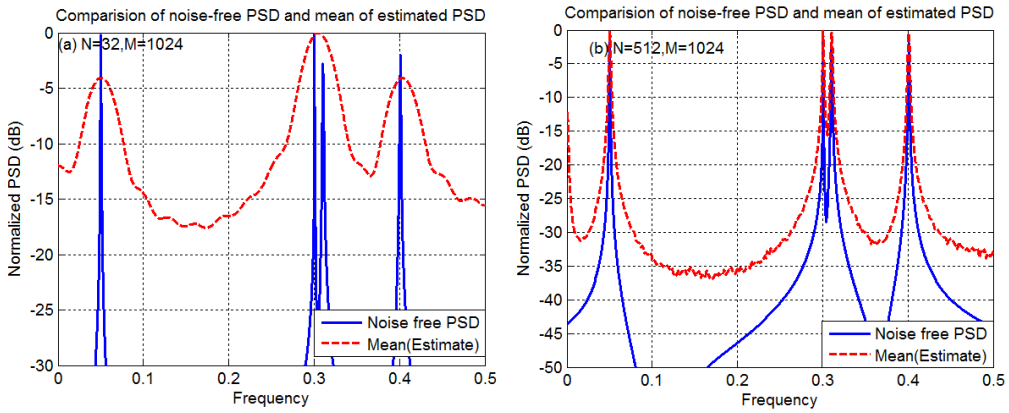


Fig. 4.16: Comparison of true PSD and mean of the estimated PSD using Blackman-Tukey method for (a) *N*=32, (b) *N*=512

As seen from Fig. 4.17, we notice that the bias of the Blackman-Tukey correlogram for *N*=512 is very small compared to *N*=32. In addition, we notice the bias of the Blackman-Tukey correlogram has reduced for in comparison with the un-windowed correlogram.

But, the estimate suffers from poorer resolution. This reduction is due to the usage of the Bartlett windows.

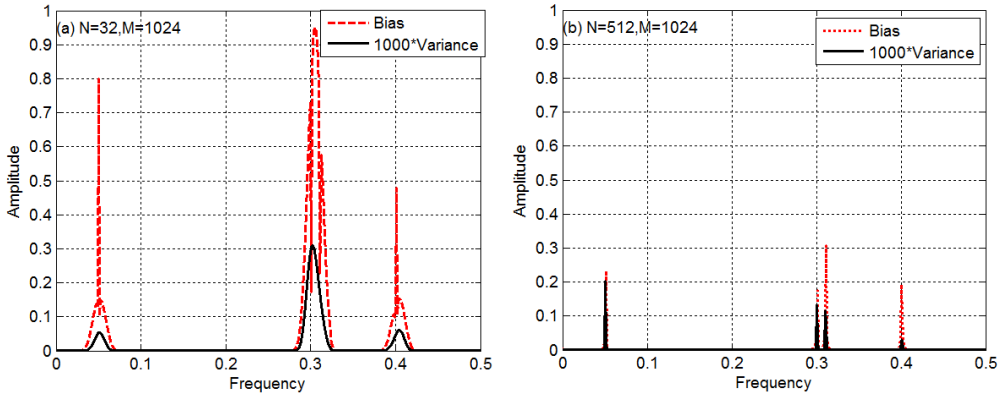


Fig. 4.17: Bias and variance of the Blackman –Tukey estimate for (a) *N*=32, (b) *N*=512

4.4.2 Range Doppler map using correlogram estimators

The correlogram estimators rely on the estimate of the autocorrelation function whose lag is no more than the signal length *N*. The un-windowed correlogram and the Blackman-Tukey correlogram methods are discussed in this section.

Fig. 4.18a shows the Range-Doppler map of the target simulated as per Table 4.1. The contour plot resulting from the correlogram is visually better than the previous techniques described in this thesis. The Doppler resolution in Fig. 4.18b is similar to that of the periodogram in 4.11b. Here, we see the absence of sidelobes unlike in the case of the periodogram, and the 3 dB width is narrower (1.22 kHz).

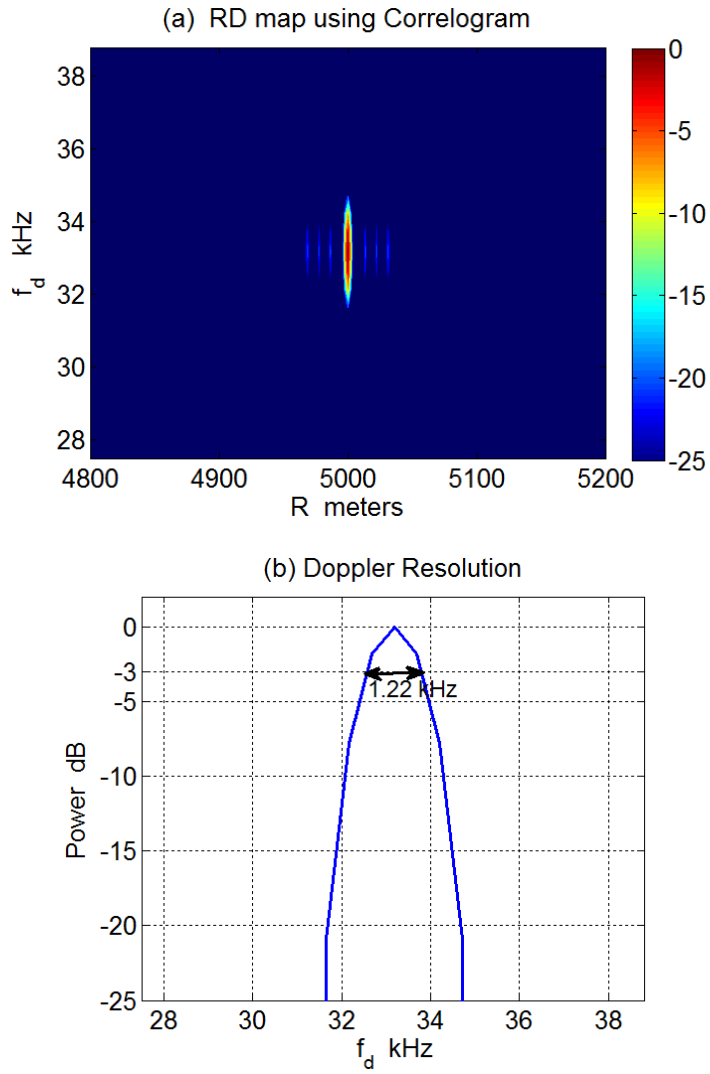


Fig. 4.18: Correlogram (a) Range Doppler map, (b) Doppler resolution

In Fig. 4.19a, the Range-Doppler map appears to be wider than the raw correlogram. This is illustrated in Fig. 4.19b in terms of the Doppler resolution. Here, we notice that there is no side lobe, but a single main lobe spreads out widely. This indicates that there is a peak at the Doppler frequency, but the overall resolution is not that good as compared to the raw correlogram. This widening of the mainlobe is due to windowing of the autocorrelation estimate.

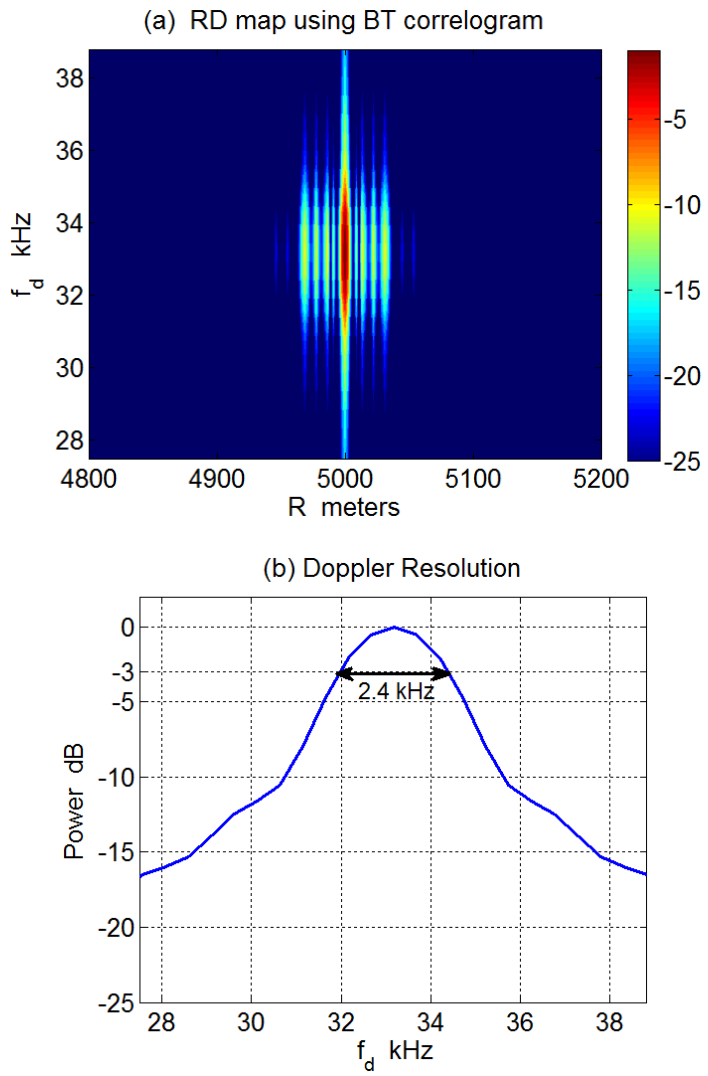


Fig. 4.19: Blackman-Tukey correlogram (a) Range Doppler map, (b) Doppler resolution

4.5 Autoregressive estimator

The AR estimator used here is an IIR filter. The filter coefficients are computed by using the Yule Walker equations.

4.5.1 Bias and variance of AR method

Fig. 4.20 shows the comparison of AR YW estimate using filter order p=10 for N=32 and N=512 with the noise free PSD. The bias and variance of the PSD estimate are shown in Fig. 4.21.

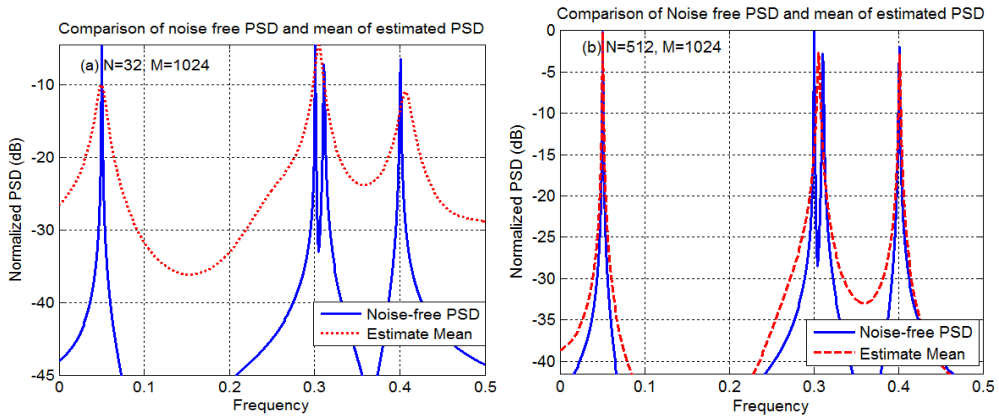


Fig. 4.20: Comparison of true PSD and mean of the estimated PSD using AR YW, p=10 for (a) N= 32, (b) N=512

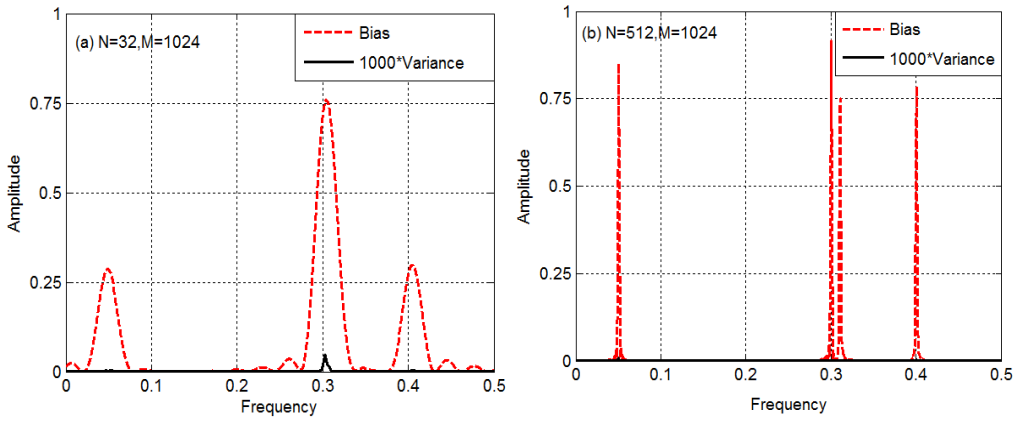


Fig. 4.21: Bias and variance of the AR Yule-Walker p=10 estimate for (a) N=32, (b) N=512

4.5.2 Range Doppler map using autoregressive estimator

Fig. 4.22a shows the Range-Doppler map resulting from using the AR YW method of model order p=10.

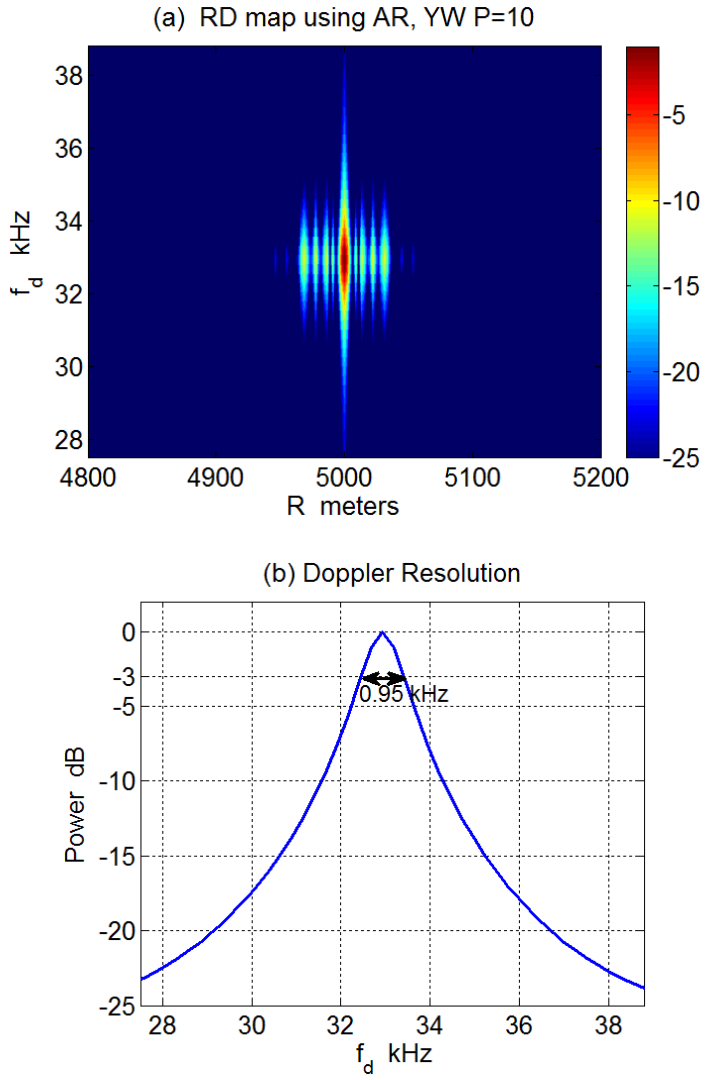


Fig. 4.22: AR YW, p=10 (a) Range Doppler map, (b) Doppler resolution

The Doppler resolution in Fig. 4.22b shows a sharp peak in the Doppler frequency with gradually decreasing mainlobe. This causes the Range-Doppler map to appear wider

than the previous spectral methods discussed. A considerable attenuation of the sidelobes is noticed when compared with the periodogram.

4.6 Comparison of execution time

The spectral estimators not only differ by their resolution and statistical parameters, but also vary in terms of time required for computing the PSD estimates. Table 4.2 shows the execution times for the spectral estimation techniques in MATLAB for 1 column and 2755 columns for the formation of range-Doppler map. The execution times in second and third columns of this table are in seconds. The simulations were performed on a personal computer having 4 gigabytes of RAM, Intel Dual-Core 2.4 GHz processor and MATLAB 7.10.

Table 4.2: Comparison of execution times of spectral estimators for one column and for the entire Range-Doppler map

Method	1 column	2755 columns
Periodogram	.0019	2.7350
Windowed Periodogram	.0019	2.8321
Correlogram	.0025	4.5868
Blackman-Tukey	.0027	4.9571
Bartlett Periodogram	.0019	2.7750
Welch Periodogram	.0054	9.7327
AR Yule Walker	.0061	10.8769

The above table is graphically interpreted, as in Fig. 4.23. By looking into this figure, one can infer that the periodogram method is the fastest, and the YW method is the slowest, among the techniques discussed in this thesis. With the background in Chapter 3, and by using the results of this chapter, Chapter 5 concludes with the advantages and disadvantages of the spectral estimation techniques.

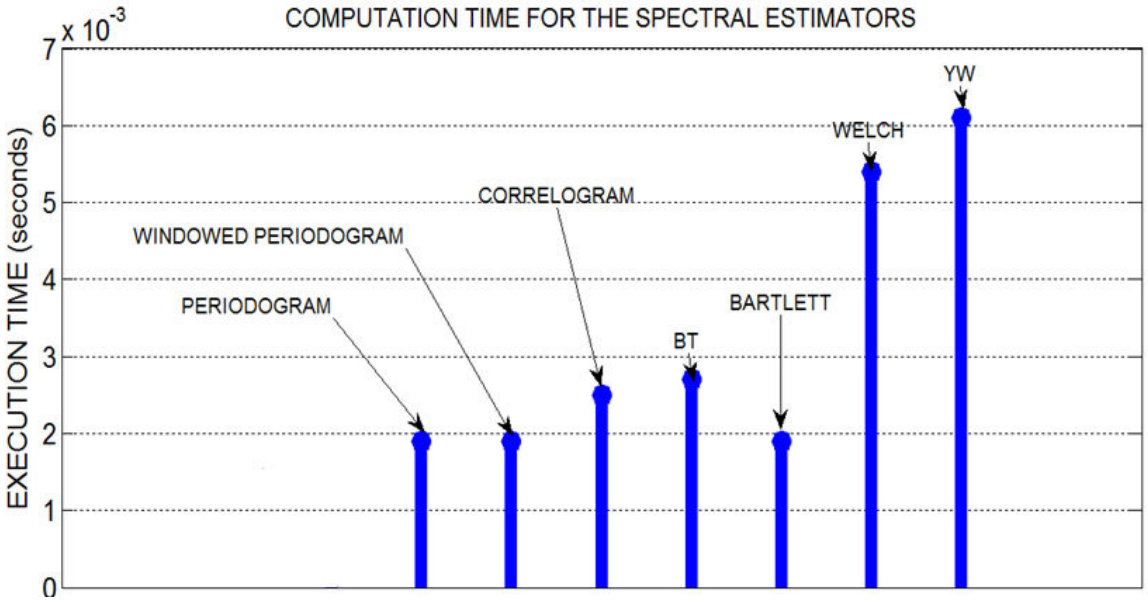


Fig. 4.23: Execution time of the spectral estimators for 1 column of range-Doppler map

Chapter 5

CONCLUSIONS

The following main objectives have been achieved in this thesis: (1) study of some of the most commonly used nonparametric and parametric spectral estimators, (2) study of bias and variance of the spectral estimators, (3) testing the applicability of the estimators to the range-Doppler map, and (4) evaluating the resolution and execution times for the spectral estimators for the formation of Range-Doppler map.

Many methods for spectral estimation have been studied over the years. But however, a few of them were implemented to demonstrate the utility of the spectral estimators. In Chapter 3, we presented a discussion on some of the most commonly used nonparametric and parametric spectral estimators. In Chapter 4, we studied the bias and variance of the estimators and applied the spectral estimation techniques to range-Doppler map to compare the resolution and the performance timings.

Throughout this investigation, we generated MATLAB codes for simulating the sinusoids, noise and aircraft target signals, and to form the Range-Doppler map. The execution times of the estimators were evaluated using the "tic"-'toc" commands in MATLAB.

The main conclusions of this work can be summarized as follows.

- The periodogram is the simplest and most natural estimator with comparatively poor resolution and larger sidelobes. These are due to the fact that the magnitude squared value of the FFT is divided by *N*. But, this problem was improved by applying the windows to the periodogram.
- Also, the averaging of the segmented periodograms such as Bartlett and Welch periodograms provided better estimates since more segments lead to better averaging.
- Correlogram estimators are slightly different to the periodogram estimators by the type of operations they use. Periodograms operate directly on the given sequence, whereas correlograms operate on the autocorrelation estimates of the sequence.
- Periodogram based estimators have better resolution and variance than the correlogram estimators. The bias of the correlogram estimates is lesser than that of the periodogram estimators.
- But the autoregressive estimator, on the other hand, offers the best resolution with a large computation time. The enormous time taken for the AR estimator is mainly due to the matrix inversion used to compute the filter coefficients.
- The windowed periodograms and the Blackman-Tukey correlogram suppress the sidelobes to a large extent, but the windowing causes the main lobe to be wider than the un-windowed case.

- The spectral estimation techniques involving segmentation of signals and averaging such as Bartlett and Welch methods are unsuitable for range-Doppler mapping unless the segments are of large length.

- The best Doppler resolution results from the AR method. But however, this method requires large computation time. On the other hand, correlogram offers a good resolution with less computation time. Thus we may conclude that the correlogram is well suited for simple range-Doppler mapping.

Future improvements can be made in two ways. First is by inclusion of scattering model and multiple targets, second is by using improved spectral estimation methods to test the Doppler resolution.

APPENDIX A

SOME STATISTICAL TERMS

1. Gaussian Distribution

The Gaussian distribution, or the Normal distribution, is an important continuous probability distribution whose probability density function (p.d.f.) looks like a bell-shaped curve. The p.d.f. of a Gaussian random variable X is defined [11] as

$$f_X(u) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{\frac{-(u-\mu)^2}{2\sigma^2}\right\},$$

where μ is the mean, and σ^2 is the variance of the random variable X. The probability distribution function of the random variable X is given by

$$F_X(u) = \Phi\left[\frac{u-\mu}{\sigma}\right], -\infty < u < \infty,$$

where

$$\Phi(x) = \int_{-\infty}^{x} \frac{\exp(-y^2/2)}{\sqrt{2\pi}} dy, -\infty < x < \infty,$$

is the standard Gaussian distribution function for a zero-mean, unit-variance, Gaussian random variable.

2. Wide-sense Stationary process

A second-order random process $N(t)$ is a wide-sense stationary (WSS) process if the following conditions are satisfied:

i. The mean of $N(t)$ is a constant, i.e.

$$E[N(t)] = \mu_X(t) = \mu_X .$$

ii. The autocorrelation of the process is function of the time difference only, i.e.

$$R_X(t, t+\tau) = R_X(\tau) .$$

3. White Noise

A noise with a flat PSD over all frequencies is the white noise. This is a theoretical concept, but practically the white noise has a flat PSD over the widest frequency used in a system. The white noise is an idealization of the thermal noise.

4. White Gaussian Noise

A random process $N(t)$ is a White Gaussian Noise (WGN) if it satisfies the following conditions:

i. $N(t)$ is a WSS process with zero mean.

ii. For any time t, the corresponding realizations of $N(t)$ are independent Gaussian random variables.

The autocorrelation function of a WGN process is

$$R_N(\tau) = \frac{N_0}{2}\delta(\tau), -\infty < \tau < \infty,$$

with a flat PSD given by

$$S_N(f) = \frac{N_0}{2}, -\infty < f < \infty,$$

where N_0 is the power per unit bandwidth of $N(t)$.

5. Band-limited White Gaussian Noise.

The white-Gaussian noise is a pure theoretical concept, and it does not exist is nature. So, a band-limited white Gaussian is implemented in practice for simulation purposes. In Fig. A.1, $S_{NN}(f)$ is a white noise of infinite bandwidth, H(f) is the transfer function of a system with a bandwidth $2B$ Hz. The $S_{NsNs}(f)$ is a white Gaussian noise band-limited to $f_s/2$ Hz, which is half the sampling frequency of the system. The sampling frequency, f_s is chosen to be greater than twice the bandwidth of the system such that the highest frequency of the band-limited white Gaussian noise appears to be a white-noise to the system under consideration [14]. The inverse Fourier transform of the rectangular pulse is a sinc wave that has zero crossing at $1/f_s$. A random number generator will produce uncorrelated samples at these zero crossings. This is the reason for choosing the bandwidth as f_s.

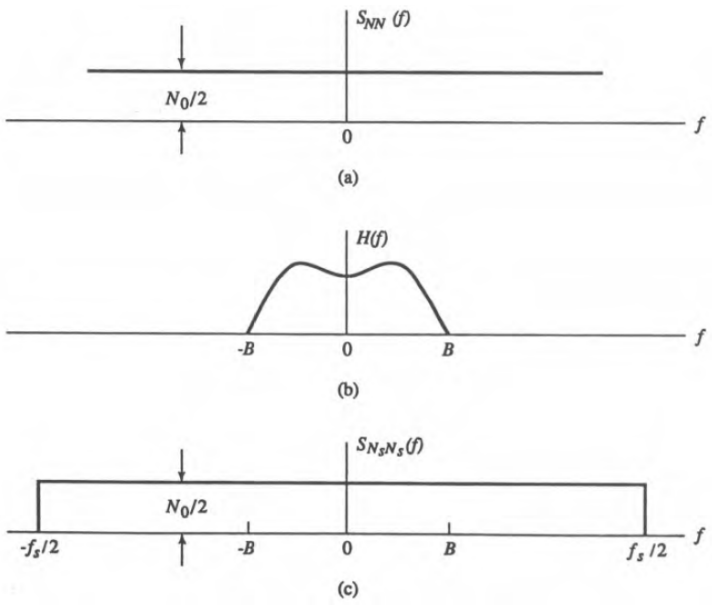


Fig. A.1: PSD of (a) White Gaussian noise, (b) Transfer function of a system, (c) Band-limited white Gaussian noise.

APPENDIX B

SOME COMMONLY USED WINDOWS

1. Rectangular Window

The rectangular window is nothing but a uniform window consisting of unit amplitude throughout the specified length. In MATLAB, the rectangular window can be generated by using *rectwin* , or the selected length can be set to 1.

Mathematically, the rectangular window of length N is defined as

$$w(n) = 1, \quad n = 0,1,2.....W_l. \tag{B.1}$$

Fig. B.1 shows a rectangular window and its magnitude spectrum.

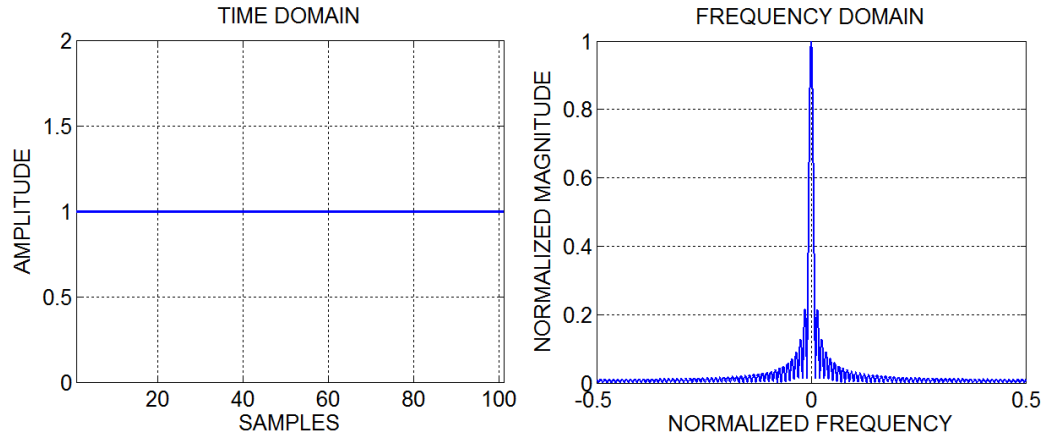


Fig. B.1: Rectangular window and its magnitude spectrum.

2. Hamming Window

The hamming and the hanning windows, both contain a single cosine function. There is very little difference in these two windows as seen by their mathematical forms. Eq. B.2 represents the Hamming window and Fig. B.2 demonstrates the Hamming window and its magnitude spectrum.

$$w(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{W_l}\right), \quad n = 0,1,2.....W_l \tag{B.2}$$

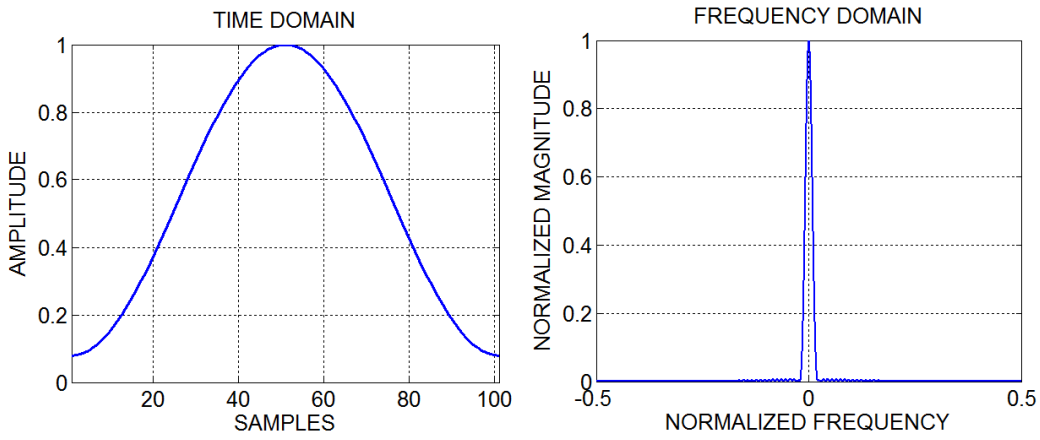


Fig. B.2: Hamming window and its magnitude spectrum

Eq. B.3 represents the Hanning window, and Fig. B.3 shows the Hanning window and its magnitude spectrum.

$$w(n) = 0.5\left[1 - \cos\left(\frac{2\pi n}{W_l}\right)\right], \quad n = 0,1,2.....W_l \tag{B.3}$$

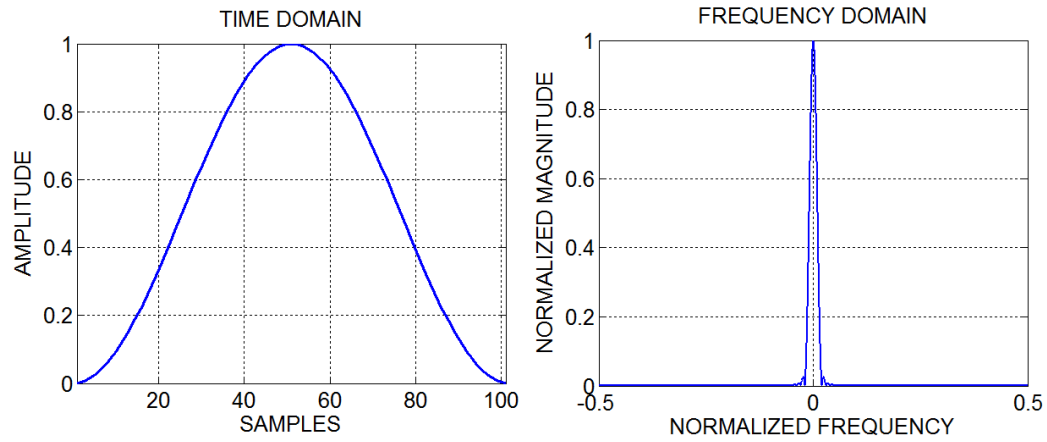


Fig. B.3: Hanning window and its magnitude spectrum

4. Bartlett Window

The Bartlett window is commonly called as triangular window, since its time domain plot appears as a triangle. The Bartlett window has a sharp taper at either ends. Mathematically, the Bartlett window is described as,

$$w(n) = \begin{cases} \dfrac{2n}{W_l}, & 0 \le n \le \dfrac{W_l}{2} \\ 2 - \dfrac{2n}{W_l}, & \dfrac{W_l}{2} \le n < W_l \end{cases}$$

(B.4)

Fig. B.4 shows a Bartlett window and its magnitude spectrum.

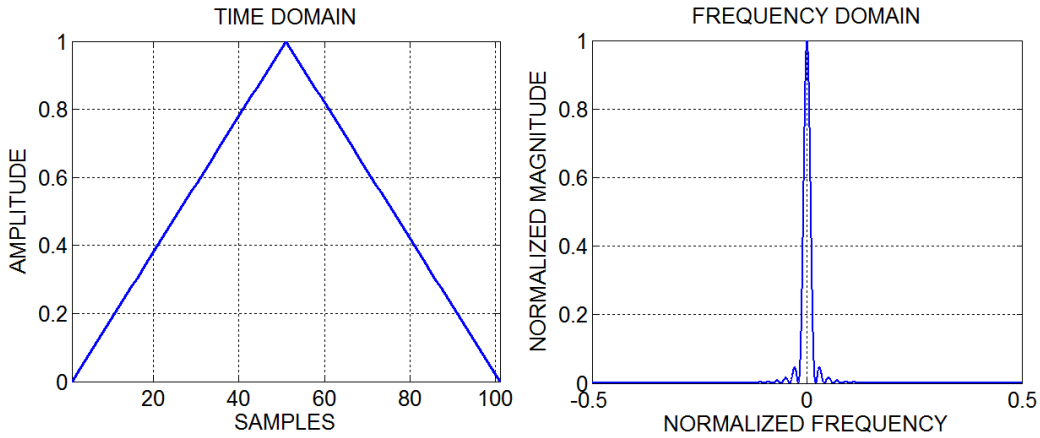


Fig. B.4: Bartlett window and its magnitude spectrum

5. Gaussian Window

The Gaussian window derived its name from the Gaussian distribution or the Bell curve. It is mathematically given by Eq. B.5, and demonstrated in Fig. B.5.

$$w(n) = \exp\left[\frac{-1}{2}\left\{\frac{n - 0.5W_l^2}{0.5\sigma W_l}\right\}^2\right], \quad n = 0,1,2.....W_l \tag{B.5}$$

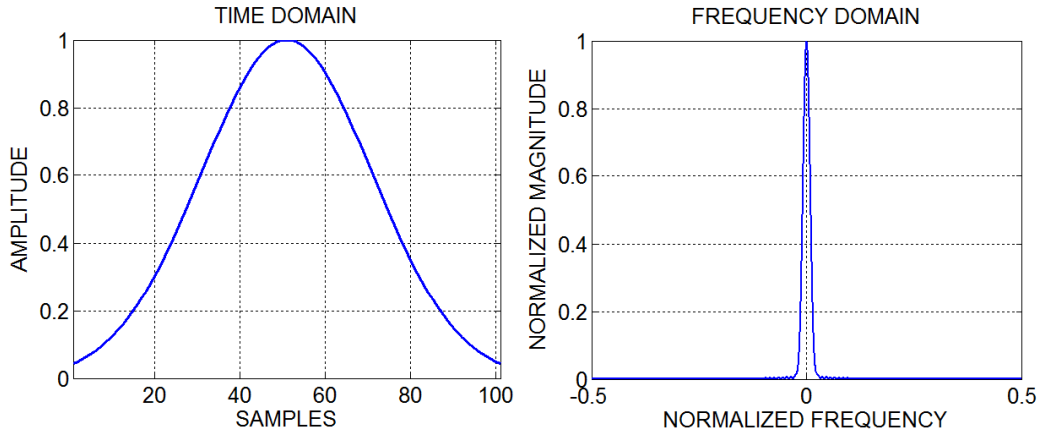


Fig. B.5: Gaussian window and its magnitude spectrum

The Table B.1 was formed, using a length of 101, for the windows described above. This table shows the different parameters of each window such as 3dB width, peak (or First) sidelobe level (SLL), last sidelobe level and the leakage factor.

Table B.1: Comparison of window parameters

WINDOW	-3dB width	First Sidelobe Level (dB)	Last Sidelobe Level (dB)	Leakage Factor (%)
RECTANGULAR	0.015625	-13.354	-40	9.26
HAMMING	0.025391	-42.54	-56.5	0.3
HANNING	0.027344	-31.5	-140	0.05
BARTLETT	0.025391	-26.5	-68	0.29
GAUSSIAN	0.027344	-43.8	-61.5	0.01

From the above table, we notice that the following facts.

- Rectangular window has the least 3dB width, larger sidelobe level and the worst leakage factor.
- Hamming and Bartlett window have same 3dB widths, and almost identical leakage factor. But however, the Hamming window has a better peak SLL, whereas the Bartlett window has a lower last SLL.
- The 3dB widths and peak SLL of the Hanning and Gaussian windows are identical. But, the Gaussian window does not suffer a high spectral leakage, as compared to the Hanning window.

APPENDIX C

MATLAB PROGRAMS

1. Program for simulating a radar target and plotting the range-Doppler map and Doppler resolution

```
% This MATLAB program is used to plot the range-Doppler map and the
% Doppler resolution for a fixed, arbitrary range and Doppler of a target
%-------------------------------------------------------------------------
clc
clear all;
close all;
%-------------------------------------------------------------------------
% Target Parameters

V = 250;        % Speed of the target (in m/s)
C = 3e8;        % Speed of light (in m/s)
R0 = 5000;      % Range of target (in meters)
H = 500;        % H is the target height (in meters)
f0 = 20e9;      % RF frequency of the Radar (in Hz)
td = 2*R0/C;    % Time delay in microseconds (in seconds)
theta = asin(H/R0);  % Angle to the target (in radians)
fd = (2*f0*V/C)*cos(theta); % Doppler frequency (in Hz)
PRF = 50e3;     % Pulse Repetition Freqency
PRI = 1/PRF;    % Pulse Repetion Interval
N_PRI = 32;     % Number of PRIs in 1 CPI
CPI = N_PRI*PRI;    % Coherent Processing Interval
%-------------------------------------------------------------------------

% Tx'ed signal section
% 13-bit Barker code for 1PRI = Tx,

Tx = zeros(1,26*20);

Tx(1,1:10) = 1 ;
Tx(1,11:14) = -1;
Tx(1,15:18) = 1;
Tx(1,19:20) = -1;
Tx(1,21:22) = 1;
Tx(1,23:24) = 1;
Tx(1,25:26) = -1;

% Tx(1:26) = 1;
PRI_length = length(Tx);
CPI_length = N_PRI*PRI_length;
```

```
%-------------------------------------------------------------------

% Time and Delay in Matlab array scale
timedelay = fix(td*1e6)*26; % t_d in array scale
freqdelay = fix(1e6/fd*26); % f_d in array scale


% PRI_time = Timescale for the Tx'ed signal for 1 PRI
PRI_time = linspace(0,PRI,PRI_length);

%CPI_time = Timescale for the Tx'ed signal for 1 CPI
CPI_time = linspace(0,CPI,CPI_length);

Rx_PRI_length = PRI_length + timedelay;
Rx_CPI_length = CPI_length + N_PRI*timedelay;
Rx_CPI = CPI+N_PRI*td;
Rx_CPI_time = linspace(0,Rx_CPI,Rx_CPI_length);
%-------------------------------------------------------------------

% Sinusoid with Doppler Freqency, fd.
Dop_wave = exp(1j*2*pi*fd*Rx_CPI_time);

%--------------------------
% figure(1)
% plot(Rx_CPI_time,real(Dop_wave));
% title('Re { complex Sinusoid } with frequency f_d','fontsize',12);
% xlabel(' Time in seconds ','fontsize',12);
% ylabel (' Amplitude ','fontsize',12);
%--------------------------

Dop_mat = zeros(N_PRI,Rx_PRI_length);
for i = 1:N_PRI
   i2=i-1;
   Dop_mat(i,:) = Dop_wave(i+(i2)*Rx_PRI_length-(i2):i*Rx_PRI_length);
end

%-------------------------------------------------------------------

% Rx_barker = td shifted barker signal
Rx_barker = zeros(1,Rx_PRI_length);
Rx_barker(timedelay+1:Rx_PRI_length) = Tx;

Rx_real = repmat(Rx_barker,N_PRI,1);

% Rx_clean = Doppler freq. shifted Rx signal
Rx_clean = Rx_real.*Dop_mat;

%--------------------------
% figure(2)
% contour(Rx_clean);
%--------------------------

% White noise
```

```
Noise = 0.1*(rand(size(Rx_clean)));

% Final Complex doppler shifted and noise corrupted Rx signal
Rx = Rx_clean+Noise;




% Matched filtereing

for i = 1:N_PRI

   cor_mat(i,:) = xcorr(Rx(i,:),Tx);
   cor_mat(i,:) = cor_mat(i,:)./max(cor_mat(i,:));
end

cor_mat = cor_mat;% + 0.1.*exp(1j*rand(size(cor_mat)));



%-------------------------------------------------------------------------
% Windowing
size_cmat = size(cor_mat);
rows = size_cmat(1);
colums = size_cmat(2);
 w1 = gausswin(rows);
% w1 = hamming (rows);
% w1 = hanning (rows);
% w1 = bartlett (rows);
% w2 = gausswin(colums);
% w = w1*w2';
%-------------------------------------------------------------------------
% FFT length = M
M = 128;
cor_len = length(cor_mat);
%-------------------------------------------------------------------------

tic_RD=tic;

for i =1:cor_len
   tic1=tic;
   % Range_dop(:,i) = fftshift(abs(fft(cor_mat(:,i),M)).^2);
   % Range_dop(:,i) = fftshift(abs(fperiodogram(cor_mat(:,i),M)));
   % Range_dop(:,i) = fftshift(abs(fperiodogram(w1.*cor_mat(:,i),M)));
   % Range_dop(:,i) = fftshift(abs(fbartperio(cor_mat(:,i),M,3)));
   % Range_dop(:,i)= fftshift((abs(fwelchm(cor_mat(:,i),M,80,8,50,4))));
   % Range_dop(:,i) =  fftshift(abs(fcorrelogram(cor_mat(:,i),N_PRI,M)));
   % Range_dop(:,i) = fftshift(abs(fBTcorrelogram(cor_mat(:,i),N_PRI,M,3)));
   % Range_dop(:,i) = fftshift(fwelch(cor_mat(:,i),M,50,4,10));
    Range_dop(:,i) = fftshift(abs(pyulear(cor_mat(:,i),10)));

   toc1=toc(tic1);
 end
 toc_RD=toc(tic_RD);
```

```
size_rd = size(Range_dop);
row1 = size_rd(1);
col1 = size_rd(2);
dopa = (1:1:row1)./row1;
dop_axis= dopa.*65.3522;
tempo=1:1:length(Range_dop);
range_axis=tempo.*6160.56./length(Range_dop);
Rmid = 2236;
R1 = 2160;
R2 = 2312;
% Rax = max(range_axis);

% Dmid = 65;
% D1 = 53;
% D2 = 77;
% Dax = max(dop_axis);

maxi_val= max(max(Range_dop));
Range_dop=(Range_dop./maxi_val);

%-------------------------------------------------------------------------
% To plot Range-Doppler map

figure(7)
con_level=-25:0;    % contour level
contour(range_axis,dop_axis,10*log10(Range_dop),con_level);
colorbar('fontsize',16)
title('(a)  RD map using ','fontsize',16);
xlabel(' R  meters ','fontsize',16);
ylabel (' f_d   kHz ','fontsize',16);
% text(5100,34,'kHz','fontsize',12);
axis([4800 5200 27.5 38.8]);
h=gca;
set(h,'fontsize',16);
fh7 = figure(7);
set(fh7, 'color', 'white')

%-------------------------------------------------------------------------
% To plot the Doppler resolution
figure(8)
plot(dop_axis,10*log10((Range_dop(:,Rmid))) ,'linewidth',2);
title('(b) Doppler Resolution','fontsize',16);
xlabel(' f_d  kHz','fontsize',16);
ylabel (' Power  dB ','fontsize',16);
grid on
axis([27.5 38.8 -25 2])
h=gca;
set(h,'fontsize',16);
fh8 = figure(8);
set(fh8, 'color', 'white');

%-------------------------------------------------------------------------
disp('Time for 1 column');
disp(toc1);
```

```
disp('Time for RangeDoppler Map');
disp(toc_RD);
%--------------------------------------------------------------------
```

2. Matlab program for comparing the noise-free PSD and the noise corrupted version and to plot the bias and variance of the estimates.

```
%--------------------------------------------------------------------

clc
clear all;
close all;
%--------------------------------------------------------------------

% Frequency components of the signal

f1=.05;
f2=0.3;
f3=0.31;
f4=0.4;

%--------------------------------------------------------------------
N1=1024;        % Length of the observation signal
t1=0:1:N1-1;    % Time indices
M=1024;
f_axis=linspace(-0.5,0.5,M);  % One-Sided Frequency axis

% Original signal for evaluting the Approximated PSD
x1=exp(1j*2*pi*f1*t1)+exp(1j*2*pi*f2*t1)+exp(1j*2*pi*f3*t1)+exp(1j*2*pi*f4*t1);
true_psd=abs(fftshift(fft(x1,M)).^2);  % FFTSHIFT to be considered
true_psd= true_psd./max(true_psd);
true_PSD_dB = 10*log10(true_psd);

figure(1)
plot(f_axis,true_PSD_dB+21.5,'r-','linewidth',2);
title('Normalized noise free PSD','fontsize',12);
grid on;
axis([0 0.5 -65 5])

%--------------------------------------------------------------------
% Reference signal for simulation purposes

N=512;       % Length of the observation signal
t=0:1:N-1;   % Time indices

% Noise free signal
x=real(exp(1j*2*pi*f1*t)+exp(1j*2*pi*f2*t)+exp(1j*2*pi*f3*t)+exp(1j*2*pi*f4*t));

%--------------------------------------------------------------------
% Monte Carlo Trails
```

```
reptimes=10;    % Number of Monte Carlo Trials
noise_var=0.25;
xcorupt=fmcarlo(x,reptimes,noise_var);
y=xcorupt;

%--------------------------------------------------------------------
% PSD Estimators

for i=1:reptimes

   % FFT
    PSD_est(i,:) = abs(fftshift(fft(y(i,:),M)));
   % Periodogram
   % PSD_est(i,:) = abs(fftshift((fperiodogram(y(i,:),M))));
   % Windowed periodogram
   % w1=hamming(N);
   % w1=hann(N);
   % w1=bartlett(N);
   % w1=gausswin(N);
   % PSD_est(i,:) = abs(fftshift((fperiodogram(w1'.*y(i,:),M))));
   % Bartlett Periodogram
   % PSD_est(i,:) = abs(fftshift((fbartperio(y(i,:),M,4))));
   % Welch Periodogram
   % PSD_est(i,:)= abs(fftshift(fwelchm(y(i,:),M,50,1)));
   % Correlogram
   % PSD_est(i,:)= abs(fftshift(fcorrelogram(y(i,:),N,M)));
   % Blackman-Tukey Correlogram
   % PSD_est(i,:)= abs(fftshift(fBTcorrelogram(y(i,:),N,M,3)));
   % Yule Walker
   % M=2047;
   % PSD_est(i,:) = abs((pyulear(y(i,:),10,M)));

end

% Normalize the PSD by dividing the maximum value

max_PSD = max(max(PSD_est));
PSD_est = PSD_est ./ max_PSD;
PSD_dB = 10*log10(PSD_est);
min_PSD = min(min(PSD_dB));
max_PSD = max(max(PSD_dB));



% Plot all realizations of the Estimated PSD

% figure (2)
% for i=1:reptimes
%     plot(f_axis,(PSD_dB(i,:)));
%     hold on
% end
% title('PSD estimate using FFT','fontsize',12);
% grid on
```

```
% xlabel(' Frequency ','fontsize',12);
% ylabel(' Normalized PSD(dB) ','fontsize',12);
% axis([0 0.5 min_PSD max_PSD]);
% text(0,0,'(b) N=512,M=1024','fontsize',12);
% h=gca;
% set(h,'fontsize',12);
% set(h,'XTick',[0 0.1 0.2 0.3 0.4 0.5]);
% fh2 = figure(2);
% set(fh2, 'color', 'white');

%------------------------------------------------------------------------
% Compute the qualities of an estimator
[a,b,v]=festqualities(true_psd,PSD_est);
a = a./max(a);
a_log = 10*log10(a);
%------------------------------------------------------------------------
% Comparision of true PSD and mean of estimated PSD
figure(3)
plot(f_axis,true_PSD_dB,f_axis,a_log,'r--','linewidth',2);
grid on;
title('Comparision of noise-free PSD and mean of estimated PSD','fontsize',12)
xlabel('Frequency','fontsize',12);
ylabel('Normalized PSD (dB)','fontsize',12);
legend('Noise free PSD','Mean(Estimate)');
axis([0 0.5 min_PSD max_PSD]);
text(0,-1,'() N=2,M=1024','fontsize',12);
h=gca;
set(h,'fontsize',12);
set(gca,'XTick',[0 0.1 0.2 0.3 0.4 0.5]);
fh3 = figure(3);
set(fh3, 'color', 'white');

%------------------------------------------------------------------------
% Plot Bias and variance

figure(4)
plot(f_axis,abs(b),'r',f_axis,1000*v,'k-','linewidth',2);
xlabel('Frequency','fontsize',12)
ylabel('Amplitude','fontsize',12);
legend('Bias','1000*Variance');
text(0,0.95,'() N=2,M=1024','fontsize',12);
axis([0 0.5 0 1]);
grid on;
h=gca;
set(h,'fontsize',12);
set(gca,'XTick',[0 0.1 0.2 0.3 0.4 0.5]);
fh4 = figure(4);
set(fh4, 'color', 'white');
%------------------------------------------------------------------------
```

3. Function to compute the periodogram of a sequence

```
function [perio]=fperiodogram(signal,M)
```

```
N=length(signal);
perio=(abs(fft(signal,M)).^2)/N;
end
%--------------------------------------------------------------------
```

4. Function to compute correlogram

```
%--------------------------------------------------------------------
%   This function computes Standard biased estimate of the Correlogram    %
%--------------------------------------------------------------------

function [Corrgram] = fcorrelogram(signal,bias_length,M)

%  bias_length = k

N=length(signal);
K(1:bias_length)=0:1:bias_length-1;

for s=1:bias_length
   r=0;
   for i=s:N
      r=signal(i)*conj(signal(i-K(s)))+r;
   end
   r_est(s)=r;
end

% R_est is the Biased Estimate

lr=length(r_est);   %  Length of R_estimate
R_est=zeros(1,2*lr-1);
R_est(1:lr-1)=fliplr(conj(r_est(2:lr))); % Negative side of Autocorrelation
R_est(lr:2*lr-1)=r_est(1:lr);   % Positive side of autocorrelation

R_est=R_est./N;            % Final R_estimate

%--------------------------------------------------------------------
%              Corellogram estimate
%--------------------------------------------------------------------

% Correlogram is the DFT of Autocorrelation estimate function
Corrgram=(abs(fft(R_est,M))).^2;
Corrgram=Corrgram./length(Corrgram);

%--------------------------------------------------------------------
```

5. Function to compute Bartlett periodogram

```
% This function computes the bartlett PSD of a given Signal.
```

```
function PSD_bart = fbartperio(signal,M,Nseg)

   N=length(signal);
   N1=fix(N/Nseg);

   for i=1:Nseg
      i1=i-1;
      x(i,:)=signal(i1*N1+1:i*N1);
      X(i,:)=fperiodogram(x(i,:),M);
   end


   PSD_bart = sum(X)/Nseg;
end
%-----------------------------------------------------------------------
```

6. Function to compute Blackman-Tukey correlogram

```
% This function computes the Blackman-Tukey PSD
% PSD_BT = the final BT-PSD
% signal = The input signal for which PSD is to be estimated
% Bias_length = the desired autocorrelation estimate lag
% M = M-points in the DFT
% wintype = 1 = Hamming window
%           2 = Hanning window
%           3 = Bartlett window
%           4 = Gaussian window
%-----------------------------------------------------------------------

function PSD_BT = fBTcorrelogram(signal,bias_length,M,wintype)

  % Biased Autocorrelation Estimate
   Rcap_BT = frestimate(signal,bias_length);
   lengt1 = length(Rcap_BT);
 %----------------------
 % Window selection
   if (wintype == 1)
      window = hamming(lengt1);
   elseif (wintype == 2)
      window = hanning(lengt1);
   elseif (wintype == 3)
      window = bartlett(lengt1);
   elseif (wintype == 4)
      window = gausswin(lengt1);
   end
   Rwin_BT = window'.*Rcap_BT;
 %----------------------



 % Blackman Tukey PSD estimate
```

```
   PSD_BT = (abs(fft(Rwin_BT,M)));
end
%-----------------------------------------------------------------
```

7. Function to compute biased autocorrelation estimate

```
%-----------------------------------------------------------------
%   This function computes Standard biased estimate R^(k)              %
%-----------------------------------------------------------------------

function [R_est] = frestimate(signal,bias_length)

%  bias_length = k

N=length(signal);
K(1:bias_length)=0:1:bias_length-1;

for s=1:bias_length
   r=0;
   for i=s:N
      r=signal(i)*conj(signal(i-K(s)))+r;
   end
   r_est(s)=r;
end

% R_est is the Biased Estimate

lr=length(r_est);
R_est=zeros(1,2*lr-1);
R_est(1:lr-1)=fliplr(conj(r_est(2:lr)));
R_est(lr:2*lr-1)=r_est(1:lr);

R_est=R_est./N;
%-----------------------------------------------------------------------
```

8. Function to compute Welch periodogram

```
% This function computes the Welch periodogram estimate for a given signal
% signal = time domain input for which welch PSD is to be estimated
% OL = Overlap factor in percentage such as 50,40,30 etc
% Nseg = # of segments to be overlapped
% M = # of points in DFT
% LS = Length of subsequence

function PSD_welch = fwelchm(signal,M,OL,Nseg,LS,win)

N = length(signal);
OS = LS*((100-OL)/100);
```

```
if win==0
   w=ones(1,LS);
elseif win==1
   w=hamming(1,LS);
elseif win==2
   w=hann(1,LS);
elseif win==3
   w=bartlett(1,LS);
elseif win==4
   w=gausswin(1,LS);
end

U=sum(w.*w)/LS;

for i=1:Nseg
   i1=i-1;
   segment(i,:)=signal(i1*OS+1:i1*OS+LS);
   Segmented_perio(i,:)=fperiodogram(w.*segment(i,:),M)/U;

end

 PSD_welch = sum(Segmented_perio)/Nseg;

end
```

9. Function to compute random values for repeated trials

```
% This function uses Monte carlo method to repeat the process
% In this function, the signal is repeated to the required number of times
% and a white noise is added
% Insignal = signal to be realized again and again
% reptimes = Number to repeats or realizations
% sigmat = Signal matrix repeated
% randmat = random matrix having size of sigmat


function [xout]=fmcarlo(insignal,reptimes,noise_var)

sigmat=repmat(insignal,reptimes,1);
randmat=noise_var*rand(size(sigmat));
xout=sigmat+randmat;
end
%--------------------------------------------------------------------------
```

10. Function to compute the bias and variance of estimators

```
% This function is used to evaluate the qualities of an estimator
```

```
% here Mean ,Bias and Variance of an estimator are calculated.
% avg = mean of the estimate
% Bias = True value - Estimated value
% Varianc = Variance of the estimate


function [avg,bias,varianc]=festqualities(true_sig,corrupted_sig)

[r,c]=size(corrupted_sig);

avg=mean(corrupted_sig);

bias=true_sig-avg;

avgrepmat=repmat(avg,r,1);
% avgrepmat is used to replicate the avg matrix in calculating the bias
% since the Matrix subtraction is faster than the for loop

varmat=avgrepmat-corrupted_sig;
varianc=mean(varmat.^2);

end
%-------------------------------------------------------------------------
```

11. Program to plot the windows and their magnitude spectrum in Appendix B

```
clc
clear all;
close all;


N=101;
M=1024;
w=zeros(5,N);
W=zeros(5,M);
w(1,:)=rectwin(N);
w(2,:)=hamming(N);
w(3,:)=hanning(N);
w(4,:)=bartlett(N);
w(5,:)=gausswin(N);

% for i=1:5
%    wvtool(w(i,:));
% end

for i=1:5
   figure(i)
   plot(w(i,:),'linewidth',2);
   grid on;
   title('TIME DOMAIN','fontsize',16)
   axis([1 N 0 1]);
   xlabel('SAMPLES','fontsize',16);
```

```
    ylabel('AMPLITUDE','fontsize',16);
    h=gca;
    set(h,'fontsize',16);
    fh = figure(i);
    set(fh, 'color', 'white');
end


for i=6:10

    W(i-5,:)=abs(fft(w(i-5,:),M));
    W(i-5,:)=fftshift(W(i-5,:)./max(W(i-5,:)));
    Waxis=linspace(-0.5,0.5,M);
    figure(i)
    plot(Waxis,W(i-5,:),'linewidth',1.75);
    grid on;
    title('FREQUENCY DOMAIN','fontsize',16)
    axis([-0.5 0.5 0 1]);
    xlabel('NORMALIZED FREQUENCY','fontsize',16);
    ylabel('NORMALIZED MAGNITUDE','fontsize',16);
    h=gca;
    set(h,'fontsize',16);
    fh = figure(i);
    set(fh, 'color', 'white');
end


%-------------------------------------------------------------------------
```

REFERENCES

[1] Richards, M.A. *et al* , "Principles of Modern Radar", Scitech Publishing Inc., Raleigh, NC, 2010.

[2] Stoica, P. and Moses, R., "Spectral Analysis of Signals", Pearson-Prentice Hall, Upper Saddle River, NJ, 2005.

[3] Vaseghi, S., "Advanced Signal Processing and Noise Reduction", John Wiley & Sons Ltd,West Sussex, 2008.

[4] Sadate, C. "Spectral Estimation of Segmented Signals", Masters' thesis, Dept. of Electrical and Computer Engineering, The University of Alabama in Huntsville, 2005.

[5] Garner, J., "Spectral Estimation of Signal with Missing Data Blocks", Masters' thesis, Dept. of Electrical and Computer Engineering, The University of Alabama in Huntsville, 2006.

[6] Mehfaza, B., "Radar Signal Analysis and Processing using MATLAB", CRC Press, Boca Raton, FL, 2009.

[7] Calvo-Gallego, J. and Pérez-Martínez, F., "Simple traffic surveillance system based on range-doppler radar images," Progress In Electromagnetics Research, Vol. 125, 343-364, 2012.

[8] Tan, X. *et al*, "Range-Doppler Imaging via a Train of Probing Pulses", IEEE Transactions on Signal Processing, Vol. 57, No. 3, March 2009

[9] Welch, P.D., "The use of Fast Fourier transform for estimation of power spectra: A method based on time averaging over short, modified periodograms", IEEE Transactions on Audio and Electroacoustics, vol. AU- 15, pp.70-73, June 1967

[10] Kay, S.M. "Fundamentals of Statistical signal Processing: Estimation Theory," Prentice Hall, Upper Saddle River, NJ, 1993.

[11] Pursley, M.B., “Introduction to Digital Communications,” Pearson-Prentice Hall, Upper Saddle River, NJ, 2005.

[12] Quinquis, A. “Digital Signal Processing using MATLAB,” ISTE, London,UK and John Wiley & Sons Ltd, Hoboken, NJ, 2007.

[13] Poularikas, A.D., “THE HANDBOOK of Formulas and Tables for Signal Processing,” CRC Press, 1999.

[14] M.C. Jeruchim et al, “Simulation of communication systems,” Kluwer Academic Publishers, NY,2000

[15] Orfanidis, S.J., “Introduction to Signal processing,” Prentice Hall, Upper Saddle River, NJ, 2010.