

PARAMETRIC SPECTRAL ESTIMATION METHODS OF CLUTTER PROFILE
FOR ADAPTIVE RADAR DETECTION AND CLASSIFICATION

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

BERNA ERASLAN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

JULY 2019

Approval of the thesis:

**PARAMETRIC SPECTRAL ESTIMATION METHODS OF CLUTTER
PROFILE FOR ADAPTIVE RADAR DETECTION AND CLASSIFICATION**

submitted by **BERNA ERASLAN** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. İlkey Ulusoy
Head of Department, **Electrical and Electronics Engineering** _____

Assist. Prof. Dr. Gökhan Muzaffer Güvensen
Supervisor, **Electrical-Electronics Eng. Dept., METU** _____

Examining Committee Members:

Prof. Dr. Yalçın Tanık
Electrical and Electronics Eng. Dept., METU _____

Assist. Prof. Dr. Gökhan Muzaffer Güvensen
Electrical and Electronics Eng. Dept., METU _____

Prof. Dr. Sencer Koç
Electrical and Electronics Eng. Dept., METU _____

Prof. Dr. Çağatay Candan
Electrical and Electronics Eng. Dept., METU _____

Prof. Dr. Orhan Arıkan
Electrical and Electronics Eng. Dept., Bilkent University _____

Date:16.07.2019

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: BERNA ERASLAN

Signature :

ABSTRACT

PARAMETRIC SPECTRAL ESTIMATION METHODS OF CLUTTER PROFILE FOR ADAPTIVE RADAR DETECTION AND CLASSIFICATION

ERASLAN, BERNA

M.S., Department of Electrical and Electronics Engineering

Supervisor: Assist. Prof. Dr. Gökhan Muzaffer Güvensen

July 2019, 128 pages

Identification of unwanted echoes in a received radar signal is crucial in order to improve the radar detection performance. In the scope of thesis, currently proposed parametric spectrum estimation techniques, such as MUSIC, ESPRIT and Burg, are evaluated in order to estimate moments of clutter components in received radar echo. Since none of these methods has the ability of estimating Doppler spread and adequate accuracy, Stochastic Maximum Likelihood (SML) method is implemented, working with the best performing optimization and line search method. Since SML estimation accuracy is highly initial point dependent and computationally expensive, a novel estimation technique (Turbo) is proposed which works recursively. Proposed Turbo method outperformed the methods suggested in literature with its high Doppler resolution, accuracy and low computational cost. Moreover, Turbo performance is optimized by utilizing Burg estimates for initial point selection. After designing nearly optimal estimator, estimated parameters is used to design the detection filter which maximizes the Normalized SINR at its output even with a small number of secondary data. Finally, for clutter classification, a problem specific Neural Network architec-

ture is designed. The proposed Neural Network performance is also evaluated with estimates of novel Turbo method.

Keywords: radar signal processing, clutter power spectrum, parameter estimation, neural networks, maximum likelihood estimation, adaptive detectors

ÖZ

ADAPTİF RADAR TESPİTİ VE SINIFLANDIRMASI İÇİN KARGAŞA SPEKTRUMU PARAMETRE KESTİRİM METOTLARI

ERASLAN, BERNA

Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü

Tez Yöneticisi: Dr. Öğr. Üyesi. Gökhan Muzaffer Güvensen

Temmuz 2019 , 128 sayfa

Radar tespit performansını iyileştirmek için, radar sinyali içindeki istenmeyen sinyallerin tespiti kritiktir. Bu tez kapsamında, kargaşa yankılarının spektral moment tahmini için MUSIC, ESPRIT ve Burg gibi parametrik spektrum tahmin teknikleri değerlendirilmiştir. İncelenen tekniklerin hiçbiri Doppler yayılımını tahmin edemediği ve yeterli tahmin doğruluğuna sahip olmadığı için, en iyi performansa sahip optimizasyon ve çizgi arama algoritması ile çalışan Stokastik Maksimum İhtimal (SML) metodu uygulanmıştır. SML tahmin doğruluğu başlangıç noktasına çok bağlı olduğu ve hesaplama olarak pahalı olduğu için, yinelemeli çalışan özgün bir tahmin metodu (Turbo) önerilmiştir. Önerilen Turbo metodu yüksek Doppler çözünürlüğü, doğruluk değeri ve hesaplama kolaylığı ile literatürde önerilen metotlardan çok daha iyi performans göstermiştir. Buna ek olarak, Burg tahminleri başlangıç noktası seçiminde kullanılarak, önerilen Turbo metodu en uygun hale getirilmiştir. En uyguna yakın tahmin metodu tasarımından sonra, tahmin edilen parametreler, az sayıda ikincil veri ile bile çıkışında sinyalin girişim ve gürültü toplamına oranının maksimum normalize değerini elde edebilen tespit filtresi tasarımında kullanılmıştır. Son olarak, kargaşa sı-

nıflandırılması için problem özel olarak yapay sinir ağı mimarisi tasarlanmıştır. Önerilen sinir ağı performansı, özgün Turbo metodu tahminleri ile değerlendirilmiştir.

Anahtar Kelimeler: radar sinyal işleme, kargaşa güç spektrumu, parametre tahmini, sinirsel ağlar, maksimum olasılık tahmini, uyarlanabilir detektörler

To my lovely mother

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my supervisor Assist. Prof. Dr. Gökhan Muzaffer GÜVENSEN for his precious guidance, criticism, encouragement and insight throughout the development of this thesis. The door of Assist. Prof. Dr. GÜVENSEN's office was always open whenever I had a question about my research. I would also like to thank Prof. Dr. Yalçın TANIK, for his valuable suggestions, comments and support. The technical discussions with them are gratefully appreciated.

My sincere thanks also go to Dr. Ülkü DOYURAN, who provided me an opportunity to join her team as a Radar System Engineer, and who gave access to the data and research facilities in ASELSAN.

I am also grateful to the members of the committee, Prof. Dr. Sencer KOÇ, Prof. Dr. Çağatay CANDAN, Prof. Dr. Orhan ARIKAN, for examining the thesis.

My special thanks also go to Prof. Dr. İsmet ERKMEN and Prof. Dr. Aydan ERKMEN who set great role models for me, and provide endless guidance throughout Bachelor and Master studies.

Finally, I want to express my very profound gratitude to my family for providing me limitless support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. Especially, the words are not enough to express my appreciation to my mother who supports me to follow my dreams and be the best version of myself.

Thank you.

TABLE OF CONTENTS

ABSTRACT	v
ÖZ	vii
ACKNOWLEDGMENTS	x
TABLE OF CONTENTS	xi
LIST OF TABLES	xviii
LIST OF FIGURES	xx
LIST OF ABBREVIATIONS	xxiv
Nomenclature	xxvi
CHAPTERS	
1 INTRODUCTION	1
1.1 Motivation and Problem Definition	2
1.2 Literature Review	3
1.2.1 Parameter Estimation	4
1.2.2 Radar Detectors	6
1.2.3 Classification Using Neural Networks	8
1.3 Contributions and Novelties	9
1.4 The Outline of the Thesis	10
1.4.1 The Notation of the Thesis	10

2	RADAR PRINCIPLES	13
2.1	Introduction to Radar Operation	13
2.2	Radar Signals	17
2.2.1	Target Signal	17
2.2.1.1	Swerling 0	17
2.2.1.2	Swerling I	18
2.2.1.3	Swerling II	18
2.2.1.4	Swerling III	18
2.2.1.5	Swerling IV	18
2.2.2	Clutter Signal	19
2.2.3	Noise Signal	19
2.3	Radar Detection	19
2.3.1	Data Matrix	20
2.4	Spectral Analysis of Clutter	21
2.4.1	Covariance Data Matrix Model	23
2.5	Clutter Types	24
2.5.1	Ground (Land) Clutter	24
2.5.2	Sea Clutter	25
2.5.3	Rain Clutter	26
2.5.4	Bird Clutter	26
3	CLUTTER PROFILE PARAMETER ESTIMATION METHODS	29
3.1	Conventional Approaches (Center Velocity Estimation Methods)	30
3.1.1	MUSIC (MUltiple Signal Classification)	31

3.1.2	ESPRIT (Estimation of Signal Parameters via Rotational Invariance Technique)	31
3.1.3	Burg with Maximum Entropy Method	32
3.1.3.1	Autoregressive (AR) Signals	32
3.1.3.2	Levinson Durbin and Maximum Entropy Method	34
3.2	Maximum Likelihood Based Technique (Stochastic Maximum Likelihood)	39
3.3	Proposed Approach	42
3.3.1	Stochastic Maximum Likelihood with Turbo Approach	42
3.4	Test Results	44
3.4.1	Distant Clutters	45
3.4.1.1	Number of Turbo Iterations	45
3.4.1.2	Performances of Algorithms	46
	MSE and CRB Comparison of Methods with Different CNR Pairs	47
	Estimated PSD Comparisons of Methods with Different CNR Pairs	49
	Estimated PSD Comparisons of Methods with Different PRFs	50
3.4.2	Closely Spaced Clutters	51
3.4.2.1	Number of Turbo Iterations	52
3.4.2.2	Performances of Algorithms	52
	MSE and CRB Comparison of Methods with Different CNR Pairs	53
	Estimated PSD Comparisons of Methods with Different CNR Pairs	55
	Estimated PSD Comparisons of Methods with Different PRFs	56

4	RADAR DETECTION BASED ON ESTIMATED CLUTTER POWER PROFILE	59
4.1	Radar Detection	60
4.1.1	Signal Model	60
4.2	Conventional Detectors for Clutter Suppression and Doppler Processing	61
4.2.1	Moving Target Indicator (MTI) and Moving Target Detection (MTD) Algorithms	62
4.2.2	Optimal Clutter Suppression Filter	63
4.3	Proposed Detector	66
4.4	Simulations	67
4.4.1	Distant Clutters	68
4.4.1.1	The Importance of Spread Knowledge on Detector Performance	68
4.4.1.2	Number of Turbo Iterations	69
4.4.1.3	Performances of Algorithms When Target Signal Contamination is Absent	70
4.4.1.4	Performance of the Designed Algorithm with Target Signal Contamination Presence	72
4.4.2	Closely Spaced Clutters	73
4.4.2.1	The Importance of Spread Knowledge on Detector Performance	73
4.4.2.2	Number of Turbo Iterations	74
4.4.2.3	Performances of Algorithms When Target Signal Contamination is Absent	75
4.4.2.4	Performance of the Designed Algorithm with Target Signal Contamination Presence	77
5	CLUTTER CLASSIFICATION METHODS	79

5.1	Artificial Intelligence Background	80
5.1.1	Artificial Neural Networks	80
	Perceptron	81
	Activation Functions and Nonlinearity	81
	Sigmoid Function	82
	Feed-Forward Neural Network (Multilayer Perceptron)	82
	Feature Extraction	83
	Training	84
	Curse of Dimensionality	84
	Dimensionality Reduction	85
	Loss Functions	85
	Minimizing Loss	86
	Regularization	87
5.2	Proposed Neural Network	87
5.2.1	Pre-Processing and Feature Extraction	89
	Phase Variations of Sample Sequence as Features	90
	Magnitude Variations of Sample Sequence as Features	90
5.2.2	Neural Network Architecture	92
5.3	Experimental Results	94
5.3.1	Train the Network	95
5.3.2	Test Neural Network	96
	5.3.2.1 Some Possible Problems	96
	Vanishing Gradient Problem	96

Exploiting Gradients	97
5.3.2.2 Hyper-parameter Selection	97
Effect of Training Data Size	97
Effect of Number of Hidden Layers and Nodes	97
Effect of Activation function	98
Effect of Learning Rate	98
Effect of Mini batch size	99
Number of epochs	99
Effect of Weight Initialization	100
Effect of Regularization	100
Effect of Distinction of Classes	100
5.3.2.3 Test Scenarios	100
Three Moving Object	101
Calm Day	101
Moderate day, light wind	102
Rough/Stormy day	102
Two Objects	103
Classification Tests after Turbo Parameter Estimation Methods	104
Distant Clutters	105
Closely Spaced Clutters	105
6 CONCLUSIONS	107
6.1 Future Works	109
REFERENCES	111

APPENDICES

A	APPENDIX	117
A.1	Derivatives of \mathbf{R}_x	117
A.2	Line Search and Optimization Method Selection	118
A.2.1	Line Search Methods	118
A.2.1.1	Uniform Search	118
A.2.1.2	Dichotomous Search	119
A.2.1.3	Fibonacci Search	119
A.2.1.4	Golden Section Search	120
A.2.1.5	Three Point Interval	121
A.2.1.6	Quadratic Search	121
A.2.1.7	Parabolic Fit	122
A.2.2	Optimization Algorithms	123
A.2.2.1	Steepest Descent	123
A.2.2.2	Newton-Raphson Method	124
A.2.2.3	Newton with Steepest Descent	125
A.2.2.4	Modified Newton Method (Rank 1 Correction)	126
A.2.2.5	Davidon Fletcher Powell Method	126
A.2.2.6	Fletcher-Reeves Method	127
A.3	Cramer Rao Lower Bound Calculation	128

LIST OF TABLES

TABLES

Table 1.1	Notation	11
Table 2.1	Douglas Sea State [1]	25
Table 5.1	Statistical Parameters	94
Table 5.2	Train and Test Accuracies for Two Scenarios	100
Table 5.3	Calm Day Scenario	101
Table 5.4	Calm Day Test Results	101
Table 5.5	Moderate Day Scenario	102
Table 5.6	Moderate Day Test Results	102
Table 5.7	Rough Day Scenario	102
Table 5.8	Rough Day Test Results	103
Table 5.9	One Clutter in Each Dataset	103
Table 5.10	One Clutter Test Results	103
Table 5.11	One or Two Clutters in Each Dataset	104
Table 5.12	Classification Test Results	104
Table 5.13	Distant Clutters Spectrum Classification	105
Table 5.14	Distant Clutters Test Results	105

Table 5.15 Closely Spaced Clutters Spectrum Classification	106
Table 5.16 Closely Spaced Clutters Test Results	106
Table A.1 Line Search Methods Test Results	123
Table A.2 Optimization Algorithms Test Results	128

LIST OF FIGURES

FIGURES

Figure 1.1	Clutter Identification	3
Figure 2.1	A Pulse Train	14
Figure 2.2	Data Matrix	21
Figure 2.3	Generic Doppler spectrum of a received echo signal	22
Figure 2.4	Generic Doppler spectrum of two clutters	23
Figure 3.1	Lattice-equivalent model of PEF of order M	36
Figure 3.2	Center velocity estimation performance of algorithms (a) rain velocity estimate error (b) sea velocity estimate error for the case $v_{rain} = 26.178m/sec$, $v_{sea} = 3.872m/sec$, $CNR_{rain} = CNR_{sea} = 50dB$, $\sigma_{v_{sea}} = 1.512m/sec$, $\sigma_{v_{rain}} = 3.108m/sec$	39
Figure 3.3	Turbo Method Flow Diagram	42
Figure 3.4	Averaged PSD estimates with different number of Turbo iterations (a) $CNR_{rain} = CNR_{sea} = 50dB$ (b) $CNR_{rain} = 40dB$ $CNR_{sea} = 50dB$ (c) $CNR_{rain} = 30dB$ $CNR_{sea} = 50dB$	46
Figure 3.5	MSE vs number of snapshots curves for parameters (a)rain velocity estimate (b)rain spread estimate (c)sea velocity estimate (d)sea spread estimate	47

Figure 3.6	MSE vs number of snapshots curves for parameters (a)rain velocity estimate (b)rain spread estimate (c)sea velocity estimate (d)sea spread estimate	48
Figure 3.7	MSE vs number of snapshots curves for parameters (a)rain velocity estimate (b)rain spread estimate (c)sea velocity estimate (d)sea spread estimate	49
Figure 3.8	Averaged PSD estimates of estimation methods (a) $CNR_{rain} = CNR_{sea} = 50\text{dB}$ (b) $CNR_{rain} = 40\text{dB}$ $CNR_{sea} = 50\text{dB}$ (c) $CNR_{rain} = 30\text{dB}$ $CNR_{sea} = 50\text{dB}$	50
Figure 3.9	Averaged PSD estimates at different PRF values (a) $CNR_{rain} = CNR_{sea} = 50\text{dB}$ (b) $CNR_{rain} = 40\text{dB}$ $CNR_{sea} = 50\text{dB}$ (c) $CNR_{rain} = 30\text{dB}$ $CNR_{sea} = 50\text{dB}$	51
Figure 3.10	Averaged PSD estimates with different number of turbo iterations (a) $CNR_{rain} = CNR_{sea} = 50\text{dB}$ (b) $CNR_{rain} = 40\text{dB}$ $CNR_{sea} = 50\text{dB}$ (c) $CNR_{rain} = 30\text{dB}$ $CNR_{sea} = 50\text{dB}$	52
Figure 3.11	MSE vs number of snapshots curves for parameters (a)rain velocity estimate (b)rain spread estimate (c)sea velocity estimate (d)sea spread estimate	53
Figure 3.12	MSE vs number of snapshots curves for parameters (a)rain velocity estimate (b)rain spread estimate (c)sea velocity estimate (d)sea spread estimate	54
Figure 3.13	MSE vs number of snapshots curves for parameters (a)rain velocity estimate (b)rain spread estimate (c)sea velocity estimate (d)sea spread estimate	55
Figure 3.14	Averaged PSD estimates of estimation algorithms (a) $CNR_{rain} = CNR_{sea} = 50\text{dB}$ (b) $CNR_{rain} = 40\text{dB}$ $CNR_{sea} = 50\text{dB}$ (c) $CNR_{rain} = 30\text{dB}$ $CNR_{sea} = 50\text{dB}$	56

Figure 3.15	Averaged PSD estimates at different PRF values (a) $CNR_{rain} = CNR_{sea} = 50\text{dB}$ (b) $CNR_{rain} = 40\text{dB}$ $CNR_{sea} = 50\text{dB}$ (c) $CNR_{rain} = 30\text{dB}$ $CNR_{sea} = 50\text{dB}$	57
Figure 4.1	Normalized Output SINR for different spread values assumptions (a) $CNR_{rain} = CNR_{sea} = 50\text{dB}$ (b) $CNR_{rain} = 40\text{dB}$ $CNR_{sea} = 50\text{dB}$ (c) $CNR_{rain} = 30\text{dB}$ $CNR_{sea} = 50\text{dB}$	69
Figure 4.2	Normalized Output SINR of different number of Turbo iterations (a) $CNR_{rain} = CNR_{sea} = 50\text{dB}$ (b) $CNR_{rain} = 40\text{dB}$ $CNR_{sea} = 50\text{dB}$ (c) $CNR_{rain} = 30\text{dB}$ $CNR_{sea} = 50\text{dB}$	70
Figure 4.3	Normalized Output SINR for estimation methods (a) $CNR_{rain} = CNR_{sea} = 50\text{dB}$ (b) $CNR_{rain} = 40\text{dB}$ $CNR_{sea} = 50\text{dB}$ (c) $CNR_{rain} = 30\text{dB}$ $CNR_{sea} = 50\text{dB}$	71
Figure 4.4	Normalized Output SINR with target contamination (a) $CNR_{rain} = CNR_{sea} = 50\text{dB}$ (b) $CNR_{rain} = 40\text{dB}$ $CNR_{sea} = 50\text{dB}$ (c) $CNR_{rain} = 30\text{dB}$ $CNR_{sea} = 50\text{dB}$	72
Figure 4.5	Normalized Output SINR for different spread values assumptions (a) $CNR_{rain} = CNR_{sea} = 50\text{dB}$ (b) $CNR_{rain} = 40\text{dB}$ $CNR_{sea} = 50\text{dB}$ (c) $CNR_{rain} = 30\text{dB}$ $CNR_{sea} = 50\text{dB}$	74
Figure 4.6	Normalized Output SINR of different number of Turbo iterations (a) $CNR_{rain} = CNR_{sea} = 50\text{dB}$ (b) $CNR_{rain} = 40\text{dB}$ $CNR_{sea} = 50\text{dB}$ (c) $CNR_{rain} = 30\text{dB}$ $CNR_{sea} = 50\text{dB}$	75
Figure 4.7	Normalized Output SINR for estimation methods (a) $CNR_{rain} = CNR_{sea} = 50\text{dB}$ (b) $CNR_{rain} = 40\text{dB}$ $CNR_{sea} = 50\text{dB}$ (c) $CNR_{rain} = 30\text{dB}$ $CNR_{sea} = 50\text{dB}$	76
Figure 4.8	Normalized Output SINR with target contamination (a) $CNR_{rain} = CNR_{sea} = 50\text{dB}$ (b) $CNR_{rain} = 40\text{dB}$ $CNR_{sea} = 50\text{dB}$ (c) $CNR_{rain} = 30\text{dB}$ $CNR_{sea} = 50\text{dB}$	77

Figure 5.1	Perceptron with n inputs, one output and an activation function	81
Figure 5.2	Sigmoid Function	82
Figure 5.3	A Feed-forward Neural Network with one hidden layer	83
Figure 5.4	Classification System Design Cycle	89
Figure 5.5	Classifier of Radar Signals	89
Figure 5.6	Designed Feed-forward Neural Network structure	93
Figure 5.7	Training accuracy	98
Figure 5.8	Learning rate	99
Figure A.1	Uniform Search	118
Figure A.2	Dichotomous Search	119
Figure A.3	Fibonacci Search	120
Figure A.4	Golden Search	121
Figure A.5	Three Point Search	121
Figure A.6	Quadratic Search	122
Figure A.7	Steepest Descent	124

LIST OF ABBREVIATIONS

ACF	Autocorrelation Function
AR	Autoregressive
BW	Bandwidth
CDF	Cumulative Distribution Function
CLT	Central Limit Theorem
CNR	Clutter-to-Noise Ratio
CPI	Coherent Processing Interval
CRB	Cramer Rao Bound
CUT	Cell Under Test
DTFT	Discrete Time Fourier Transform
FFT	Fast Fourier Transform
MC	Monte Carlo
MEM	Maximum Entropy Method
METU	Middle East Technical University
MLE	Maximum Likelihood Estimation
MSE	Mean Square Error
MTI	Moving Target Indicator
MUSIC	Multiple Signal Classification
pdf	Probability Density Function
P_d	Probability of Detection
P_{fa}	Probability of False Alarm
PRI	Pulse Repetition Interval
PRF	Pulse Repetition Frequency
PSD	Power Spectral Density

PW	Pulse Width
RADAR	RAdio Detection And Ranging
RCS	Radar Cross Section
RMS	Root Mean Square
SIR	Signal to Interference Ratio
SINR	Signal to Interference plus Noise Ratio
SML	Stochastic Maximum Likelihood
SNR	Signal-to-Noise Ratio
WSS	Wide Sense Stationary

Nomenclature

$\boldsymbol{\mu}$	Estimated Parameter Vector
∇	Gradient Operator
$\hat{\boldsymbol{\mu}}$	Parameter Vector
λ	Wavelength (m)
ρ_i	ith Reflection Coefficient
σ	Radar Cross Section (m^2)
σ_{v_i}	Clutter Spread (m/sec)
$\sigma_{v_{rain}}$	Rain Clutter Spread (m/sec)
$\sigma_{v_{sea}}$	Sea Clutter Spread (m/sec)
\mathbf{H}	Hessian Matrix
c	Speed of Light (m/sec)
CNR_i	Clutter to Noise Ratio of ith Clutter (dB)
f_i	ith Clutter Doppler Frequency (Hz)
f_{op}	Radar Operational Frequency (Hz)
$L(\boldsymbol{\mu})$	Negative Log-likelihood
P_i	Power Value of ith Clutter (W)
v_{rain}	Rain Clutter Velocity (m/sec)
v_{sea}	Sea Clutter Velocity (m/sec)

CHAPTER 1

INTRODUCTION

RADAR is an acronym standing for RAdio Detection And Ranging. It is a system, aiming the detection of objects. Detection is accomplished by transmitting electromagnetic waves with the help of an antenna and analyzing the echoes reflected from objects in environment to extract information about ones of interest, targets. In other words, it is an electromagnetic remote sensing instrument, which can be used for detecting and tracking the targets. Basically, an electromagnetic wave is transmitted to illuminate a volume of space and the echo coming from the target is extracted to determine its direction, distance to the system or speed. The return signal is composed of some components such as the direct path return of the target, multipath returns, echoes reflected from other objects, thermal noise, and jammer. Anything except the target echo can be called as an interference signal or unwanted radar return. In short, interference signals can be listed as noise, jammer and clutter.

First of all, external noise and internal noise are two types of noise signals, must be suppressed. The external noise is received through the radar antenna. Generally, its source is solar activity. On the other hand, the intrinsic noise is caused by the hardware of radar. It is also called as thermal noise. Noise term generally stands for the internal one since it is more dominant than the external one. In most of the radar systems, noise is modelled as a zero-mean, white, Gaussian process [2].

Second type interference signal is clutter which differs from noise by its different correlation properties. Moreover, clutter is a type of echo; hence its power depends on the operational radar parameters, whereas noise power is modelled independently. The clutter signal model will be investigated in following chapters.

Thirdly, as an offensive Electronic countermeasure, jamming techniques are used to degrade the ability of radar during detection of targets. The difference between jam-

mer and clutter signal comes from the type of interference signal. In other words, the clutter causes a passive interference, while the jammer signal is an active one.

It is useful to note that, the definition of target and clutter is application dependent. In other words, it differs with the aimed application area of the radar. For instance, clouds are considered as clutter signals for air traffic radars, while as targets for weather radars. Similarly, synthetic aperture radars' target is the surface of earth, yet it is the clutter of surveillance radars.

The clutter echo is an unwanted band-limited signal which has to be suppressed for operation. Thus, it is desired to obtain the characteristic of clutter accurately in order to increase the performance during detection of targets. For a successful characterization of clutter, its spectral and statistical properties must be under consideration. Because of that, clutter Power Spectrum Density (PSD) will be estimated by using various methods. A suitable spectrum modelling is utilized such that moments of PSD enables full characterization. In addition to the estimation, the effect of clutter power spectrum parameter estimation on radar detection performance will be studied with the utilization of an adaptive parametric filter. Finally, estimated parameters will be used to determine the type of clutter. A classification approach will be implemented with features, obtained from estimated spectral parameters.

1.1 Motivation and Problem Definition

The aim of designing most of the modern radar systems is to optimize the detection performance. Various effective signal processing techniques have been proposed to suppress the clutter. Successful suppression of clutter signals will increase the signal power to interference power ratio. If the radar system knows the clutter characteristics such that it processes the radar echoes based on that, characterization accuracy of detection will significantly improve. Therefore, clutter spectral profile identification is crucial.

In the case of fixed ground radars, working in a relatively stationary environment, the clutter characteristic will be predictable and easy to obtain. However, radars may be located over moving platforms. Additionally, if the environmental conditions are not predictable, the stored information of the working medium will be useless. For ex-

ample, detection of targets on a rainy day will be troublesome because of the abrupt changes in rain characteristics with varying velocity vector direction of wind or the rate of rainfall or the turbulence strength etc. Another example is that both the rain and sea clutter may coexist. It is impossible to model or predict such changeable media. Therefore, the radar should adapt itself to the instantly changing clutter environments in order to achieve solid elimination of unwanted echoes. By that, instantaneous characterization of the environment must be accomplished with estimating of clutter parameters adaptively. Real time environment identification is critical if high performance target detection is required in a difficult operation environment.

The main motivation of the thesis is to evaluate the performance of recently proposed clutter spectral parameter estimation techniques and improve their performance with novel approaches. After accomplishing optimum estimation, the estimation effect on radar detection performance will be measured with an adaptive filter. Lastly, classification of the owner or owners of the received signal components is aimed.

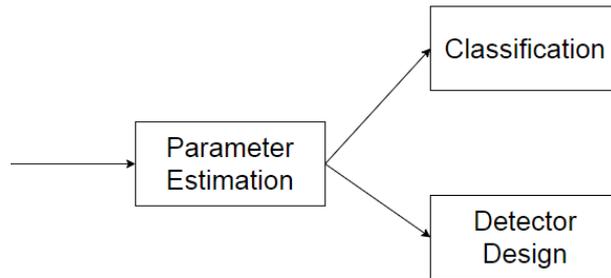


Figure 1.1: Clutter Identification

1.2 Literature Review

Clutter power spectrum estimation and classification is the main research theme, aiming to increase radar detection performance. A detailed literature survey has been specifically directed to review the strengths and weaknesses of existing algorithms.

1.2.1 Parameter Estimation

Spectral moment estimation techniques can be used in many areas. In weather radars [3], spectral moment estimation methods are used to decrease required data size for identification in a stationary environment. In [4], optimal range information is determined by estimation. As an application area, different from radar, the estimation is used to increase resolution for accurate diagnostic criteria selection. Radio astronomy uses telescopes instead of antenna arrays to determine a 'target's location [5]. Thus, estimation methods become important like as in radar applications. Despite the fact that each of the listed areas has different aims, the common goal is to determine the moments with a less complex algorithm giving adequately accurate results.

Clutter spectral moment estimation becomes a popular area when radar detection performance became a concern. A robust algorithm is desired to obtain optimum detection performance. In literature, some of non-parametric and parametric methods, suggested to estimate moments, are examined. Non-parametric methods are preferred because of their low computational complexity. For example, Pulse Pair (PP) method [6] and periodogram based [3] ones are most widely used ones. Their computational load is relatively light. However, they have some disadvantages such as increased bias with increase in spectral width. Moreover, their performances are poor compared with parametric ones [7]. Moreover, in [8], it is shown that PP will give erroneous estimates in case of signals having low Doppler values, i.e. near to zero. It will not give results as precise as ones of Maximum Likelihood estimation method [9]. Actually, all of the drawbacks are resulted from trade-off between computational complexity and accuracy of estimation algorithms. High accuracy is required for clutter identification, so non-parametric algorithms are not investigated in the scope of this thesis.

Since, the parametrization of PSD models constrains the search space; parametric methods have higher accuracy than non-parametric methods. The main goal is to determine the first three spectral moments for Doppler radars. Zeroth moment of PSD is related to the power of received echo. The first moment gives the information about mean Doppler frequency. The second moment represents spectrum standard deviation, Doppler spread value. In other words, spectral width is expressed as the standard deviation of the Doppler frequencies. First spectral moment estimation is crucial for pulsed-Doppler data processing since it is related with Doppler frequency. Some sug-

gested methods in literature are studied to estimate Doppler frequency. First of all, two subspace based techniques, MUSIC and ESPRIT, are implemented. Then, Burg with Maximum Entropy Method is studied and tested.

Subspace based methods for frequency estimation is first suggested by Schmidt in 1986 [10]. They are chosen because of high resolution capacity. Firstly, MUSIC is examined. In [10], it is claimed that MUSIC is asymptotically efficient if the number of signals in received echo is one. If received signal consists of more than one echo, it may give biased estimates. It is claimed that MUSIC is also applicable for narrow-band signals. Similarly, in [11], superposition of two close spectrums is investigated to measure sensitivity of MUSIC. The accuracy is dependent on the test scenario, yet it claims that it is possible to estimate two parameters with high accuracy. However, it also notes that MUSIC needs high number of snapshots. If number of snapshots is not enough, it may fail even if in case of small modelling errors. The ESPRIT is another proposed method, used for parameter estimation. In [12], MUSIC and ESPRIT algorithms are compared with respect to estimation accuracy. It is shown that MUSIC has better performance than ESPRIT. However, ESPRIT is used as an alternative to MUSIC since while MUSIC algorithm requires search, but ESPRIT needs no search to estimate Doppler frequencies. In [13], it is shown that ESPRIT can reduce computational and storage cost. Moreover, it is claimed that ESPRIT is less sensitive to the array imperfections of antenna than MUSIC. However, ESPRIT gives biased results. Same trade-off between computational complexity and accuracy occurs.

As a final method to estimate frequency, Burg with Maximum Entropy Method (MEM) is investigated. An autoregressive (AR) spectrum can be computed using Burg estimation or solving Yule Walker equations. MEM using autoregressive signal model is proposed by Burg [14]. AR modelling has great spectral resolution. In [15], a comparison is studied between Fast Fourier transform (FFT) methods and AR spectrum estimation methods. It is shown that AR is more consistent than FFT. Moreover, FFT method accuracies are dependent on available data duration which is an impediment for real time estimation.

When spectrum has non-zero spread, frequency estimation is also studied by adding a multiplicative noise to the signal. Many efficient algorithms are suggested using this model, yet none of them is able to give information about the spread value.

Maximum Likelihood (ML) is investigated to estimate spread value. ML solutions are

generally dependent on non-linear equations. In literature, ML approach for parameter estimation has appeared in two versions referred to as Stochastic (SML) [16] and Deterministic Maximum Likelihood (DML) estimation [17]. Çırpan's work [18] is a useful source to compare performances of SML and DML. In his work, both of the methods are developed and iterative solutions are proposed. By interpreting founding of the paper, it is decided to use SML instead of focusing on DML. Huge variety of proposed approaches in the field of ML spectral-moments estimation is studied such as [3] and [19]. They both show that parametric ML outperformed the non-parametric classical approaches. In [20], Levin used ML to estimate the mean Doppler frequency and spectral width. However, an assumption is made which data recording is much longer than correlation time of the process. Additionally, Maximum Likelihood estimator is studied over weather spectral parameters in [21] which was computationally too heavy for implementing in real time operation. Moreover, it was not optimal in accuracy. The milestone of the research was investigating the paper of Boyer [22] in which a SML technique is implemented to obtain accuracy at high resolution. A second order optimization algorithm starting from actual value was suggested in the paper in order to estimate all spectral parameters with same algorithm. This approach is not robust to the selected initial values for each moment of the clutter spectrum. Moreover, an algorithm, including all moments, requires tremendous computational effort. Thus, a novel algorithm must be designed and tested.

Note that, parameters of test scenarios are generated by using values in literature. In [23], clutter resulted from windblown trees is studied, supported by a comparison with the previous studies. This study also gives some information about spread and velocity values of clutters.

1.2.2 Radar Detectors

In order to measure the effect of clutter PSD estimation on radar detection performance. Some common detectors in literature are investigated. The first work about adaptive detectors was belonging to Kelly, published in 1986 [24]. In addition to detection range bin (Cell Under Test i.e CUT), Kelly proposed to use secondary range cells during detection hypothesis choice. He derived corresponding false alarm and detection probabilities P_{fa} , P_d and generalized likelihood ratio test (GLRT) analytic

expressions. Target is assumed to be located in 1 dimensional subspace of signal model. As a following work, Kelly and Forsythe [25] generalized the method for systems having multi-channels. Similar formulations are used to design various detectors. For example, Adaptive Matched Filter (AMF) is suggested in [26], which assumes CUT data is Gaussian distributed. Thus, CUT data covariance matrix is required during detector design. The secondary cell covariance matrices are also used during hypothesis tests. The adequate number of secondary cells is required for successful detection. Reed-Mallet [27] derived a rule to measure the effect of secondary cell numbers on output Signal to Noise Ratio (SNR). They proposed a limit for number of observations. If output SNR loss is desired to be 3 dB, at least two times of covariance matrix size must be investigated. The effect of power difference between CUT and clutter signals are investigated by [28] and [29]. Adaptive Normalized Matched Filter (ANMF) is suggested as detector. ANMF assumes secondary cells and CUT have same normalized covariance matrix, while clutter signals in secondary cells and CUT have power values modelled with deterministic unknown variables. In [30], a sensitive detector is tried to be designed by suggesting that an additional signal exists in CUT, belonging neither noise nor target. The approach is useful in order to detect closely spaced multiple targets. In [31], clutter signal covariance matrix is assumed to be modelled as an AR process in order to design the adaptive detector. In [32], previously suggested detectors are examined in a class named as “invariant detectors”. By using covariance matrix model, various detectors are designed. An additional detector design is suggested and its performance is measured in [33] and [34], when non-homogeneity exists between CUT and secondary cells. In order to decrease P_{fa} , the importance of covariance matrix estimation is stressed in [18], such that a recursive covariance matrix estimation method is investigated. The effect of estimation method is measured with real data. The results were not promising because of non-stationarity of clutter in time and spatial domain.

Adaptive detectors, mentioned above are Sample Matrix Inversion (SMI) based techniques necessitating large number of secondary cells, multiples of autocorrelation matrix size. In this thesis, instead of using SMI adaptive detectors, a parametric filter is adopted by using proper parameters of clutters, obtained via spectral estimation method. In other words, Wiener filter approach is employed which will be adequate

with the small number of secondary cells to achieve genie-aided detector which will be used as a performance benchmark. The estimated covariance matrix is used for filter design which is explained in Chapter 4. Thus, the high number of secondary cell requirement is significantly diminished with the use of parametric spectrum estimation methods.

1.2.3 Classification Using Neural Networks

Many application areas using classification with the help of Artificial Intelligence (AI) become popular in recent years such as image processing, speech and character recognition and forecasting. Specifically for radars, AI is commonly used for target classification. As a good example, target classification with neural network is investigated in [35].

In the scope of this work, a classifier is aimed to be designed which will work with the proposed clutter parameter estimator. This is a unique approach introduced in the area of clutter classification.

In [36], comprehensive background information is given about pattern classification. It is shown that the most of the successful classification methods depends on Neural Networks (NN). Neural networks are compatible with real world problems since they are developed to provide real-time response with high-performance. As suggested, appropriate features must be selected. The features are selected by using [37]. Haykin's work was a useful source for data generation. However, the suggested neural network architecture was not satisfying when thinking neural network designs used today for classification. In [38], it is suggested that fuzzy logic and Bayesian classification can also be used, but their results are not consistent. A number of performance comparisons between neural and conventional classifiers have been made by many studies [39]. Neural networks outperformed classical approaches. As a similar work, [40] is examined which performs classification by using features (entropy, skewness and kurtosis) of different clutter models with multilayer perceptron. Features selected in Haykin's paper were more comprehensive than in [40]. In [41] clutter classification is performed with multi-segment Burg algorithm with K distribution model. The paper was similar to our work, yet shape parameter estimation is not adequately ap-

appropriate approach.

All in all, designing an optimal radar clutter classifier, which can also work with an estimator, is a gap. Thus, this thesis tries to fill the gap.

1.3 Contributions and Novelties

In this thesis, we will design a favourable estimator, which forecasts the moments of radar echo Power Spectrum Density. After that, detector performance with estimated parameters will be observed by implementing an adaptive filter. Finally, type of estimated signals will be determined by a classifier using Artificial Neural Networks.

Our contributions are as follows:

- The most encountered clutter types are explained with related power spectrum parameters.
- The comparison between widely used Doppler frequency estimation methods is performed in order to choose the most efficient one, used for initial point selection.
- The performances of various optimization and line search methods are measured to find the optimal one working with SML method suggested in [22].
- A novel estimator is designed by implementing a Turbo approach which estimates first and second moments of spectrum recursively.
- The detector design with estimated parameters is performed to observe the effect of parameter estimation over detection performance of radar.
- The proposed detector achieves the performance of genie-aided detector (with perfect knowledge of the spectrum) even with a small number of secondary data which is crucial for non-stationary environments.
- A neural network structure is designed and used to determine the types of clutters from estimated parameters.

1.4 The Outline of the Thesis

In Chapter 2, radar operation is explained briefly. Encountered signal types are investigated with related spectral properties. After that, the focus is on the clutter spectrum characteristics which will be used in estimation. Finally, most common clutter types are given.

In Chapter 3, various methods aiming to estimate power spectrum density parameters are examined. First of all, definition and spectral parameters of some clutter types are stated. After that, subspace based methods and AR signal model with Maximum Entropy Method are represented which are used to estimate center Doppler frequencies. As another approach, Stochastic Maximum Likelihood (SML) performance is evaluated. Moreover, its output accuracy is increased with the help of examining various optimization algorithms and line search methods. Additionally, Turbo estimator architecture is designed in order to increase accuracy while decreasing computational cost. Turbo is an algorithm integrated with SML technique.

Chapter 4 covers the analyses of detection performance improvement after determining the clutter spectral parameters. Received signal is processed such that detector performances with experimental and analytical results are compared.

In Chapter 5, classification of clutter signals from spectral parameters by using deep learning concept, neural networks, is illustrated with compressive background information, steps of designing the network and test results.

Finally, in Chapter 6, the conclusions are given and the possible future works are stated.

1.4.1 The Notation of the Thesis

The notation used in this thesis is summarized below,

Table 1.1: Notation

	General Notation
a, b, \dots, z	Scalar quantities
$\mathbf{a}, \mathbf{b}, \dots, \mathbf{z}$	m dimensional column or row vectors
$\mathbf{A}, \mathbf{B}, \dots, \mathbf{Z}$	$m \times n$ dimensional matrices
$\mathbf{a}^T, \mathbf{A}^T$	Transpose of vector \mathbf{a} , transpose of matrix \mathbf{A}
$\mathbf{a}^H, \mathbf{A}^H$	Hermitian of vector \mathbf{a} , Hermitian of matrix \mathbf{A}
$[\mathbf{A}]_{(i,j)}$	Entry of matrix \mathbf{A} located at i th row and j th column
a_i	i th entry of vector \mathbf{a}

CHAPTER 2

RADAR PRINCIPLES

As mentioned, the most basic aim of radar is to extract necessary information from received signal which is reflection of transmitted electromagnetic waves from objects in the environment. The necessary information definition depends on application area of radar. Mainly, location (distance to the radar), velocity with travel direction, and elevation (altitude with respect to radar system) of both stationary and non-stationary objects must be extracted for full identification of environment. In this work, target and interference signals are under-interest.

In Section 2, firstly, radar operation is briefly explained; spectral analysis of clutter is introduced with examples on well-known clutter types.

2.1 Introduction to Radar Operation

The idea of radar comes from nature, the echolocation animals such as bats. Radar uses electromagnetic waves for detection, finding location and speed measurement of anything under interest. Basically, it transmits electromagnetic waves into space and collects reflected echoes from environment. If the transmitting and receiving of signals are performed by same antenna, the system is called as *mono-static*. If they are located at different places, the system is a *bi-static* radar system. In this work, mono-static radar system is investigated.

The key goal of radar operation is to distinguish the signal as the one reflected from target or only interference. In most of the radars, detection is done with cancellation of clutter and thresholding the output. Thresholding implies comparing the amplitude of processed signal with an adaptively computed threshold value, which generally

depends on the received echo amplitude.

It is easy to deduce that the range of a target can be interpreted from time delay between transmitted signal and received echo. Thus, for monostatic radar, formulation of target range is

$$R = \frac{c \cdot t_D}{2} \quad (2.1.1)$$

Backscattered signals are processed digitally. Thus, received signal will be sampled after filtering and demodulation such that samples forming range bins will have size:

$$\Delta R = \frac{c \cdot t_s}{2} \quad (2.1.2)$$

in which t_s is sampling period which is one over sampling rate (frequency), $\frac{1}{f_s}$. Sampling frequency must be chosen such that samples will be independent.

Radar transmits modulated carrier signals. The most common one is a pulse train, pulses repeated at intervals. In other words, the electromagnetic waves are emitted in short bursts. It is shown in Figure 2.1.

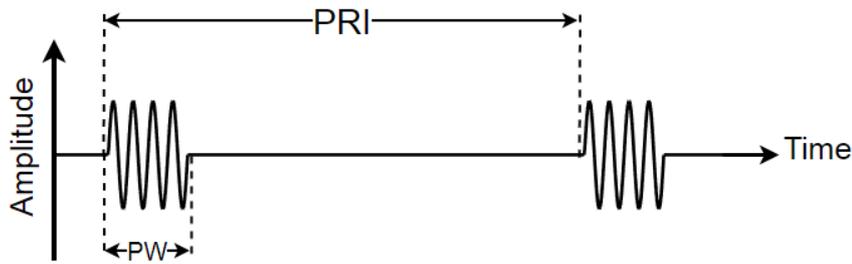


Figure 2.1: A Pulse Train

Pulse width of the transmitted pulse, PW , is the duration of pulse in one Pulse Repetition Interval (PRI). PRI is the interval between pulses. Inverse of PRI is referred as Pulse Repetition Frequency (PRF) which is the number of transmitted pulses per second. Dwell time is defined as time spent on target. Thus, in short, coherent trains of P pulse are transmitted from radar antenna. After receiving echo reflected from objects in environment, radar properly demodulates filters and samples the incoming narrowband waveform.

Width of the transmitted pulse, PW , determines the range resolution of radar which becomes important when the detection of close targets will be performed. Range resolution is the minimum distance between two close targets, required to distinguish

them. It is calculated in 2.1.3 in which bandwidth is denoted by BW . Thus, range resolution may be different than range cell size.

$$\Delta R_{res} = \frac{c \cdot PW}{2} = \frac{c}{2 \cdot BW} \quad (2.1.3)$$

Coherency in radars defined as constant phase relationships of transmitted signal and reference signal. Coherency in radars enables to notice even small phase shifts in received echo. Phase shift is used to calculate Doppler frequency related with velocity of target. Coherent Processing Interval (CPI) is defined as the total time of sampling a pre-specified group of pulses having same PRF and operational frequency. Thus, if P is number of transmitted pulses for accurate detection, CPI can be formulated as;

$$CPI = P \cdot PRI \quad (2.1.4)$$

The radial velocity of target can be interpreted from Doppler shift in received echo which has carrier (operating) frequency f_0 . For stationary monostatic radar suppose that the target is moving with a radial velocity v toward the radar. Doppler shift for received electromagnetic waves is given in 2.1.5. The received frequency will be

$$f_R = \left(\frac{1 + \frac{v}{c}}{1 - \frac{v}{c}} \right) f_0 \quad (2.1.5)$$

Thus, if target moves towards the antenna received frequency will increase. If it recedes, a decrease will occur. Since target will have much lower velocity than speed of light. The previous equation can be simplified to,

$$f_R = \left(1 + 2\frac{v}{c} + 2\left(\frac{v}{c}\right)^2 + \dots \right) f_0 \quad (2.1.6)$$

If we get rid of all second and higher order terms of $\frac{v}{c}$,

$$f_R \cong \left(1 + 2\frac{v}{c} \right) f_0 \quad (2.1.7)$$

Thus, Doppler shift or Doppler frequency, f_d , is defined as the difference between transmitted and received echo signal frequencies can be written as,

$$f_d = 2\frac{v}{c} f_0 \quad (2.1.8)$$

Equivalently,

$$v = \frac{c \cdot f_d}{2f_0} \quad (2.1.9)$$

Doppler frequency of a target, f_d ; must be found with spectrum estimation techniques. Some targets will not be visible to the radar because of the velocity values. This velocity is called as the *blind speed*. It depends on the operating frequency and Pulse Repetition Frequency (PRF) of the radar unit. If PRI of received signal is same or multiple of transmitted signal PRI, zero signals will be obtained by radar.

$$v_{blind} = \frac{c}{2 \cdot PRI \cdot f_0} \quad (2.1.10)$$

If P_t stands for the transmitted power from a directional antenna with gain G_t , then power density at a distance R can be defined as;

$$P_{dt} = \frac{P_t G_t}{4\pi R^2} \quad (2.1.11)$$

After transmit, reflected echo power to the antenna surface is dependent on target Radar Cross Section (RCS), σ , such that

$$P_{de} = \frac{P_t G_t}{4\pi R^2} \frac{\sigma}{4\pi R^2} \quad (2.1.12)$$

The total amount of received power by the antenna, P_r , depends on affective aperture A_e of antenna, which is defined as

$$A_e = \frac{G_r}{4\pi \lambda^2} \quad (2.1.13)$$

in which G_r is receiver antenna gain and λ is the wavelength, depending on operation (carrier) frequency, f_0 . Moreover, received power will be affected by some losses such as system losses and atmospheric attenuation. If the total loss can be represented by L , the relation between received echo and transmitted powers can be written by radar equation;

$$P_r = \frac{P_t \cdot G_t \cdot G_r \cdot \lambda^2 \cdot \sigma}{(4\pi)^3 \cdot R^4 \cdot L} \quad (2.1.14)$$

Another form of 2.1.14 can be used to calculate maximum range of the target.

$$R = \left\{ \frac{P_t \cdot G_t \cdot G_r \cdot \lambda^2 \cdot \sigma}{(4\pi)^3 \cdot P_r \cdot L} \right\}^{1/4} \quad (2.1.15)$$

Radar detection performance is evaluated by some criteria such as probabilities of detection (P_d) and false alarm (P_{fa}). They usually depend on Signal to Interference plus Noise Ratio (SINR), which is the power ratio of the useful and unwanted signals.

$$SINR = \frac{P_{signal}}{P_{interference} + P_{noise}} \quad (2.1.16)$$

Similarly, Clutter to Noise Ratio is defined as

$$CNR = \frac{P_{clutter}}{P_{noise}} \quad (2.1.17)$$

Finally, Signal to Noise Ratio is

$$SNR = \frac{P_{signal}}{P_{noise}} \quad (2.1.18)$$

2.2 Radar Signals

Received echo contains signals reflected from targets, clutter, thermal noise and other types of interference. Identification of each signal is substantial during radar operation.

2.2.1 Target Signal

As it is derived, the target echo signal power is related with target radar cross section (RCS). RCS of a target generally depends on angle, frequency and polarization. Thus, fluctuations will occur in RCS value. Swerling models are used to model RCS fluctuations [2]. Swerling models depends on a probability density function and a decorrelation time for the target RCS. Scan to scan decorrelation implies pulses collected on one sweep have the same complex amplitude value, whereas the pulses from next scan will have another constant value, independent from the previous one. On the other hand, pulse-to-pulse decorrelation implies each pulse in one scan will take an independent value for the RCS from other pulses.

2.2.1.1 Swerling 0

It is an ideal target model, having constant return. In other words, its RCS is fixed during operation.

2.2.1.2 Swerling I

Signal has decorrelation from scan to scan. The radar cross-section is constant from pulse-to-pulse. Thus, its RCS is permanent over one antenna scan. Entire pulse train will have the constant amplitude, which is a single random variable with a Rayleigh pdf.

$$P(\sigma) = \frac{1}{\sigma_{avg}} \exp\left(-\frac{\sigma}{\sigma_{avg}}\right) \quad (2.2.1)$$

2.2.1.3 Swerling II

Fluctuations occur from pulse to pulse. Similarly with the Swerling I model, the amplitude of each pulse in the train is a statistically independent random variable with the a Rayleigh pdf. The variations are faster than ones of Swerling I target model.

Swerling I and II are generally used to model targets composed of many independent scatterers with similar area values like airplanes.

2.2.1.4 Swerling III

Similar to the Swerling I model, each pulse of a train will have the same amplitude. Yet, pulse amplitude along train is assumed to be a random variable with a one-dominant-plus-Rayleigh,

$$P(\sigma) = \frac{4\sigma}{\sigma_{avg}^2} \exp\left(-2\frac{\sigma}{\sigma_{avg}}\right) \quad (2.2.2)$$

2.2.1.5 Swerling IV

Swerling IV target model resembles to Swerling III, but RCS variations occur from pulse to pulse rather than from scan to scan

2.2.2 Clutter Signal

Clutter modelling becomes more challenging with a high-resolution radar at a low incident angle. Since it is a random signal, clutter can be defined by means of its spatial and temporal statistical characteristics. Radar clutter can be modelled with a Gaussian speckle, \mathbf{g} ; modulated by a texture parameter, τ ;

$$\mathbf{c} = \sqrt{\tau}\mathbf{g} \quad (2.2.3)$$

In the scope of this work, it is assumed that texture parameter is constant. Thus, clutter signal is purely Gaussian. This assumption is reasonable since clutter component in received echo has a large number of scatterers, Central Limit Theorem can be used to claim that clutter returns model is a multivariate Gaussian. Power spectrum of clutter represents amplitude variations of the signal received from a given range bin.

2.2.3 Noise Signal

The thermal noise in the system is generally assumed as an independent and identically distributed complex white Gaussian random vector. Thus, it will be zero mean and has diagonal autocorrelation matrix shown below;

$$\underline{\mathbf{u}} = \underline{\mathbf{u}}_I + j\underline{\mathbf{u}}_Q \quad (2.2.4)$$

$$E\{\underline{\mathbf{u}}\underline{\mathbf{u}}^H\} = \sigma_n^2\mathbf{I} \quad (2.2.5)$$

$\underline{\mathbf{u}}$ is the complex noise vector with zero mean in phase and quadrature components; $\underline{\mathbf{u}}_I, \underline{\mathbf{u}}_Q \cdot \sigma_n^2$ stands for noise variance.

2.3 Radar Detection

For most of the radars, classical detection process starts with clutter suppression followed by Doppler processing and thresholding. Clutter suppression may be included in Doppler processing or performed apart from it. In modern radar systems, interference suppression, Doppler processing and thresholding operations are performed

simultaneously. Along each range bin, received pulses are processed. Pulse samples are separated by a Pulse Repetition Interval (PRI).

Several clutter suppression methods can be implemented such as Moving Target Indication (MTI) canceller. They are inefficient for some scenarios such as in case of slow targets etc. Moreover, Fast-Fourier Transformation (FFT) is used to analyze spectra and extract Doppler frequency information. FFT has low resolution such that contamination of target signals by the other targets having neighboring Doppler frequencies may occur. Windowing can be used to prevent that, but it decreases SNR and expands the main lobe of beam which is undesired for radar operation. Thus, a more effective processing approach must be employed which is particularly investigated in Chapter 4. After processing, thresholding is applied. The threshold dependence on the received echo is resulted from the aim of keeping the false alarm rate constant. Because of that the operation is called as Constant False Alarm Rate (CFAR) method [42] which is also explained at length in Chapter 4.

2.3.1 Data Matrix

Throughout radar signal processing, in order to make it easier, a well-defined data structure is necessary. The structure must be created such that it will enable spectral and statistical operations over data. Depending on the data acquisition scenario, dimensions of the structure can be changed. It is assumed that the angular position of the antenna remains constant in the data acquisition system. Moreover, in the scope of this work, only the pulse number and the delay will be under interest. Thus, the obtained data will be a matrix which is shown in Figure 2.2.

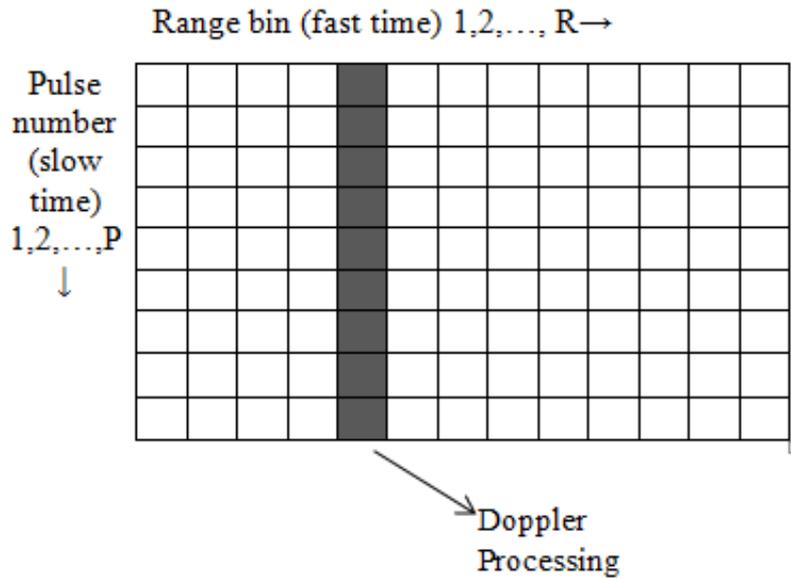


Figure 2.2: Data Matrix

In which P is the number of transmitted train pulses. PRI is separated into R successive range cells (range bins or delay bins). Each range cell represents a delay values. Therefore, the return signal can be sampled and stored in a $P \times R$ data matrix structure. Rows represent the samples at a fixed PRI. On the other hand, columns stand for the samples taken from successive pulses after a fixed delay time. Doppler processing is performed over each range bin along slow time.

2.4 Spectral Analysis of Clutter

The spectral analysis of clutter is of great importance for adaptive radar operation. In the Figure 2.3, a sample representation of spectral contents for targets, noise and clutter is illustrated. As it can be predicted, the zero-Doppler bin consists of the returns from stationary objects. Moving ones are located in the spectrum according to their relative radial velocity with respect to the radar. Additionally, noise is spread uniformly over the whole spectrum. Finally on Figure 2.3, the clutter usually occupies a region near to the zero-Doppler bin. Non-zero Doppler values of clutter spectrum are due to the intrinsic motion of clutter sources.

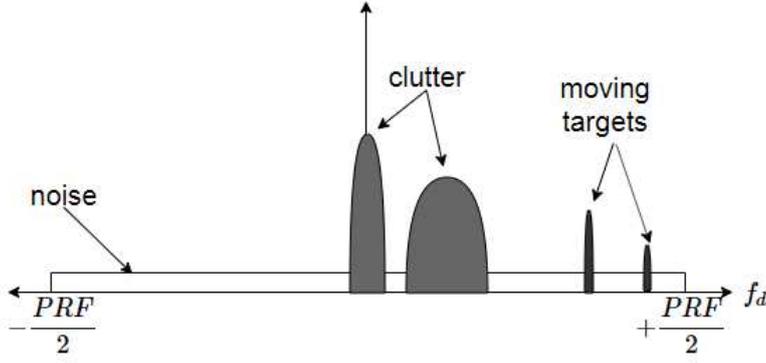


Figure 2.3: Generic Doppler spectrum of a received echo signal

From this spectrum, it can be seen that for a target which is located outside the clutter region, the only interference will be the thermal noise. On the other hand, targets having low velocity values will be dominated by clutter signals. If the information about clutter spectral characteristics is inadequate, the performance of radars will be significantly degraded. Spectral characteristics of clutter components can be estimated from Power Spectral Density of the slow time samples at a specific range bin shown in Figure 2.2.

During modelling and estimation of clutter Power Spectrum Density two assumptions are suggested.

- A1) *Power spectrum is formed by N number of Gaussian echoes.*
- A2) *The numbers of clutters and targets in PSD are known.*

σ_n^2 stands for additive white Gaussian circular noise representing the radar receiver noise power spectral density. P_i 's are the mean power and f_i is the mean frequency of i^{th} clutter signal. Mean Doppler frequency can be calculated by using velocity in m/s with 2.4.1.

$$f_i = \frac{2 \cdot v_i}{\lambda} \quad (2.4.1)$$

Moreover, the spreads of the Doppler spectra of the clutters, σ_i^2 's, are found from spread of Doppler velocity spectrum by using 2.4.2.

$$\sigma_i = \frac{2 \cdot \sigma_{v_i}}{\lambda} \quad (2.4.2)$$

Thus, by using A1 and A2, clutter spectra are typically approximated by Gaussian shaped Power Spectrum Density. This may often be a computational convenience rather than a realistic modelling. All in all, power spectrum of received signal $P_s(f)$ can be written as it was in [22];

$$P_s(f) = \left(\sum_{i=1}^N S_i(f) + \sigma_n^2 \right) |\beta|^2 \quad \text{with} \quad S_i(f) = \frac{P_i}{\sqrt{2\pi\sigma_i^2}} e^{-\frac{1}{2}\left(\frac{f-f_i}{\sigma_i}\right)^2} \quad (2.4.3)$$

with asymptotic autocorrelation sequence

$$\gamma(t) = \sum_{i=1}^N P_i e^{-2\pi^2\sigma_i^2 t^2 + j2\pi f_i t} + \sigma_n^2 |\delta(t)| \quad (2.4.4)$$

A sample power spectrum is shown in Figure 2.4

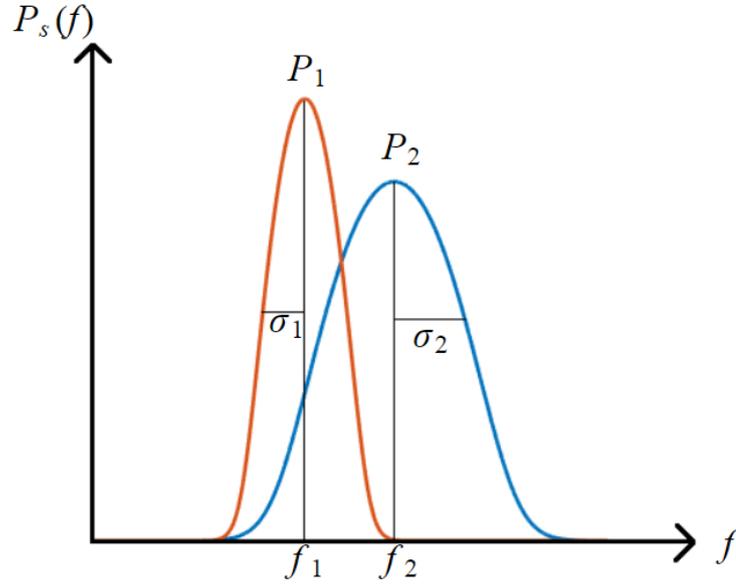


Figure 2.4: Generic Doppler spectrum of two clutters

The clutter covariance matrix determines the PSD of clutter.

2.4.1 Covariance Data Matrix Model

Received signal is $\mathbf{x}(k) = [x_1(k) \dots x_P(k)]^T$ where $0 \leq k \leq K$. K stands for the number of snapshots, secondary range cells, to the pre-specified range bin and P is the pulse number. Index k can be dropped for convenience, so that received signal

is $\mathbf{x} = \mathbf{y} + \mathbf{n}$. The vector \mathbf{x} is a stochastic Gaussian vector, having zero mean and covariance, \mathbf{R}_x .

For each clutter, steering matrix can be expressed as;

$$\mathbf{A}(w_i) = \text{diag}\left[1 \quad e^{jw_iPRI} \quad e^{j2w_iPRI} \quad \dots \quad e^{j(P-1)w_iPRI}\right] \quad (2.4.5)$$

which only depends on the mean Doppler frequency of signal components. Spectral width related component is written as;

$$[\mathbf{B}]_{(k,l)} = e^{-2\pi^2\sigma_i^2(k-l)^2PRI^2} \quad (2.4.6)$$

Thus, each clutter will have covariance

$$\mathbf{R}_{y_i}(P_i, w_i, \sigma_i^2) = P_i\mathbf{A}(w_i)\mathbf{B}(\sigma_i^2)\mathbf{A}^*(w_i) \quad (2.4.7)$$

If more than one clutter signal is received, a composite clutter case, a superposition approach can be applied. If N is the number of clutter signals;

$$\mathbf{R}_y = \sum_{i=1}^N \mathbf{R}_{y_i}(P_i, w_i, \sigma_i^2) \quad (2.4.8)$$

Noise and clutter signals are uncorrelated such that \mathbf{R}_x can be written as the summation of signal and noise covariance matrices.

$$\mathbf{R}_x = \mathbf{R}_y + \sigma_n^2\mathbf{I} \quad (2.4.9)$$

As it can be seen \mathbf{R}_x is full rank.

2.5 Clutter Types

Each of the different types of radar clutter possesses distinct spectral characteristics, which can be used to determine the type of clutter. In this part, some common clutter types are explained with corresponding spectral characteristics.

2.5.1 Ground (Land) Clutter

The modelling of ground clutter is really complex. If man-made and natural discrete scatterers on the ground are not included, the clutter will always be stationary, having

zero Doppler. Thus, ground clutter can have a strictly narrow spectrum, centralized at zero Doppler. In other words, it has impulse like spectrum. By that way, the land clutter can be classified and detected by using Doppler information. Its spectrum has lower standard deviation value compared to other clutter types (approx.: 0-1 m/s). Because of the fact that it can be identified from zero center Doppler value, ground clutter is not included during analysis of scenarios having multiple clutters in one range cell.

2.5.2 Sea Clutter

Sea clutter effect generally depends on particular characteristics of measurement environment, such as sea waves, or wind speed. They have significant effect on characteristics of sea clutter spectrum. Sea clutter mean Doppler frequency is determined by the orbital velocity of wind driven waves. Useful empirical models are studied in literature such as Rayleigh, K-Distributed. However, it is adequate to treat it as if it is Gaussian distributed. Sea clutter is dependent on the waves and state of sea. Douglas Sea State, given in 2.1, is used to determine approximate velocity intervals, spread and CNR values of sea clutter for tests. Note that, 1 knot is approximately 0.5144 m/sec.

Table 2.1: Douglas Sea State [1]

Sea State	Description	Wind Speed (kts)	Sea Wave Height (ft)
1	Smooth	0 - 6	0 - 1
2	Slight	6 - 12	1 - 3
3	Moderate	12 - 15	3 - 5
4	Rough	15 - 20	5 - 8
5	Very rough	20 - 25	8 - 12
6	High	25 - 30	12 - 20
7	Very high	30 - 50	20 - 40
8	Precipitous	> 50	> 40

2.5.3 Rain Clutter

Unlike the spiky characteristic of sea clutter returns, rain echoes response is flat. Rain, similarly hail or snow, generates clutter signal which can be characterized by having continuous return over long ranges and at wide angles. It is a typical volume clutter. Thus, its spectrum has a higher spread than sea clutter, with a possibly larger shift in the center frequency. It has high possibility to mask possible targets. In reality, as weather condition become rough, sea clutter spectrum must also be affected. However, the exact relation between sea and rain cannot be foreseen. Luckily, we do not have to know the exact relation in the scope of this work, they assumed to be independent signals.

2.5.4 Bird Clutter

Clutter caused by birds is difficult to eliminate because of its target like characteristic. It is a type of point clutter. Because birds can fly at up to approximately 25 m/s, their returns are not rejected by Doppler or MTI processing. They are treated as moving point targets. Mean radar cross section of a bird is small, but a group of bird returns can fluctuate up to a high level (aircraft). It confuses the radar while detecting targets with low cross sections. In addition to that, intrinsic motion of the scatters will be relatively heavy because of that the amount of spectral spread is high. Unlike water droplets of rain clutter, moving passively with the wind, birds are individually powered scatterers moving in different directions with various speeds. Thus, bird clutter effect must be under consideration during design.

To summarize, it is seen that most common forms of clutters have more or less definite spectral characteristics which can be summarized as:

- Ground clutter spectrum is remarkably narrow and located at zero Doppler frequency
- Sea clutter has a spread spectrum with a possibly moderate shift in the center frequency
- Rain clutter has a wider spread, with a possibly higher shift in the center fre-

quency than sea clutter

- Bird clutter has target like spectrum with a higher spread resulted from large amount of scatterer intrinsic motion

It is also desired to identify target echoes. Despite the fact that radar target definition depends on the application; in general, it is assumed that targets will be characterized by having impulse like spectrum located at center Doppler frequency ranges higher than ones of clutters. However, other cases are also investigated such as targets having velocities close to the ones of clutter.

CHAPTER 3

CLUTTER PROFILE PARAMETER ESTIMATION METHODS

Estimation is the process of deducing the value of a quantity of interest using some noisy measurements or observations. As stated in Chapter 1, clutter signals are those received from undesired scatterers, defined depending on the aim of application.

In literature, non-parametric methods are used to estimate spectral characteristics of signals. They are derived from the Power Spectral Density (PSD) definitions of signals. Some examples are periodogram and correlogram, which will not be examined in the scope of this work. Because of the fact that, non-parametric techniques have moderate frequency resolution, as complexity of spectrum increases, they will not be able to separate the different signal contributions to the spectrum. For example, in case of overlapped echoes having a big difference between amplitudes, non-parametric methods have difficulty to distinguish signal components. Thus, various parametric algorithms are implemented instead of non-parametric ones with the hope of finding the most efficient one.

In parametric estimation methods, PSD is assumed to be fit to a certain functional form. Parametric estimation methods are also known as the model based methods such that spectral estimation is implemented to determine parameters of the signal model as accurately as possible. Parameters of signal model are defined as moments of PSD. Zeroth moment of PSD is related to the power of received echo. The first moment gives the information about mean Doppler frequency. The second moment represents spectrum standard deviation, Doppler spread value. It is important to note that, 1st and 2nd moments of the signals are fixed during operation time. In other words, clutter signals are assumed to be Wide Sense Stationary (WSS).

First moment of spectrum is estimated using model based methods. First implemented one is MUSIC algorithm, which is based on a parametrization of time series

autocorrelation function. Secondly, ESPRIT algorithm is tested. MUSIC and ESPRIT give poor results when spectral spreads are moderate. Thus, as an alternative method Burg algorithm is implemented. It also gives information about zeroth moment. Burg showed the most satisfying first moment estimation performance. However, none of these methods has the ability to provide second moment knowledge. Thus, Stochastic Maximum Likelihood is implemented which is proposed in [22]. The suggested algorithm accuracy was highly initial point dependent. If the initial parameter values are not selected close to the real ones, the algorithm gives erroneous estimates. It is also computationally heavy because it tries to find all three moments of each component at the same time. An well-suited optimization method must be determined which gives more flexibility while searching the optimal point. Thus, various optimization algorithms and line search methods are employed to get rid of initial value dependence and increase the accuracy of estimation. In addition to that, the number of estimated parameters is decreased by using information obtained from the other parametric moment estimation methods. Moreover, accuracy is improved by choosing initial values with the help of previously mentioned estimation methods.

In Chapter 3, implemented conventional parametric estimation methods are illustrated with related signal models. Proposed novel algorithm working principle and implementation steps are explained. Finally, the most substantial test results with related comments are represented at the end of the chapter.

3.1 Conventional Approaches (Center Velocity Estimation Methods)

The classical frequency estimation techniques are generally known as time domain ones, based on auto-correlation or cross-correlation of signals. MUSIC and ESPRIT are two parametric ones. They depend on the assumption that number of tones in a measurement is known. Note that, MUSIC and ESPRIT are only applicable to narrowband signals. For the case of interest, sources are independent and noise signals are uncorrelated. Additionally, as stated covariance matrix of received samples, \mathbf{R}_x , is full rank. First of all, as a parameter estimation algorithm, MUSIC will be investigated. It will be followed by ESPRIT and Burg methods to estimate Doppler frequencies of clutters.

3.1.1 MUSIC (Multiple Signal Classification)

MUSIC is a high resolution parameter estimation method, enabling velocity estimation of multiple signals of interests, target, clutter, etc. It has high accuracy. Moreover, it can simultaneously estimate multiple frequencies. It is also applicable to short data circumstances. It is designed by improving Pisarenko harmonic method estimator [43]. At core, MUSIC algorithm aims to obtain characteristic decomposition from autocorrelation of received signal. First of all, sample covariance matrix is computed from received echo, and its eigen decomposition is performed;

$$\mathbf{R}_x = \mathbf{U} \cdot \mathbf{D} \cdot \mathbf{V}' \quad (3.1.1)$$

After that, ordered eigenvalues of sample covariance matrix are extracted.

$$\mathbf{U} = [\mathbf{u}_1 \ \mathbf{u}_2 \ \dots \ \mathbf{u}_n] \quad (3.1.2)$$

MUSIC algorithm decomposes observation space into two sub-spaces, signal and noise. They are orthogonal to each other. If r is number of estimated frequencies, null space of autocorrelation Hermitian is spanned by columns of \mathbf{G} such that;

$$\mathbf{G} = [\mathbf{u}_{r+1} \ \mathbf{u}_{r+2} \ \dots \ \mathbf{u}_n] \quad (3.1.3)$$

Frequency estimates are interpreted from angular positions of roots of equation 3.1.4. The columns of \mathbf{G} belong to null space[43].

$$\mathbf{a}^T(z^{-1}) \cdot \mathbf{G} \cdot \mathbf{G}^* \cdot \mathbf{a}(z) = 0 \quad (3.1.4)$$

Roots nearest and inside to unit circle are found. The angular positions of roots are used to estimate Doppler frequencies, i.e. velocities. MUSIC is computationally expensive, but it shows strong center velocity estimation performance in case of small standard deviations. When spread is relatively large, it may fail. As expected, it generally gives superior results when CNR increases.

3.1.2 ESPRIT (Estimation of Signal Parameters via Rotational Invariance Technique)

Since, MUSIC is computationally heavy, ESPRIT method is implemented which takes much less computational effort. Its lower calculation load is due to leaving

out a search over all values. In ESPRIT, similar to MUSIC, firstly sample covariance matrix is calculated. After eigen decomposition; matrix, having columns as signal space eigenvectors, is found.

$$\mathbf{R}_x = \mathbf{U} \cdot \mathbf{D} \cdot \mathbf{V}' \quad (3.1.5)$$

Shift invariability between the discrete time series causes rotational invariance between the corresponding signal subspaces. The shift invariance, illustrated below, is the basis of ESPRIT method. Rotation will be achieved by using matrices 3.1.6;

$$\mathbf{\Gamma}_1 = [\mathbf{I}_{m-1} \quad \mathbf{0}] \quad \mathbf{\Gamma}_2 = [\mathbf{0} \quad \mathbf{I}_{m-1}] \quad (3.1.6)$$

Two subspaces spanned by eigenvectors are defined as

$$\mathbf{S}_1 = \mathbf{\Gamma}_1 \mathbf{U} \quad \mathbf{S}_2 = \mathbf{\Gamma}_2 \mathbf{U} \quad (3.1.7)$$

By using relation of rotation operation with a non-singular matrix ϕ

$$\mathbf{S}_1 \phi = \mathbf{S}_2 \quad (3.1.8)$$

in which ESPRIT method estimates the frequencies from eigenvalues of estimated ϕ ;

$$\hat{\phi} = \hat{\mathbf{S}}_2 \hat{\mathbf{S}}_1^{-1} \quad (3.1.9)$$

The negative of angles of eigenvalues give information about Doppler frequencies of scatterers.

3.1.3 Burg with Maximum Entropy Method

Starting point is the idea claiming that radar clutter signals can be modelled with a low order Autoregressive (AR) process. Well-known Burg estimator fits an autoregressive model to the input data with the help of Burg method. Fitting operation is a constraint optimization problem of a Prediction Error Filter model. The aim is to minimize the forward and backward prediction errors while AR parameters fulfill the Levinson-Durbin recursion [44].

3.1.3.1 Autoregressive (AR) Signals

The autoregressive model output variable is determined recursively. In other words, its value depends on linear combinations of its own previous values and on a stochas-

tic term [45]. AR signals are obtained by filtering white noise with a filter. The filter must have no zeros and p poles. The filter given 3.1.10

$$H(z) = \frac{B(z)}{A(z)} = \frac{b_0}{1 + \sum_{k=1}^p a_k z^{-k}} \quad (3.1.10)$$

The parameters of AR signal can be estimated by solving a set of linear equations, called as Yule-Walker equation set. The Yule-Walker relations give the relation between AR process auto-covariance and its parameters. The Yule-Walker equation set is shown in 3.1.11 in which σ_n^2 stands for noise variance;

$$r(k) = \begin{cases} -\sum_{l=1}^p a_l r(k-l) + \sigma_n^2 |b_0|^2 & , \text{if } k = 0 \\ -\sum_{l=1}^p a_l r(k-l) & , \text{if } k > 0 \end{cases} \quad (3.1.11)$$

The same equation can be written in matrix form shown as follows;

$$\begin{bmatrix} r(0) & r(-1) & \dots & r(-p) \\ r(-1) & r(0) & \dots & r(-p+1) \\ \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ r(p) & r(p-1) & \dots & r(0) \end{bmatrix} \times \begin{bmatrix} 1 \\ a_1 \\ \cdot \\ \cdot \\ \cdot \\ a_p \end{bmatrix} = \sigma_n^2 |b_0|^2 \begin{bmatrix} 1 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix} \quad (3.1.12)$$

Auto-covariance function is Hermitian symmetric such that the negative lags can be replaced by their positive lag counterparts. By using last p equations;

$$\begin{bmatrix} r(0) & r^*(1) & \dots & r^*(p-1) \\ r(1) & r(0) & \dots & r^*(p-2) \\ \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ r(p-1) & r(p-2) & \dots & r(0) \end{bmatrix} \times \begin{bmatrix} 1 \\ a_1 \\ \cdot \\ \cdot \\ \cdot \\ a_p \end{bmatrix} = - \begin{bmatrix} r(1) \\ r(2) \\ \cdot \\ \cdot \\ \cdot \\ r(p) \end{bmatrix} \quad (3.1.13)$$

3.1.12 and 3.1.13 can be combined into;

$$r(0) + \boldsymbol{\alpha}_p^T \mathbf{r}_p = \sigma_n^2 \mathbf{R}_p \boldsymbol{\alpha}_p = -\mathbf{r}_p \quad (3.1.14)$$

In 3.1.13, R_p represents a doubly symmetrical Toeplitz matrix with $p \times p$ dimensions. The parameter vector and auto covariance vector is defined without the first AR coefficient a_0 as

$$\boldsymbol{\alpha}_p = [a_1 \ a_2 \ \dots \ a_p] \quad (3.1.15)$$

$$\mathbf{r}_p = [r(1) \ r(2) \ \dots \ r(p)] \quad (3.1.16)$$

Thus, the mathematical solution can be found.

$$\boldsymbol{\alpha}_p = -\mathbf{R}_p^{-1} \mathbf{r}_p \quad (3.1.17)$$

A recursive algorithm is used to obtain the solution of 3.1.17. The algorithm is called as Levinson-Durbin algorithm.

3.1.3.2 Levinson Durbin and Maximum Entropy Method

Levinson Durbin recursion is employed in order to find poles of all pole infinite impulse response filters, shown in 3.1.10, used to obtain AR signals. MATLAB built-in algorithm uses pre-described deterministic autocorrelation sequence. Instead of using that, a new algorithm is written by referencing [46].

By using Shannon's information theory, information of random variables can be measured with entropy and mutual information. Entropy measures the uncertainty of random variables. Mutual information indicates how two variables are related.

In his work [46], Haykin uses Maximum Entropy Method (MEM) to increase resolution and stability of estimation from short radar echoes. MEM, as it can be understood from its name, aims to find the least constrained time series spectrum, related with the known values of autocorrelation function.

By using Wiener-Khinchin theorem, the power spectrum $S_x(f)$ and the autocorrelation function $R_x(m)$ relation can be written as in 3.1.18.

$$S_x(f) = \Delta t \sum_{m=-\infty}^{\infty} R_x(m) e^{-j2\pi m f \Delta t} \quad (3.1.18)$$

where Δt is the sampling period.

In information theoretic sense, it is the prediction of autocorrelation function maximizing the entropy of process.

$$Entropy(x) = - \sum_m p(x) \log_2(p(x)) \quad (3.1.19)$$

For Gaussian realizations entropy rate can be written as;

$$h = \frac{1}{4W} \int_{-W}^W \ln[S_x(f)]df \quad (3.1.20)$$

Assume that first $2M + 1$ values of the autocorrelation function, $R_x(m)$, are known. If so, the desired unknown autocorrelation values must be the ones adding no information to the process such that;

$$\frac{\partial h}{\partial R_x(m)} = 0, \quad |m| \geq M + 1 \quad (3.1.21)$$

Note that, M , order of filter, will be chosen as the number of clutters.

A Prediction Error Filter (PEF) can represent beneficial information from data with relatively less number of coefficients. After PEF obtained, it can be used to estimate missing data. If M is the maximum PEF order, by combining 3.1.18 and 3.1.21 spectrum can be estimated as;

$$\hat{S}_x(f) = \frac{P_M}{2W \left| \sum_{m=0}^M |a_{M,m} \exp(-2j\pi m f \Delta t)|^2 \right.} \quad (3.1.22)$$

where the $a_{M,m}$ are the coefficients of a prediction error filter (PEF) of order M , and P_M is the average value of the output power of the filter [46].

As it can be understood from its name; the prediction error is the difference between actual sample value $x(n)$ and its prediction $\hat{x}(n)$. If predictor uses the previously taken samples, it is called forward predictor. Forward prediction error is calculated as in 3.1.23. It is the difference between, $x(n)$ and its estimated value, calculated by using $x(n - 1), x(n - 2), \dots, x(n - M)$.

$$f_M(n) = x(n) - \sum_{k=0}^M a_{M,k} x(n - k), \quad n = M + 1, \dots, P \quad (3.1.23)$$

where $a_{M,0} = 1$ for all M . Note that, P is number of pulses.

Conversely, backward prediction is to estimate a sample by using values of next samples. Backward prediction error calculation is shown in 3.1.24. It represents the difference between actual $x(n - M)$ and its predicted value. Backward prediction of $x(n - M)$ is found by using samples $x(n - M + 1), x(n - M + 2), \dots, x(n)$.

$$b_M(n) = x(n - M) - \sum_{k=0}^M a_{M,k}^* x(n - M + k), \quad n = M + 1, \dots, P \quad (3.1.24)$$

Lattice predictors can combine forward and backward one with reflection coefficients ρ_i . Thus, PEF forward and backward procedure can be shown as a lattice filter [46] with cascaded M stages as in Figure 3.1

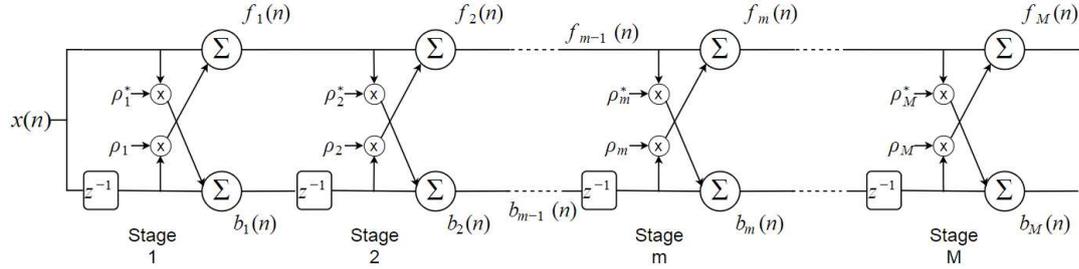


Figure 3.1: Lattice-equivalent model of PEF of order M

For stage $m + 1$ of this filter, ρ_m is called the reflection coefficient of stage m . We have

$$f_{m+1}(n) = f_m(n) + \rho_{m+1}b_m(n - 1) \quad (3.1.25)$$

$$b_{m+1}(n) = b_m(n) + \rho_{m+1}^*f_m(n) \quad (3.1.26)$$

where $n = m + 2, m + 3, \dots, P$, $m = 0, 1, 2, \dots, M - 1$. Moreover, all reflection coefficients must have magnitudes smaller than unity, so the lattice filter structure is stable.

Levinson Recursion comes to scene such that it will be used to compute the PEF coefficients from related set of reflection coefficients. The algorithm will take complex valued input signal $x(n)$, received echo. After that, it will produce outputs listed below.

- P_M : PEF output powers
- ρ : Reflection coefficients
- a_{Mm} : PEF coefficients (denominator pynomial)
- $f_M(n)$: Forward Prediction Error
- $b_M(n)$: Backward Prediction Error

n is the sample index.

$$a_{m+1,k} = a_{m,k} + \rho_{m+1} a_{m,m+1-k}^* \quad (3.1.27)$$

where $k = 0, 1, 2, \dots, m+1$ and $m = 0, 1, 2, \dots, M-1$, with

$$a_{m+1,k} = \begin{cases} 1 & , \text{ if } k = 0 \\ \rho_{m+1} & , \text{ if } k = m+1 \\ 0 & , \text{ if } k > m+1 \end{cases} \quad (3.1.28)$$

PEF output powers are calculated recursively.

$$P_{m+1} = (1 - |\rho_{m+1}|^2) P_m \text{ where } m = 0, 1, 2, \dots, M-1 \quad (3.1.29)$$

Reflection coefficients are needed which will be estimated from forward and backward prediction errors. Burg's method aims to diminish the sum of forward and backward squared prediction errors.

$$E^{m+1} = \sum_{n=m+2}^{P-1} (f_{m+1}(n))^2 (b_{m+1}(n))^2 \quad (3.1.30)$$

If 3.1.25 and 3.1.26 are substituted in 3.1.30.

$$E^{m+1} = \sum_{n=m+2}^{P-1} f_m(n) + \rho_{m+1} b_m(n-1)^2 b_m(n) + \rho_{m+1}^* f_m(n)^2 \quad (3.1.31)$$

If the derivative is taken with respect to ρ_{m+1} and equated to zero in order to find optimal reflection coefficients, the result will be

$$\hat{\rho}_{m+1} = \frac{-2 \sum_{n=m+2}^P f_m(n) * b_m(n-1)}{\sum_{n=m+2}^P [|f_m(n)|^2 + |b_m(n-1)|^2]} \quad (3.1.32)$$

The summary of algorithm:

1. Initial conditions:

$$f_0(n) = b_0(n) = x(n), \quad n = 1, 2, 3, \dots, P \quad (3.1.33)$$

$$P_0 = \frac{1}{N} \sum_{n=1}^N |x(n)|^2 \quad (3.1.34)$$

2. Put $m=0$. Compute estimate of 1st reflection coefficient:

$$\hat{\rho}_1 = \frac{-2 \sum_{n=2}^P x(n) * x(n-1)}{\sum_{n=m+2}^P [|x(n)|^2 + |x(n-1)|^2]} \quad (3.1.35)$$

$$P_1 = (1 - |\hat{\rho}_1|^2) P_0 \quad (3.1.36)$$

From the Levinson recursion, $a_{1,1} = 1 = \hat{\rho}_1$

3. Find the forward $f_1(n)$ and backward $b_1(n)$ prediction errors at the first output stage of the lattice filter:

$$f_{m+1}(n) = f_m(n) + \rho_{m+1} b_m(n-1) \quad (3.1.37)$$

$$b_{m+1}(n) = b_m(n) + \rho_{m+1}^* f_m(n-1) \quad (3.1.38)$$

4. Utilize Burg's formula in order to estimate stage 2 reflection coefficient. After that, PEF coefficients can be calculated.

$$a_{2,1} = a_{1,1} + \hat{\rho}_2 * a_{1,1} = \hat{\rho}_1 + \hat{\rho}_2 * \hat{\rho}_1 \quad (3.1.39)$$

$$a_{2,2} = \hat{\rho}_2 \quad (3.1.40)$$

$$P_2 = (1 - |\hat{\rho}_2|^2) P_1 \quad (3.1.41)$$

5. Evaluate forward prediction error $f_2(n)$ and backward prediction error $b_2(n)$ using $\hat{\rho}_2$. (Put $m=2$, find 3 etc.)

6. When the prescribed order M of the PEF is reached, the computation will be terminated.

7. Find roots of PEF coefficients. Angle of each root gives the phase change from one sample to the next in one PRI.

8. Estimate radial velocity of m th component

$$\hat{v}_m = \frac{c}{4\pi \cdot f_{op} \cdot PRI} \theta_m \quad (3.1.42)$$

9. The power of m th component is calculated from PEF output powers.

The algorithm computes estimates over each snapshot data. The final power and frequency estimation of Burg is selected as median of estimated powers and frequencies

of each snapshot, secondary range cell.

MUSIC, ESPRIT and Burg method performances are compared for 2 clutter case with respect to number of secondary range cells, snapshots. The result shown in Figure 3.2,

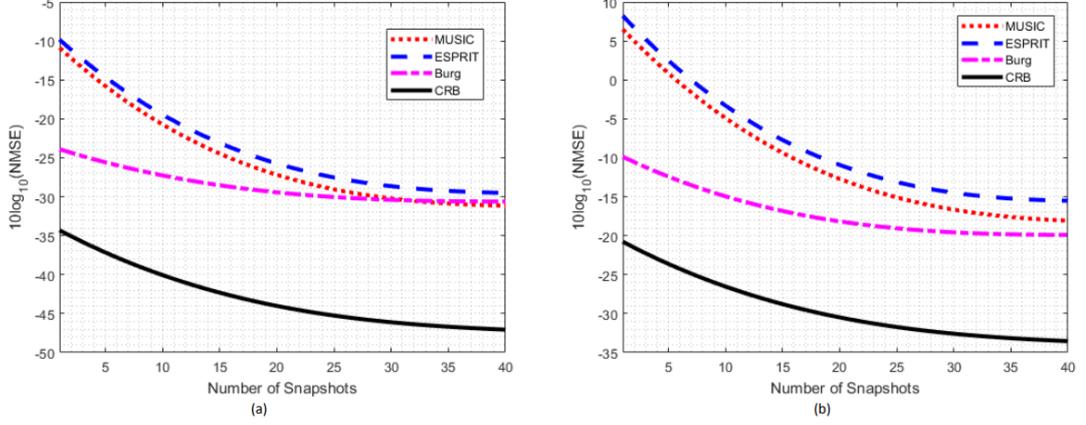


Figure 3.2: Center velocity estimation performance of algorithms (a) rain velocity estimate error (b) sea velocity estimate error for the case $v_{rain} = 26.178m/sec$, $v_{sea} = 3.872m/sec$, $CNR_{rain} = CNR_{sea} = 50dB$, $\sigma_{v_{sea}} = 1.512m/sec$, $\sigma_{v_{rain}} = 3.108m/sec$

Burg estimation has high accuracy on mean velocity estimation. Additionally, it gives estimated power values. However, frequency estimation accuracy is not close to Cramer Rao Lower Bound (CRB). Additionally, information quality about second power of the spectrum was poor. In other words, spread values cannot be estimated accurately. Since it depends on the intrinsic motion of the scatterers, the value of spread is a beneficial indication of the source of clutter. Thus, it must be estimated accurately.

3.2 Maximum Likelihood Based Technique (Stochastic Maximum Likelihood)

Maximum Likelihood (ML) estimation methods aim to find probability distribution, making observed data *most likely*. In other words, parameter vector that maximizes the likelihood function is looked for.

Some properties of ML estimator can be listed as [47]:

1. It is asymptotically unbiased.
2. It is asymptotically efficient. In other words, lowest variance of parameter estimations is achieved asymptotically.
3. It is asymptotically Gaussian with a mean and variance. The mean equals to the true value of the parameter to be estimated and the variance is given by the Cramer Rao Lower Bound (CRB).

Additionally, in case of adequate information in the measurements, Cramer Rao Bound will approach to zero such that the ML estimation variance will also converge to zero. This implies ML estimate will reach to the true value. It will be consistent in that case. Therefore, the same ML solution can be obtained independent of the parametrization used.

Since the aim is to determine PSD of clutter, parameter vector can be set as; $\boldsymbol{\mu} = [w_1 \ \sigma_1^2 \ P_1 \ \dots \ w_N \ \sigma_N^2 \ P_N \ \sigma_n^2]$. N stands for the number of clutters. w_i is the i 'th clutter angular Doppler frequency, related with first moment of spectrum. σ_i is i 'th clutter spread value, second moment of spectrum. P_i is zeroth moment of i 'th clutter PSD. Noise power σ_n^2 is assumed to be one. Thus, P_i 's become equal to CNR values of corresponding clutters.

Maximizing log-likelihood is same as minimizing negative log-likelihood [22], given in 3.2.1.

$$L(\boldsymbol{\mu}) = \log(|\mathbf{R}_x(\boldsymbol{\mu})|) + Tr\{\mathbf{R}_x^{-1}(\boldsymbol{\mu})\hat{\mathbf{R}}_x\} \quad (3.2.1)$$

where

$$\hat{\mathbf{R}}_x = \frac{1}{K} \sum_{k=1}^K \mathbf{x}(k)\mathbf{x}^H(k) \quad (3.2.2)$$

K is the number of snapshots, secondary range cells. $\hat{\mathbf{R}}_x$ is obtained from measurements, whereas $\mathbf{R}_x(\boldsymbol{\mu})$ is calculated using parameter vector $\boldsymbol{\mu}$. Using covariance data matrix model, explained before, $\mathbf{R}_x(\boldsymbol{\mu})$ can be found.

$$\mathbf{R}_x(\boldsymbol{\mu}) = \mathbf{R}_y(\boldsymbol{\mu}) + \sigma_n^2 \mathbf{I} \quad (3.2.3)$$

where,

$$\mathbf{R}_y(\boldsymbol{\mu}) = \sum_{i=1}^N \mathbf{R}_{y_i}(P_i, w_i, \sigma_i^2) \quad (3.2.4)$$

$\mathbf{R}_x(\boldsymbol{\mu})$ is invertible because it is the addition of $\sigma_n^2 \mathbf{I}$ to the parametrized matrix $\mathbf{R}_y(\boldsymbol{\mu})$. Since, only matrix inversion is applied on $\mathbf{R}_x(\boldsymbol{\mu})$, $\hat{\mathbf{R}}_x$ does not have to be full rank such that the expression in 3.2.1 works even with one snapshot. In order to minimize the value of Trace in 3.2.1, the best interference space is searched.

Optimum parameters will be found by solving multi-dimensional non-linear optimization problem such that;

$$\hat{\boldsymbol{\mu}} = \operatorname{argmin}\{L(\boldsymbol{\mu})\} \quad (3.2.5)$$

Since the likelihood expression is complicated, finding minimum of negative log-likelihood function analytically is impossible. Moreover, the search over all possible values will be time consuming. Thus, an iterative method to obtain an approximate solution must be found.

Iterative algorithms will converge to the exact solution to the problem, whenever it exists. They begin with the given initial guess of the solution and try to improve it. Selection of initial guess is generally depends on a good guess with the knowledge on the problem. All optimization algorithms will start from initial guess and move step by step to the optimal point. Each step vector will have a direction and length. While designing optimum algorithm to find minimum of objective function, line search and descent direction search must be considered.

Note that while computing Gradient (∇) and Hessian (\mathbf{H}), analytic expressions, given in 3.2.6 and 3.2.7, are used. Required first and second derivatives can be calculated from formulas given at A.

$$\nabla_i = \operatorname{Tr}\left\{\mathbf{R}_x^{-1} \frac{\partial \mathbf{R}_x^{-1}}{\partial \mu_i} (\mathbf{I} - \mathbf{R}_x^{-1} \hat{\mathbf{R}}_x)\right\} \quad (3.2.6)$$

$$\begin{aligned} [\mathbf{H}]_{(i,j)} = \operatorname{Tr}\left\{\left[\mathbf{R}_x^{-1} \frac{\partial^2 \mathbf{R}_x}{\partial \mu_i \partial \mu_j} - \mathbf{R}_x^{-1} \frac{\partial \mathbf{R}_x}{\partial \mu_i} \mathbf{R}_x^{-1} \frac{\partial \mathbf{R}_x}{\partial \mu_j}\right] \right. \\ \left. (\mathbf{I} - \mathbf{R}_x^{-1} \hat{\mathbf{R}}_x) + \mathbf{R}_x^{-1} \frac{\partial \mathbf{R}_x}{\partial \mu_i} \mathbf{R}_x^{-1} \frac{\partial \mathbf{R}_x}{\partial \mu_j} \mathbf{R}_x^{-1} \hat{\mathbf{R}}_x\right\} \quad (3.2.7) \end{aligned}$$

Generally, the idea behind most minimization methods is to compute and evaluate a value after moving a step along a given search direction, d_k . The formula of iterations

is given by

$$\boldsymbol{\mu}_{k+1} = \boldsymbol{\mu}_k + \alpha_k \mathbf{d}_k \quad k = 0, 1, 2, \dots \quad (3.2.8)$$

where the step length (size), α_k , must be chosen such that

$$\alpha_k = \arg \min_{\alpha} L(\boldsymbol{\mu}_k + \alpha \mathbf{d}_k) \quad k = 0, 1, 2, \dots \quad (3.2.9)$$

Step size, α_k , will be found by line search methods. The definition of \mathbf{d}_k depends on optimization method. As it is explained in A, in detail, from various optimization and line search methods, Fletcher Reeves with Three Point Line Search algorithm is selected.

3.3 Proposed Approach

3.3.1 Stochastic Maximum Likelihood with Turbo Approach

Since the estimation of all spectral parameters will be computationally expensive. A novel method is suggested in which frequency and spread values are estimated recursively.

First of all, Burg is used to estimate the Doppler frequencies and Power values of clutters. The estimates obtained from Burg are assumed to be true and standard deviations σ_i 's are estimated. After that, estimated standard deviations are assumed to be true values and Doppler frequencies of clutters are estimated. This loop is repeated for number of iterations times. The total number of iterations is represented as *numOfIt* in Figure 3.3.

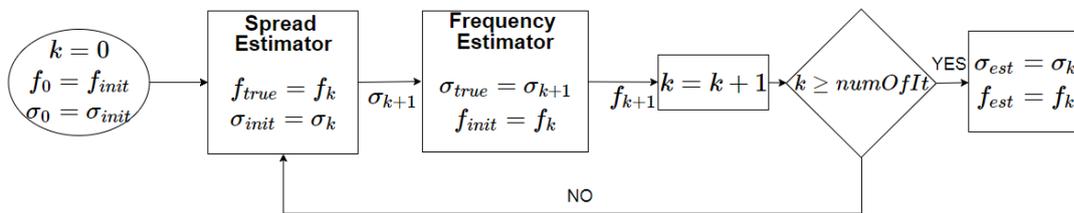


Figure 3.3: Turbo Method Flow Diagram

As it can be seen Turbo method consists of two SML estimator. The spread estimator is SML method with parameter vector, $\hat{\boldsymbol{\mu}} = [\sigma_1^2 \dots \sigma_N^2]$. True frequencies are assumed to be the ones estimated from Burg algorithm at first iteration. The other iterations pretend the previously estimated frequencies as true values. The output of spread estimator is taken as true values during frequency estimation. The frequency estimator is basically a SML method with parameter vectors, $\hat{\boldsymbol{\mu}} = [w_1 \dots w_N]$. The initial values of frequencies are selected as previously estimated ones after first iteration. The pseudo code of suggested Turbo method is given below,

Algorithm 1 Turbo Estimation Method

Require: $numOfIt > 0 \vee \boldsymbol{\sigma}_{init}, \mathbf{w}_{init}$

Ensure: $\mathbf{P}_{true} \leftarrow \mathbf{P}_{Burg}$

for $k \leftarrow 1$ to $numOfIt$ **do**

if $k \neq 1$ **then**

$\boldsymbol{\sigma}_{init} \leftarrow \boldsymbol{\sigma}_{k-1}$

$\mathbf{w}_{init} \leftarrow \mathbf{w}_{k-1}$

end if

$\mathbf{w}_{true} \leftarrow \mathbf{w}_{init}$

$\boldsymbol{\sigma}_k \leftarrow SML(\boldsymbol{\sigma}_{init})$

$\boldsymbol{\sigma}_{true} \leftarrow \boldsymbol{\sigma}_k$

$\mathbf{w}_k \leftarrow SML(\mathbf{w}_{init})$

end for

$\hat{\boldsymbol{\mu}} \leftarrow [w_{numOfIt}(1) \ \sigma_{numOfIt}^2(1) \ \dots \ w_{numOfIt}(N) \ \sigma_{numOfIt}^2(N)]$

Since the number of estimated parameters is decreased; Turbo approach will reduce computational effort significantly. At the same time, accuracy of estimations will be improved because of simplification of the optimization problem. In addition to that, initial values are chosen wisely which ensures convergence to the optimal parameter values.

3.4 Test Results

Mainly, two clutter PSD scenarios are investigated as the ones having distant and closely spaced components in spectral domain. First of all, the selection of number of Turbo iterations is explained by comparing accuracy of estimated PSD's after each of Turbo iterations. After selecting the number of Turbo iterations, performances of estimation algorithms are compared with CRB. CRB calculation is studied in A. The comparison is performed by examining the error and CRB with respect to number of snapshots, secondary range cells. Mainly, SML in [22] with initial frequencies picked from Burg algorithm and selected as 0 are investigated. Similarly, Turbo approach is implemented with different frequency initializations. Constant values during simulations can be listed as;

- A Coherent Processing Interval (CPI) consists of 16 pulses.
- In a CPI, PRI does not change and it is 100 μ sec.
- Operational frequency is selected as 10 GHz.
- The initial values of spreads are taken equal for each clutter as 0.5 m/sec at all cases.
- Number of snapshots is 5.
- $\sigma_{v_{sea}}$ is 1.512 m/sec and $\sigma_{v_{rain}}$ is selected as 3.108 m/sec.
- $\sigma_{v_{init}}$'s are 0.5 m/sec.
- Fletcher-Reeves optimization algorithm with Three Point Line Search method is used to find the maximum of Log-likelihood function.
- Number of iterations of optimization algorithm is selected as 50.
- Tolerance of optimization algorithm is chosen as 10^{-7} .
- Sufficiently large number (≥ 50) of Monte Carlo samples are generated for each scenario.

3.4.1 Distant Clutters

As the distinction between clutter Doppler frequencies increases, the estimation performance is expected to be improved. Thus, first of all, spectrum composed of two relatively isolated clutters is estimated. Sea and rain velocities are selected as 3.872 and 26.178 (m/sec). Sea CNR value is taken as 50 dB during tests only rain CNR is changed to 30, 40 or 50 dB.

3.4.1.1 Number of Turbo Iterations

Before comparing the suggested estimation algorithms, Turbo loop repetition count must be selected. The number of Turbo iterations in Figure 3.3 is selected as 1,2 and 3. If 1 Turbo iteration is performed, Doppler frequencies will be estimated starting from the estimates of Burg. By accepting estimated frequencies as true ones, spreads will be estimated. 2 and 3 iteration works recursively. The estimated PSDs at each of Monte Carlo trials is averaged. Results are illustrated in Figure 3.4.

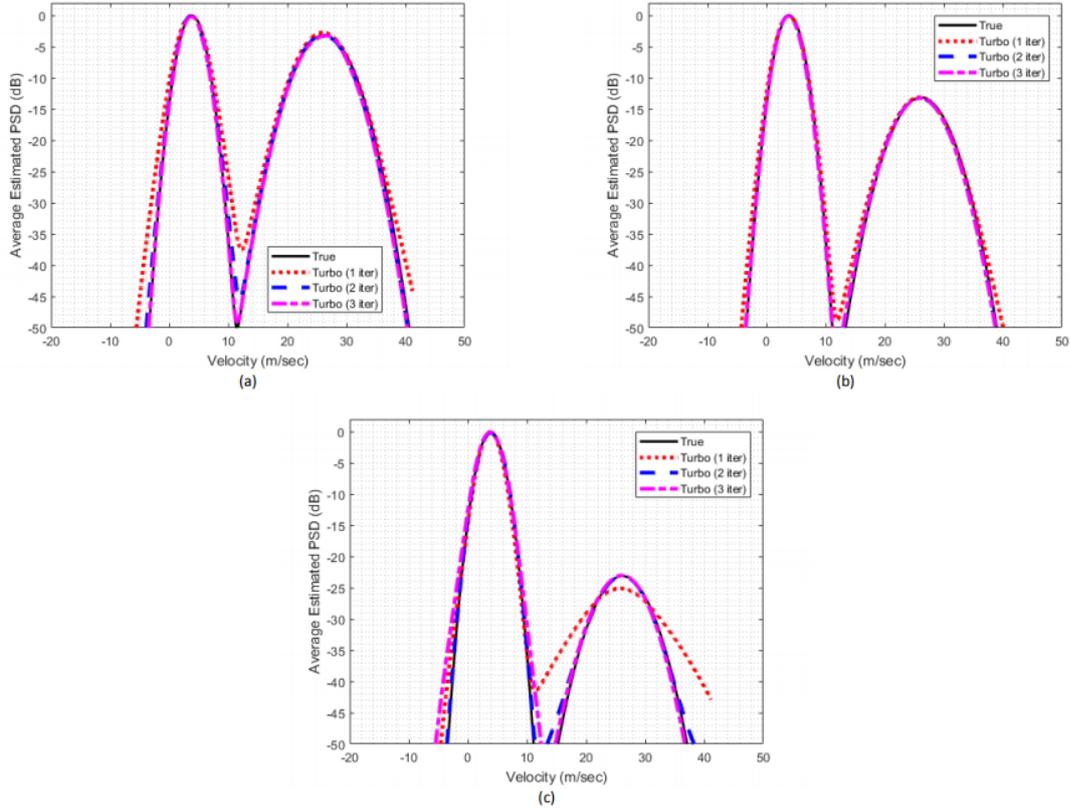


Figure 3.4: Averaged PSD estimates with different number of Turbo iterations
(a) $CNR_{rain} = CNR_{sea} = 50\text{dB}$ (b) $CNR_{rain} = 40\text{dB}$ $CNR_{sea} = 50\text{dB}$
(c) $CNR_{rain} = 30\text{dB}$ $CNR_{sea} = 50\text{dB}$

As it can be seen, 2 and 3 iterations gave nearly same results. Turbo iteration number is selected as 3 for accuracy.

3.4.1.2 Performances of Algorithms

After selecting number of Turbo iterations, performances of suggested algorithms will be compared. For distant clutter scenario, Mean Square Error (MSE) of estimated parameters are compared with CRB. Algorithms under interest are SML with zero initial Doppler frequencies, SML with initial velocities estimated by Burg, Turbo with 3 iterations started from zero initial frequencies and 3 iterations Turbo initial frequencies are estimated by Burg. Note that, since they are outlier estimates, spread values, greater than 5 m/sec, and velocities, bigger than 100 m/sec, are not included

in averaged error calculation of trials. Mainly, 50 Monte Carlo trials are performed.

MSE and CRB Comparison of Methods with Different CNR Pairs Firstly, the parameters of distant clutters are estimated when they have same power values. In other words, both sea and rain clutters have 50 dB CNR. The normalized MSE of estimation are shown in Figure 3.5.

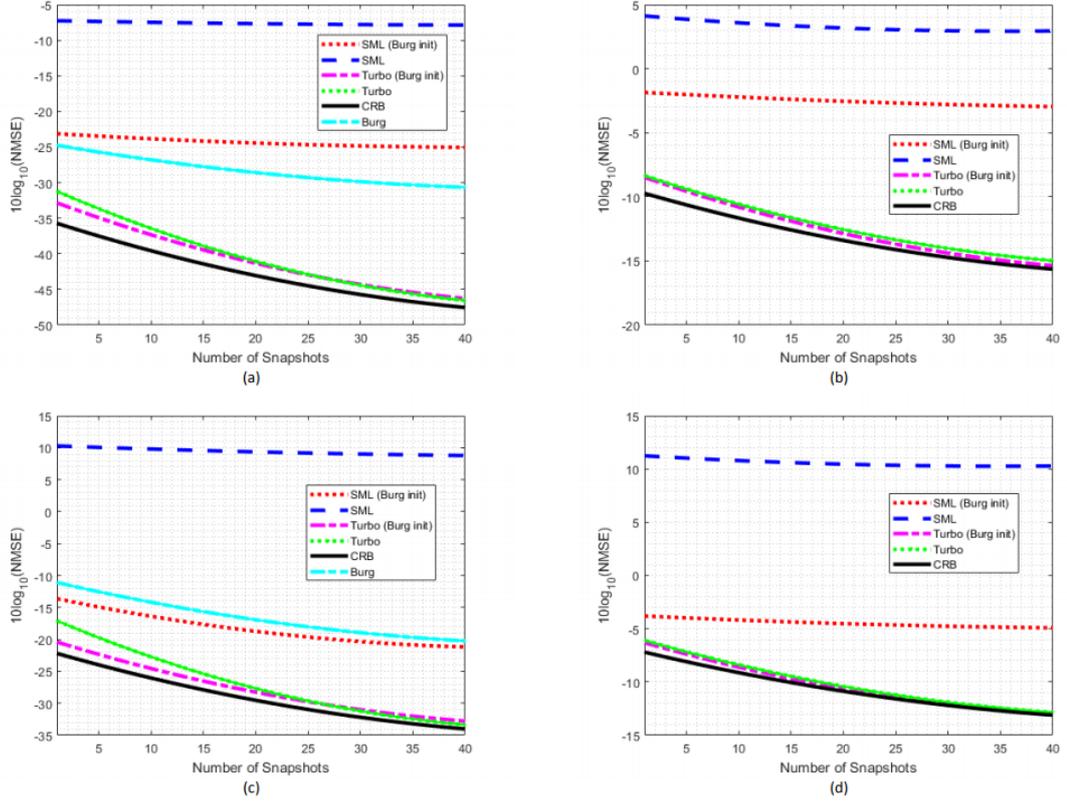


Figure 3.5: MSE vs number of snapshots curves for parameters (a)rain velocity estimate (b)rain spread estimate (c)sea velocity estimate (d)sea spread estimate

Turbo methods outperformed SML significantly. Turbo methods achieved the CRB. In other words, they give the best possible estimates for parameters of interest. While, initiation of frequency estimate from Burg improved SML estimation accuracy, its effect on Turbo was not considerable.

Secondly, the parameters of distant clutters are estimated when they have 10 dB power difference. CNR_{sea} is equal to 50 dB, while CNR_{rain} is 40 dB. The normalized MSE of estimation are plotted with respect to number of snapshots in Figure 3.6.

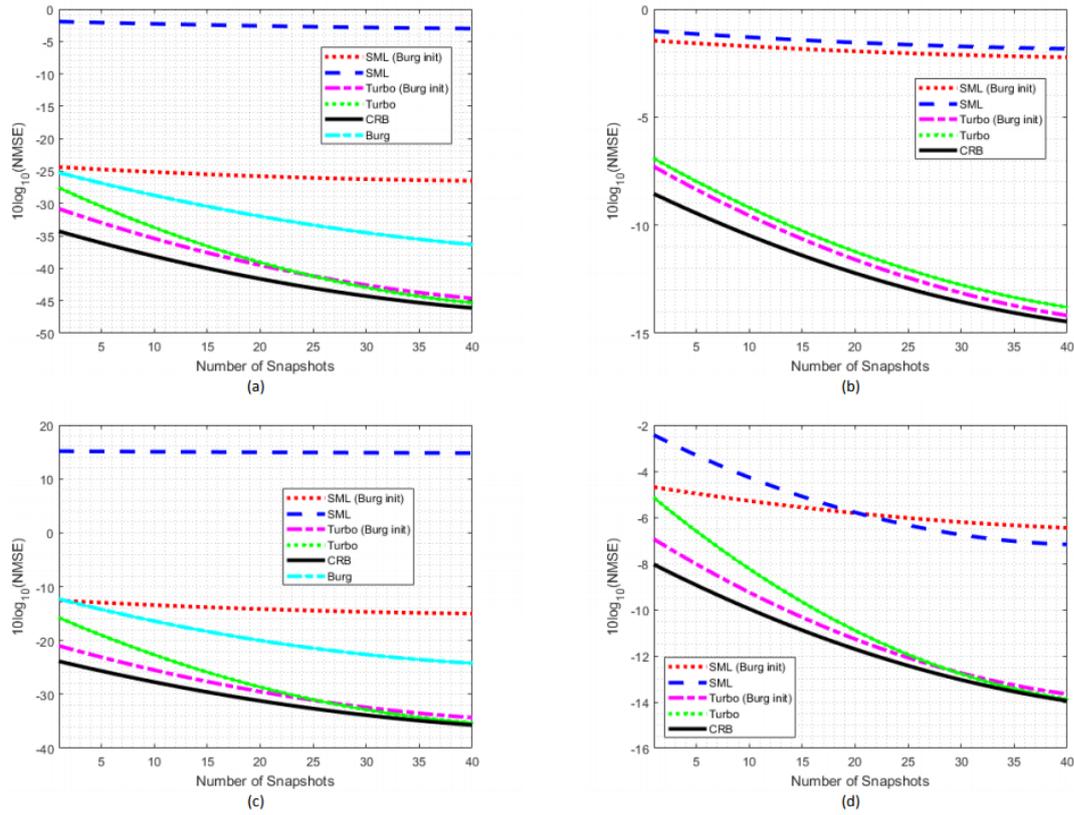


Figure 3.6: MSE vs number of snapshots curves for parameters (a)rain velocity estimate (b)rain spread estimate (c)sea velocity estimate (d)sea spread estimate

Again, Turbo methods performed better than SML. Selecting initial frequencies from Burg improved both Turbo and SML methods.

Finally, the parameters of distant clutters are estimated when they have 20 dB power difference. CNR_{sea} is equal to 50 dB, while CNR_{rain} is 30 dB. The normalized MSE of estimation are plotted with respect to number of snapshots in Figure 3.7.

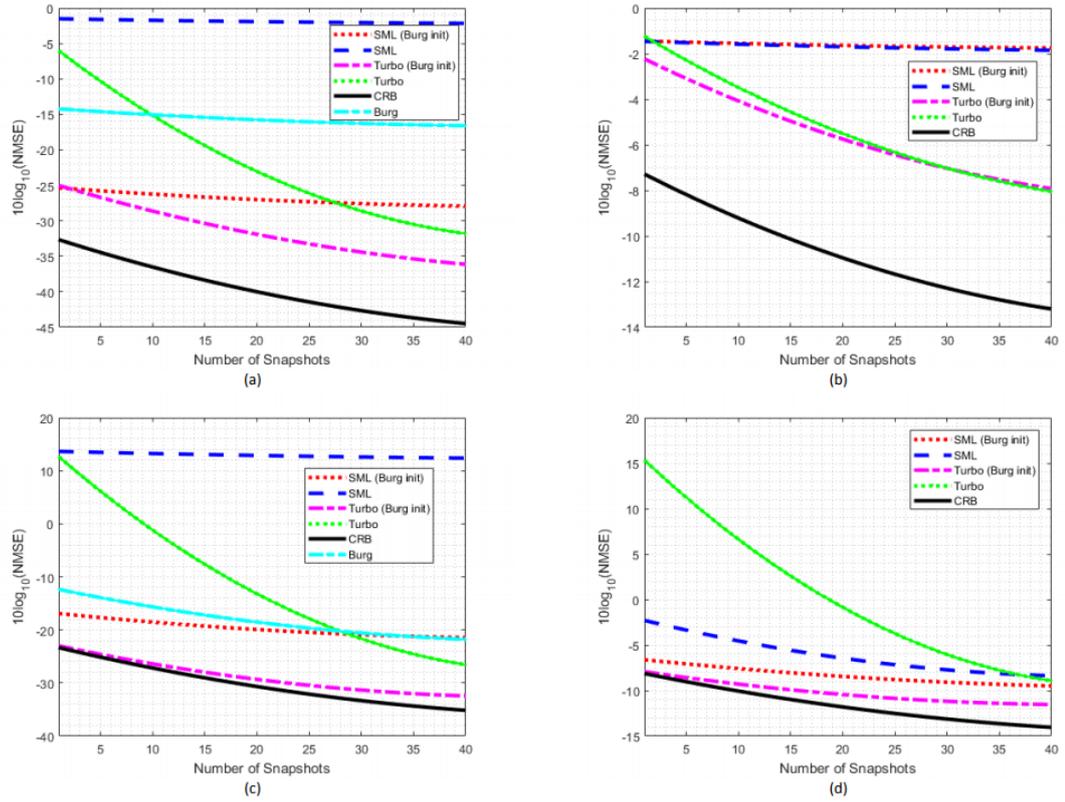


Figure 3.7: MSE vs number of snapshots curves for parameters (a)rain velocity estimate (b)rain spread estimate (c)sea velocity estimate (d)sea spread estimate

As power difference between clutters become noticeable, the importance of Burg initiation can be observed. As expected, Turbo initiated by Burg has most accuracy. Since the power of rain decreased, its spread estimation becomes poor.

Estimated PSD Comparisons of Methods with Different CNR Pairs The PSD estimation performance of suggested algorithms is observed, when number of snapshots is fixed to 5. The PSD estimates at each Monte Carlo trial are averaged such that the results can be seen in Figure 3.8.

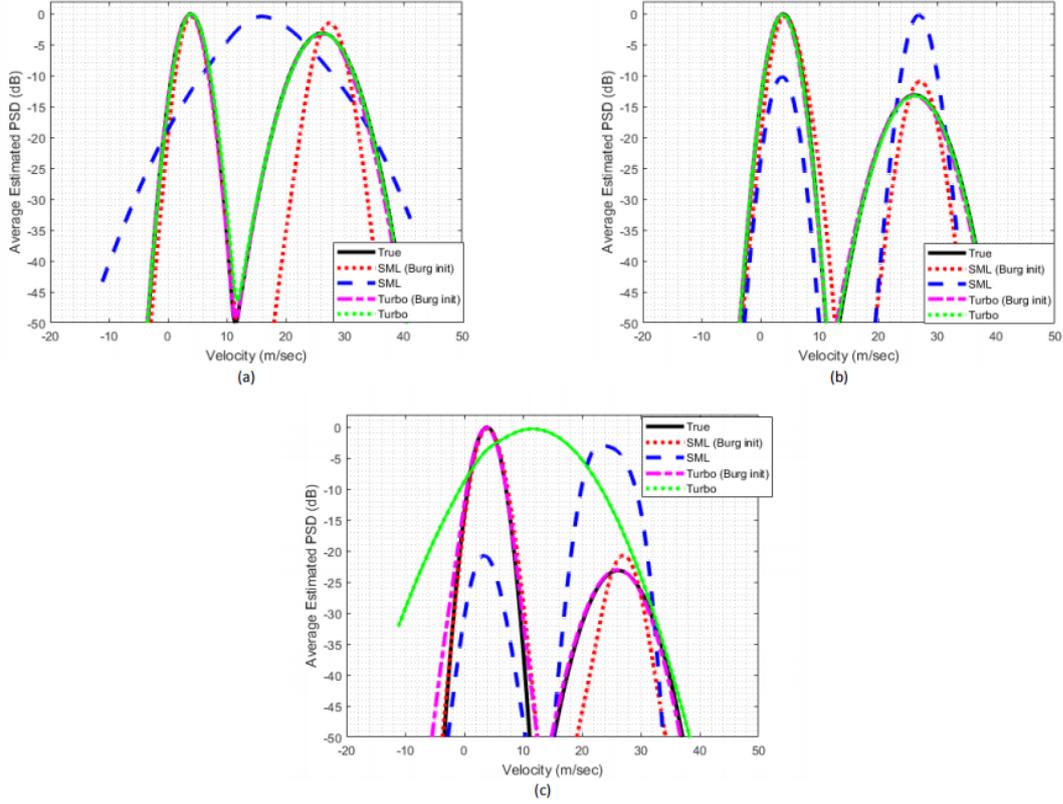


Figure 3.8: Averaged PSD estimates of estimation methods (a) $CNR_{rain} = CNR_{sea} = 50\text{dB}$ (b) $CNR_{rain} = 40\text{dB}$ $CNR_{sea} = 50\text{dB}$ (c) $CNR_{rain} = 30\text{dB}$ $CNR_{sea} = 50\text{dB}$

As stated Turbo methods are preferable than SML. When power difference is greater than or equal to 20 dB. The initial frequency selection must be performed wisely.

Estimated PSD Comparisons of Methods with Different PRFs Low PRF values are generally defined between 1 and 8 kHz. They are advantageous during maximum range detection, but long transmit pulse duration is necessary in order to send enough power for detection. Range measurements are unambiguous, while velocity measurements are ambiguous. Medium PRF values are generally in the range of 8-30 kHz. The medium PRF will be a good selection for scanning radars. Both range and Doppler ambiguities may occur but they are less severe. High PRF values are between 30-250 kHz. They enable greatest detection range. Unlike low PRFs, range measurements are ambiguous, velocity measurements are unambiguous. In order to

justify the performance of estimation methods in all PRF regimes, several scenarios are tested. Firstly, 4 kHz PRF is selected with 8 pulses. Secondly, 10 kHz with 20 pulses are implemented. Finally, for high PRF 40 kHz is selected with 80 pulses. All scenarios will have same CPI. The result can be seen in Figure 3.9.

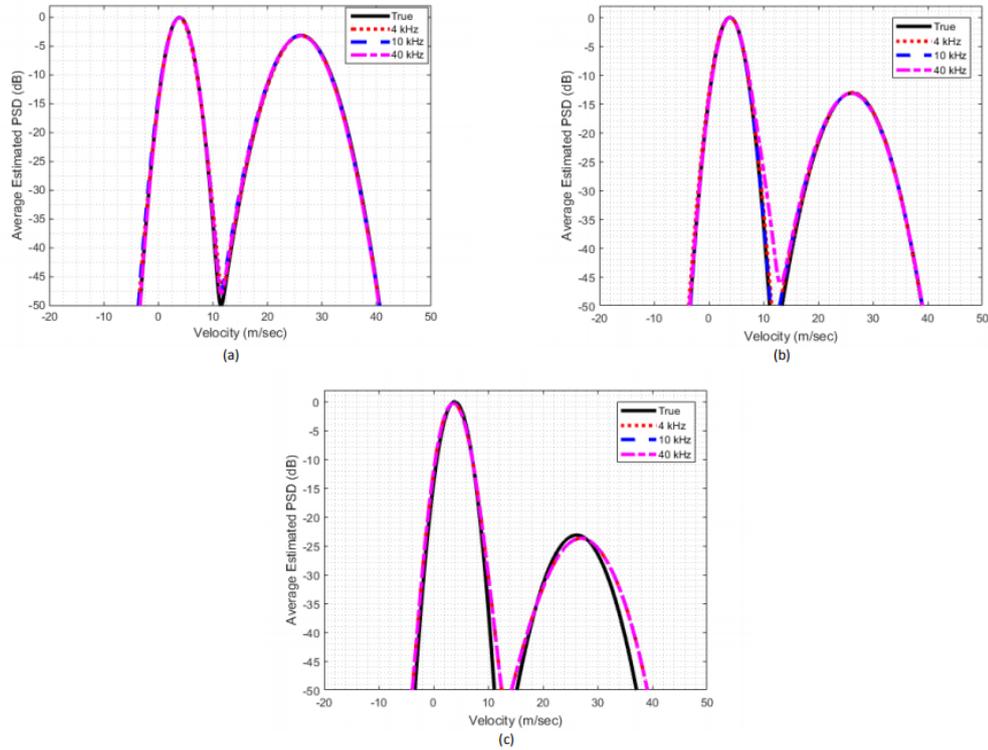


Figure 3.9: Averaged PSD estimates at different PRF values (a) $CNR_{rain} = CNR_{sea} = 50\text{dB}$ (b) $CNR_{rain} = 40\text{dB}$ $CNR_{sea} = 50\text{dB}$ (c) $CNR_{rain} = 30\text{dB}$ $CNR_{sea} = 50\text{dB}$

3.4.2 Closely Spaced Clutters

When weather conditions is tough, the detection becomes troublesome. Sea and rain velocities are selected as 3.872 and 12.345 (m/sec). Similar scenarios with distant clutter case are investigated.

3.4.2.1 Number of Turbo Iterations

One, two and three Turbo iteration are applied. The effect of iteration number can be seen in Figure 3.3 from averaged PSDs obtained at each trial.

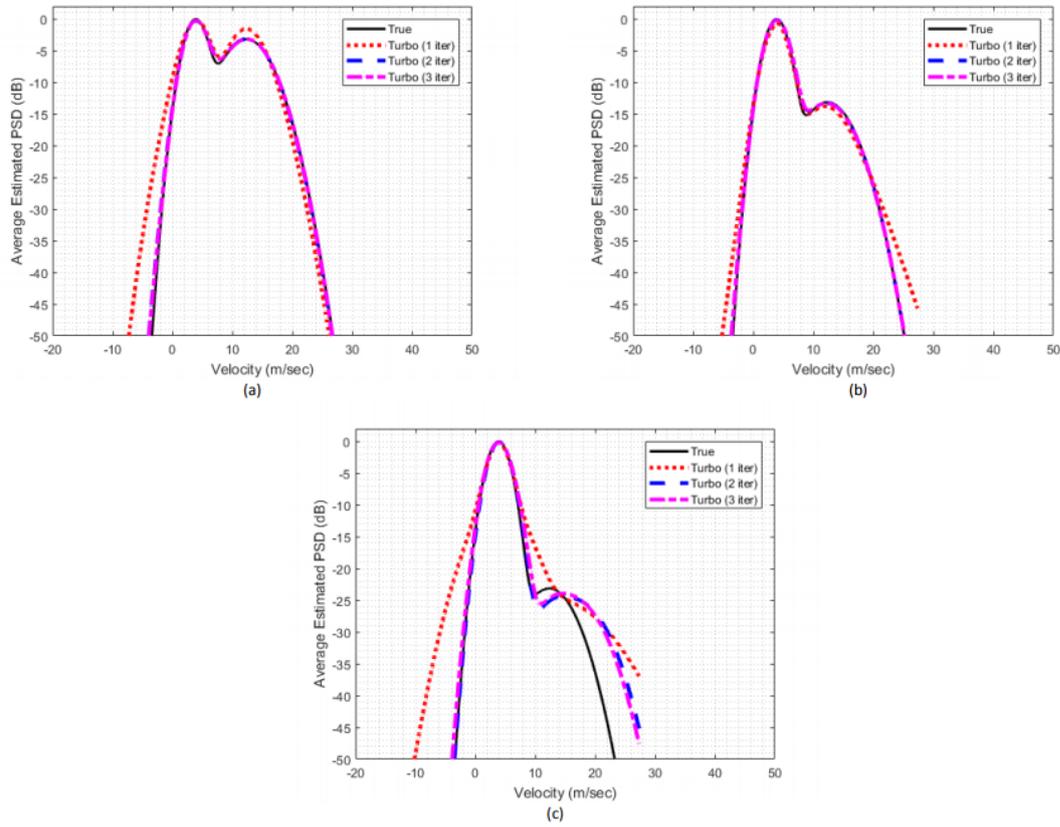


Figure 3.10: Averaged PSD estimates with different number of turbo iterations
 (a) $CNR_{rain} = CNR_{sea} = 50\text{dB}$ (b) $CNR_{rain} = 40\text{dB}$ $CNR_{sea} = 50\text{dB}$
 (c) $CNR_{rain} = 30\text{dB}$ $CNR_{sea} = 50\text{dB}$

Turbo iteration number can be selected as 2 or 3. It is chosen as 3.

3.4.2.2 Performances of Algorithms

For closely spaced clutter scenario, MSE of estimated parameters are compared with CRB. Algorithms under interest are SML with zero initial Doppler frequencies, SML with initial velocities estimated by Burg, Turbo with 3 iterations started from zero initial frequencies and 3 iterations Turbo frequencies initiated by Burg. Outliers are

excluded.

MSE and CRB Comparison of Methods with Different CNR Pairs First case is closely spaced clutters having same powers. In other words, both sea and rain clutters have 50 dB CNR. The normalized MSE of estimation are shown in Figure 3.11.

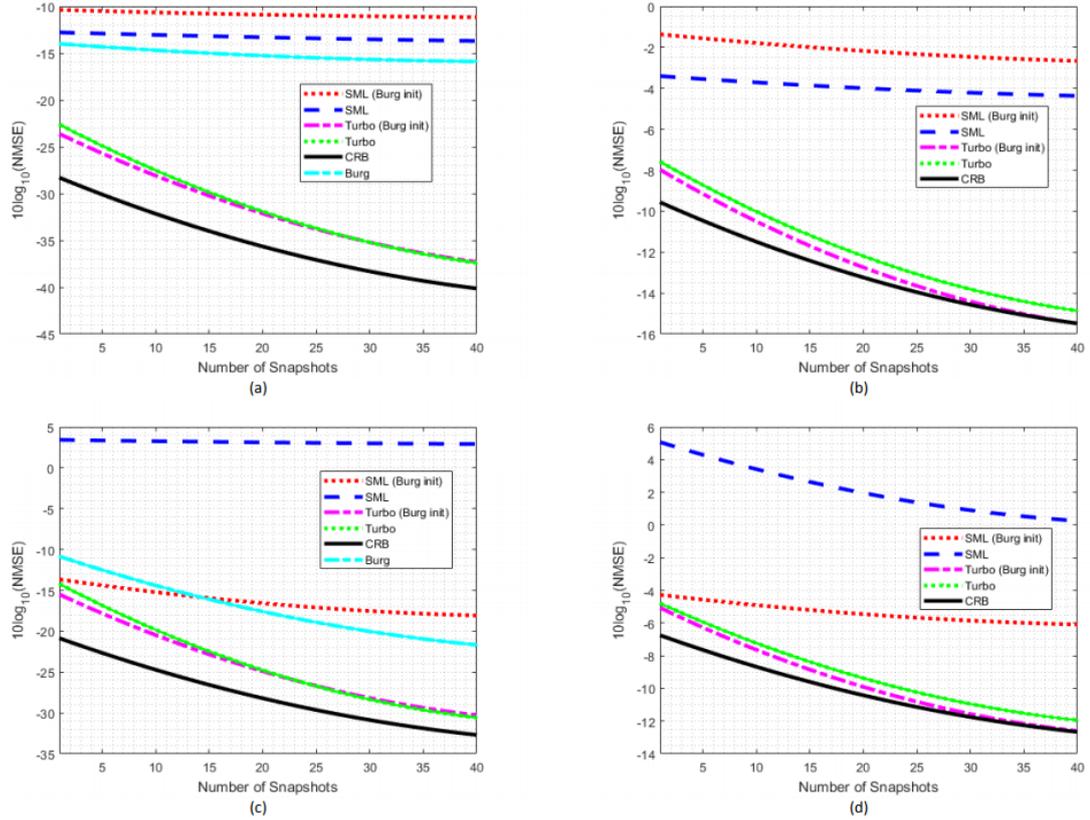


Figure 3.11: MSE vs number of snapshots curves for parameters (a)rain velocity estimate (b)rain spread estimate (c)sea velocity estimate (d)sea spread estimate

Turbo methods outperformed SML significantly. Turbo methods nearly achieve the CRB. Initiation of frequency estimate from Burg improves SML estimation accuracy significantly. Unlike distant clutters, its effect on Turbo is visible.

Second case is closely spaced clutters having 10 dB power difference. CNR_{sea} is equal to 50 dB, while CNR_{rain} is 40 dB. The normalized MSE of estimation are plotted with respect to number of snapshots in Figure 3.12.

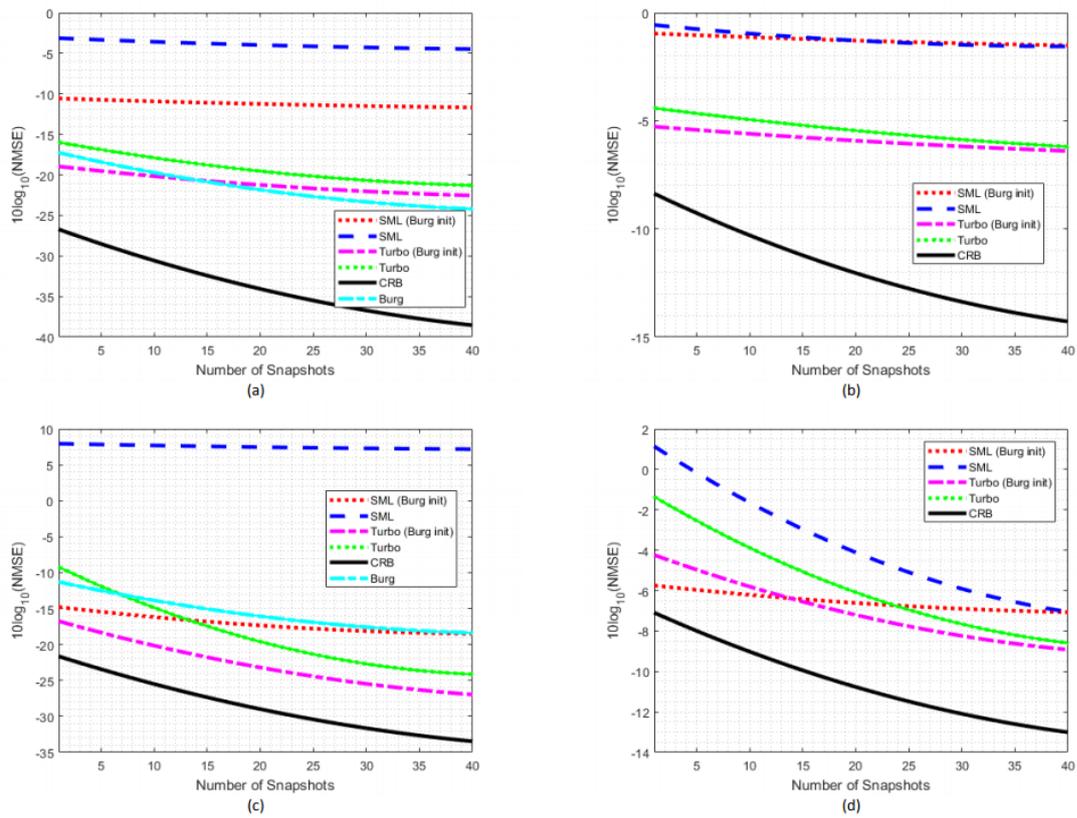


Figure 3.12: MSE vs number of snapshots curves for parameters (a)rain velocity estimate (b)rain spread estimate (c)sea velocity estimate (d)sea spread estimate

As in all scenarios, Turbo methods perform better than SML. The selection of initial frequencies by Burg advances both Turbo and SML methods.

The last case is closely spaced clutters having 20 dB power difference. CNR_{sea} is equal to 50 dB, while CNR_{rain} is 30 dB. The normalized MSE of estimation are plotted with respect to number of snapshots in Figure 3.13.

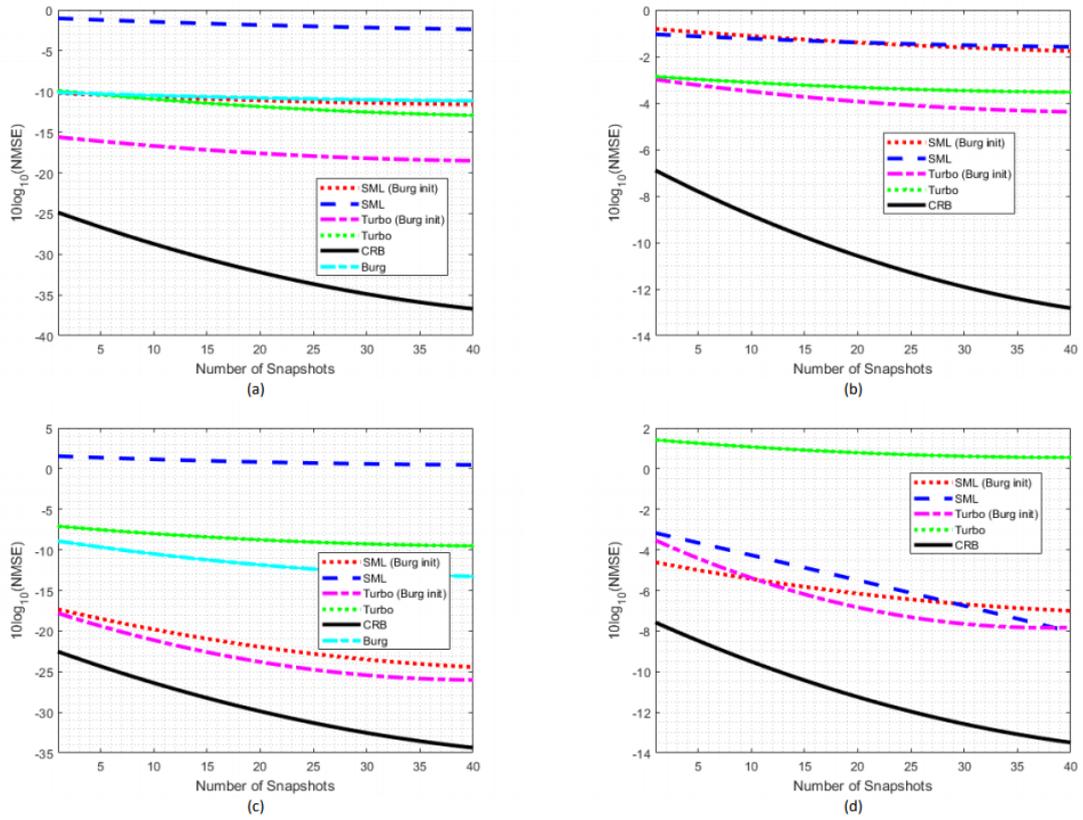


Figure 3.13: MSE vs number of snapshots curves for parameters (a)rain velocity estimate (b)rain spread estimate (c)sea velocity estimate (d)sea spread estimate

The initiation of frequencies by Burg improves performance especially power difference between is greater than 10 dB.

Estimated PSD Comparisons of Methods with Different CNR Pairs The PSD estimation performance of suggested algorithms for two closely spaced clutters is observed. The number of snapshots is fixed to 5. The PSD estimates at each Monte Carlo trial are averaged such that the results can be seen in Figure 3.14.

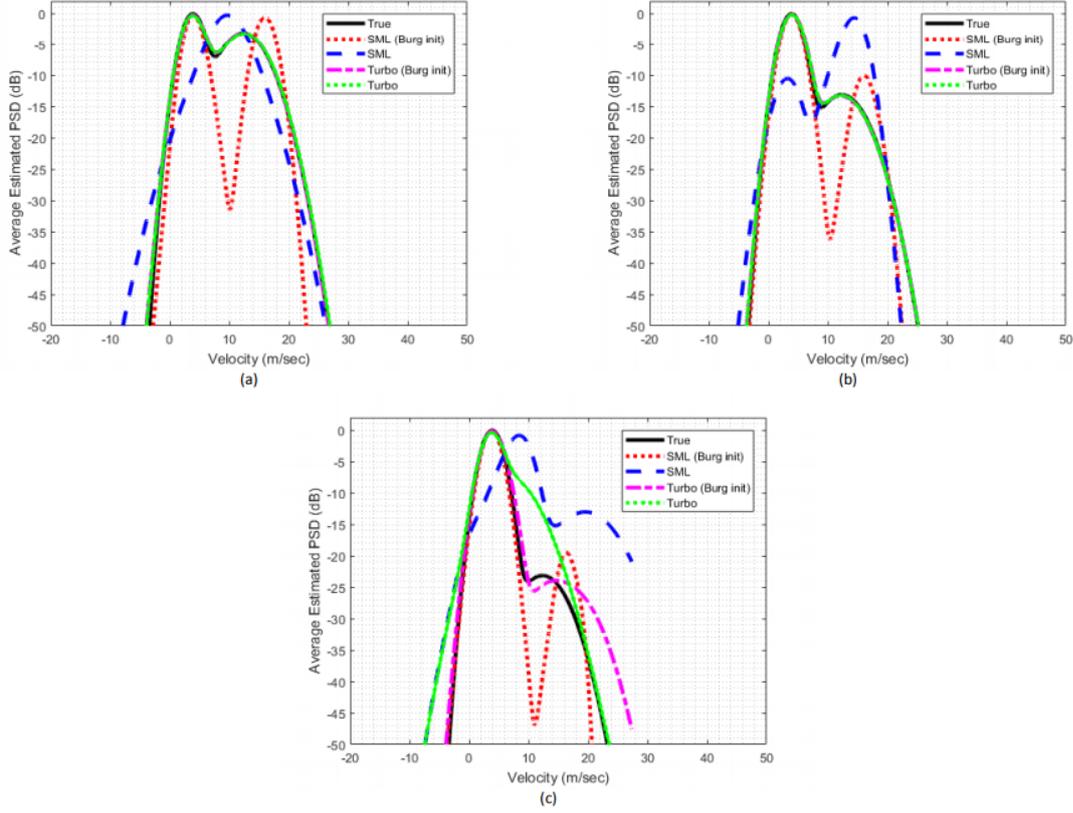


Figure 3.14: Averaged PSD estimates of estimation algorithms (a) $CNR_{rain} = CNR_{sea} = 50\text{dB}$ (b) $CNR_{rain} = 40\text{dB}$ $CNR_{sea} = 50\text{dB}$ (c) $CNR_{rain} = 30\text{dB}$ $CNR_{sea} = 50\text{dB}$

It is good to note that the initial frequency selection must be performed carefully. The slight degradation of Turbo performance in Figure 3.14 (c) is resulted from Burg frequency estimates.

All in all, the designed novel Turbo estimation method shows its prominent power over SML, suggested in literature.

Estimated PSD Comparisons of Methods with Different PRFs Similar to the distant clutter case, PRFs are selected as 4, 10 and 40 kHz with corresponding number of pulses, 8, 20 and 80. The compatibility of Turbo to each PRF regimes is illustrated for closely spaced clutters. The result can be seen in Figure 3.15.

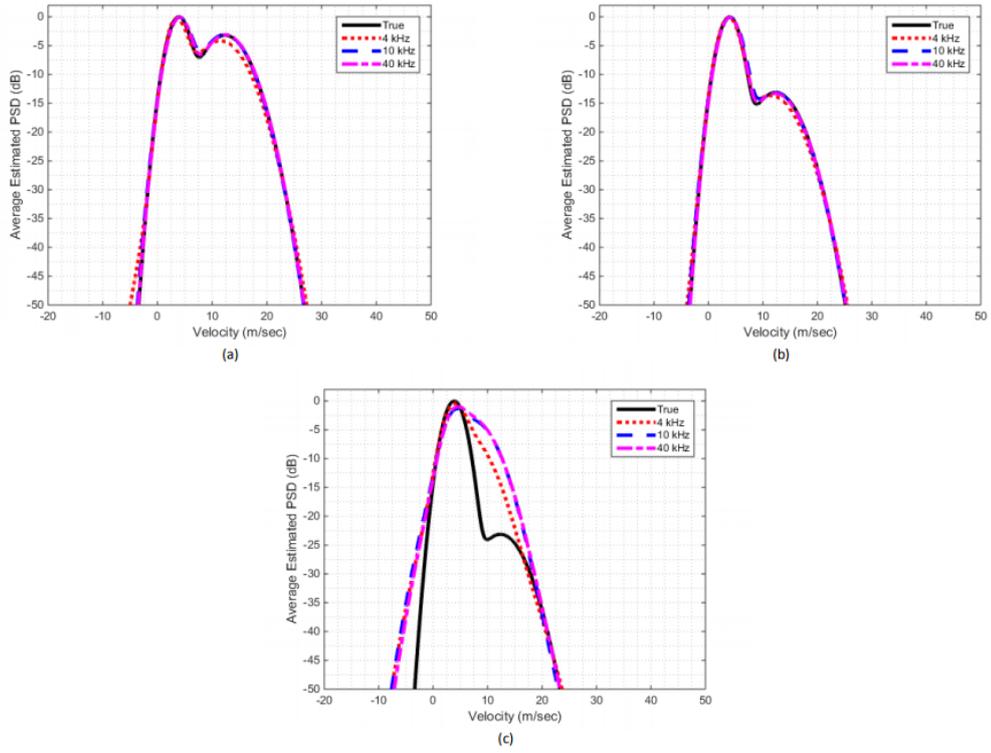


Figure 3.15: Averaged PSD estimates at different PRF values (a) $CNR_{rain} = CNR_{sea} = 50\text{dB}$ (b) $CNR_{rain} = 40\text{dB}$ $CNR_{sea} = 50\text{dB}$ (c) $CNR_{rain} = 30\text{dB}$ $CNR_{sea} = 50\text{dB}$

CHAPTER 4

RADAR DETECTION BASED ON ESTIMATED CLUTTER POWER PROFILE

Radar detection is defined as deducing the presence of a target by analyzing received echo signal. In harsh weather conditions, radar detection performance may degrade because of interference signals. Thus, robust algorithms must be designed in order to suppress clutter echoes. The purpose of slow-time clutter suppression is to maintain target signal having a certain Doppler frequency during suppression as possible.

Received echo components, target and interference signals, are random because of that radar detection process can be considered as a statistical one. Thus, clutter statistical properties is crucial while synthesize detection algorithm. After correlation features of clutter are determined as precise as possible, an appropriate detector will be formulated.

As stated, since clutter return contains a large number of scatterers, Central Limit Theorem can be used. Thus, clutter returns can be modelled with multivariate Gaussian statistics. With a proper and robust estimation algorithm, Power Spectrum Density of clutter signals in received echo is estimated accurately in 3. Thus, a detector can be designed during operation adaptively.

In this chapter, the performance of clutter estimation methods will be observed with a detector implementation. A coherent radar pulse set, comprised from a target signal, two clutters and noise, is simulated for performance measures. First of all, conventional methods are compared. After that, proposed method is analyzed in detail. Finally, all mentioned methods are compared in order to observe detection performance of proposed detector.

4.1 Radar Detection

As stated in Chapter 2, classical detection process follows three steps. Firstly, clutter suppression is done by filtering. After that, Doppler processing is applied to the received echo. And finally, thresholding is implemented. Doppler processing may include clutter suppression for some cases. In a modern radar system, all of three steps are performed at the same time.

4.1.1 Signal Model

Radar detection of targets can be modelled by a binary hypothesis test. If target is absent, H_0 is the hypothesis. In case of target presence, hypothesis is represented by H_1 .

$$\begin{cases} H_0 : \mathbf{x} = \mathbf{c} + \mathbf{n} \\ H_1 : \mathbf{x} = \mathbf{s} + \mathbf{c} + \mathbf{n} \end{cases} \quad (4.1.1)$$

\mathbf{x} stands for the signal from the cell under test (CUT), at a specific range bin. In other words, it is the collection P samples, collected from a specific range bin in slow time. \mathbf{c} is a vector having size P , standing for clutter samples. Similarly, vector \mathbf{n} represents the white noise in received echo. \mathbf{s} is the contribution of target. In case of target is presence, H_0 is decided; this is defined as a *miss*. On the other hand, if decision is H_1 , then it is *detection*. If target is absent, and its absence, H_0 , is determined, *correct rejection* is accomplished. However, if decision is H_1 then a *false alarm* occurs.

The absence and presence decisions for target are interpreted by thresholding the current case. The most common criteria used for thresholding is Neyman-Pearson which aims to obtain maximum probability of detection with a predetermined constant false alarm probability. The detectors using Neyman-Pearson criterion is named as Constant False Alarm Rate (CFAR) detectors.

If an integral is taken over region R_1 , containing all measurements, a target presence is decided with likelihood functions. Probability of detection (P_d) and Probability of

false alarm (P_{fa}) is calculated by

$$P_d = \int_{R_1} p_{\mathbf{x}}(\mathbf{x}|H_1) d\mathbf{x} \quad (4.1.2)$$

$$P_{fa} = \int_{R_1} p_{\mathbf{x}}(\mathbf{x}|H_0) d\mathbf{x} \quad (4.1.3)$$

Conditional probability density functions (pdfs) stand for likelihood functions.

If maximum P_{fa} is bounded by a constant value, a threshold can be obtained. In other words, Neyman-Pearson criterion is applied to obtain decision rule.

$$\frac{p_{\mathbf{x}}(\mathbf{x}|H_1)}{p_{\mathbf{x}}(\mathbf{x}|H_0)} \underset{H_0}{\overset{H_1}{\gtrless}} \gamma \quad (4.1.4)$$

Two conditional pdfs can only be decided upon knowledge of interference distribution in CUT. If clutter and noise distributions are perfectly known, then ideal detection of target can be performed.

Received echo, \mathbf{x} ; is actually a column array having dimension P , number of pulses transmitted in one PRI.

$$\mathbf{x} = [x(1) \ x(2) \ x(3) \ \dots \ x(P)]^T \quad (4.1.5)$$

During the thesis interference, \mathbf{c} ; represents clutter signals. $\mathbf{a}(\theta_s)$ stands for Doppler steering vector of target having Doppler frequency f_d ,

$$\mathbf{a}(\theta_s) = [1 \ e^{j2\pi f_d P R I} \ e^{j2\pi f_d P R I} \ \dots \ e^{j(P-1)2\pi f_d P R I}] \quad (4.1.6)$$

Since noise power is accepted as 1, for Swerling-1 target model, received target signal can be written as

$$\mathbf{s} = \sqrt{\frac{SNR}{P}} \times \alpha \times \mathbf{a}(\theta_s) \quad (4.1.7)$$

α is assumed to be a complex Gaussian random variable having zero mean and variance 1, in short $\alpha \sim N(0, 1)$.

It is desired to decrease the effect of clutter and noise in order to obtain accurate information about target.

4.2 Conventional Detectors for Clutter Suppression and Doppler Processing

Pulses from a pre-specified range bin are processed such that interference will be removed while getting useful target Doppler information. The processing is applied to each range bin. Pulse samples are separated by a Pulse Repetition Interval (PRI).

If number of pulses is denoted by P , the input signal in case of target to the Doppler processing unit will be;

$$x_j = s_j + c_j + n_j \quad j = 0, 1, 2, \dots, P \quad (4.2.1)$$

In classical approaches, received and sampled signal vector is filtered with a LTI filter followed by a DFT operation.

4.2.1 Moving Target Indicator (MTI) and Moving Target Detection (MTD) Algorithms

In order to get rid of clutter signal in received echo, Moving Target Indicator (MTI) and Moving Target Detection (MTD) are two commonly used methods.

MTI method removes the stationary clutter components, caused by reflections from mountains, buildings and hills. It uses the phase difference in received echo, resulted from moving targets. Since stationary objects cannot cause different phase shift between pulses, they can be eliminated. MTI algorithms are also known as pulse cancellers such as Single or Double Delay Line Cancellers. The method is simple and practical, but it has low performance in case of low-radial velocity targets. Additionally, it will fail to suppress clutters having moderate velocity. If radar will operate on a rainy day, its detection performance will be degraded. MTI algorithms are also impractical for rotating radars. Since stationary clutters will be perceived like a moving one because of rotation, they will be eliminated.

If steered MTI filter coefficients are denoted with w_{MTI} Assume that MTI filter is $H(z) = 1 - z^{-1}$, such that signal after MTI can be found by formula $\mathbf{x}_{MTI} = \mathbf{M}\mathbf{x}$

$$\mathbf{x}_{MTI} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & -1 & 1 & \dots & 0 & 0 \\ \cdot & \cdot & \cdot & \dots & \cdot & \cdot \\ \cdot & \cdot & \cdot & \dots & \cdot & \cdot \\ \cdot & \cdot & \cdot & \dots & \cdot & \cdot \\ 0 & 0 & 0 & \dots & -1 & 1 \end{bmatrix} \mathbf{x} \quad (4.2.2)$$

Note that, first sample in \mathbf{x} remains same after filtering. The reason behind that steady state response of filter will be observed at least 2 samples. Thus, if DFT is applied

output of DFT will be:

$$\mathbf{x}_{MTI+DFT} = (\mathbf{a}(\phi)(2 : P))^H \mathbf{x}_{MTI}(2 : P) \quad (4.2.3)$$

On the other hand, MTD has a capacity to solve problems encountered while working with MTI. The difference between MTI and MTD is that MTD can recognize moving targets. A sub-optimum filter can be obtained by using MTD. MTD only uses an approximate ground clutter covariance matrix and receiver noise to detect moving targets. MTD concepts combine Doppler filters with Constant False Alarm Rate (CFAR) Detectors.

The classical Doppler processing is implemented with Fast-Fourier Transformation (FFT). This method fails to detect targets having relatively close Doppler values. If the disadvantage is tried to be removed with windowing, there will be a loss in SNR. Thus, an optimum filter design is required other than previously mentioned methods. As a conventional method, windowed DFT filter can also be used. DFT is similar to a set of passband filters. The method has wide minimum detectable velocity range. It is used to transform time domain to frequency domain. Windowing reduced signal ripples in the frequency domain. It is important while designing filters. Proper selection of window function increases the spectral frequency resolution. If \mathbf{win} stands for P-point Chebyshev window with 70 dB side-lobe magnitude factor, the Windowed DFT filter coefficients will be,

$$\mathbf{w} = \text{diag}(\mathbf{win})\mathbf{a}(\theta_s) \quad (4.2.4)$$

While MTI techniques just separates moving targets from clutter signals, Pulsed Doppler techniques separate targets into different regimes additional to clutter cancellation.

4.2.2 Optimal Clutter Suppression Filter

A filter, which is equivalent to MMSE or Wiener filter, is suggested in Doppler processing. It minimizes the mean squared error between its output and the target return. It is known as the optimum clutter suppression filter. In the case of deterministic signal, it is in the same class with the Max-SINR or Eigen-filter. The filter aims

to maximize the signal to interference power ratio at its output when second order statistics of clutter is known. Let's go back to the signal model. For a phased array radar, antenna array receives a narrowband signal at the time instant k which can be represented mathematically as,

$$\mathbf{x}(k) = \mathbf{s}(k) + \mathbf{c}(k) + \mathbf{n}(k) \quad (4.2.5)$$

$\mathbf{x}(k)$, $\mathbf{s}(k)$, $\mathbf{c}(k)$ and $\mathbf{n}(k)$ are all vectors having length, equals to number of pulses, P . $\mathbf{s}(k)$ stands for the desired signal, $\mathbf{c}(k)$ is clutter, and $\mathbf{n}(k)$ is noise term.

The received signal is assumed to be zero-mean and nearly stationary. Moreover, it is assumed that the desired signal is uncorrelated with the clutter and noise signals.

$$\mathbf{s}(k) = s(k)\mathbf{a}(\theta_s) \quad (4.2.6)$$

where $s(k)$ is the signal waveform and $\mathbf{a}(\theta_s)$ is the desired signal steering vector. For convenience, $s(k)$ is taken as constant.

If filtering is applied to received signal, such that

$$z(k) = \mathbf{w}^H \mathbf{x}(k) \quad (4.2.7)$$

\mathbf{w} represents complex weight vector. It is the column vector having length, equals to number of pulses. The interference and noise covariance matrix is defined as

$$\mathbf{R}_{c+n} = E\{(\mathbf{c} + \mathbf{n})(\mathbf{c} + \mathbf{n})^H\} \quad (4.2.8)$$

In case of a point source, steering vector is known. Thus, filter output SINR can be given as;

$$SINR_{out} = \frac{E[|\mathbf{w}^H \mathbf{s}|^2]}{E[|\mathbf{w}^H (\mathbf{c} + \mathbf{n})|^2]} = \frac{E[|s(k)|^2] |\mathbf{w}^H \mathbf{a}(\theta_s)|^2}{\mathbf{w}^H \mathbf{R}_{c+n} \mathbf{w}} \quad (4.2.9)$$

For most of the radar systems, the main goal is to minimize signal-to-interference-plus-noise ratio (SINR).

$$SINR = \frac{\mathbf{w}^H \mathbf{s} \mathbf{s}^H \mathbf{w}}{\mathbf{w}^H \mathbf{R}_{c+n} \mathbf{w}} \quad (4.2.10)$$

where Additionally, the matrix \mathbf{C} is defined such that the noise and clutter covariance matrix can be written as

$$\mathbf{R}_{c+n} = \mathbf{C}^H \mathbf{C} \quad (4.2.11)$$

The optimum filter must maximize SINR at output. Cauchy–Schwarz inequality can be used to find optimum filter coefficients which are shown in 4.2.12.

$$|\langle \mathbf{x}, \mathbf{y} \rangle|^2 \leq \|\mathbf{x}\|^2 + \|\mathbf{y}\|^2 \quad (4.2.12)$$

Thus, the following is obtained

$$\mathbf{w}^H \mathbf{s} \mathbf{s}^H \mathbf{w} \leq \|\mathbf{C}\mathbf{w}\|^2 + \|(\mathbf{C}^H)^{-1}\mathbf{s}\|^2 = (\mathbf{w}^H \mathbf{R}_{c+n} \mathbf{w})(\mathbf{s}^H (\mathbf{R}_{c+n})^{-1} \mathbf{s}) \quad (4.2.13)$$

such that

$$SINR \leq (\mathbf{s}^H (\mathbf{R}_{c+n})^{-1} \mathbf{s}) \quad (4.2.14)$$

To maximize output SINR, denominator of 4.2.9 must be minimized. In other words, summation of clutter and noise powers at the output of adaptive filter must be minimized, while keeping the numerator fixed. Corresponding optimization problem can be written as;

$$\min_w \mathbf{w}^H \mathbf{R}_{c+n} \mathbf{w} \quad \mathbf{w}^H \mathbf{a}(\theta_s) = 1 \quad (4.2.15)$$

The solution to this optimization problem can be found as

$$\mathbf{w} = (\mathbf{R}_{c+n})^{-1} \mathbf{a}(\theta_s) \quad (4.2.16)$$

The optimal weights are found as;

$$\mathbf{w} = \mathbf{R}_x^{-1} \mathbf{a}(\theta_s) \quad (4.2.17)$$

\mathbf{R}_x^{-1} represents covariance of interference signals. Since noise and clutter signals are uncorrelated, covariance of their summation is equal to the sum of covariance matrices of the estimated clutter and noise.

The output of filter is obtained as;

$$z = \mathbf{w}^H \mathbf{x} \quad (4.2.18)$$

Overall, optimum clutter suppression filter requires adequate knowledge about clutter covariance matrix. Thus, the covariance matrix of estimated parameters is used to obtain optimum filter coefficients for adaptive detection.

It will definitely be a preferable option over pulse cancellers. It eliminates the sensitivity of antenna rotation. Moreover, detector design can be accomplished effortlessly.

Generally, during operation, the interference-plus-noise covariance matrix \mathbf{R}_{c+n} is unknown. Thus, data sample covariance matrix can be found such that;

$$\hat{\mathbf{R}}_x = \frac{1}{K} \sum_{k=1}^K \mathbf{x}(k)\mathbf{x}^H(k) \quad (4.2.19)$$

where K is number of snapshots, secondary range cells.

In order to compare the detection performances, Normalized Output SINR versus target velocity is examined. In Neyman–Pearson theory aims to maximize P_d by the likelihood ratio test; however, the test uses unknown covariance matrix, such that an optimum detection criterion for practical scenarios cannot be found. Thus, instead of finding P_d , estimation loss is observed with Normalized Output SINR.

$$NSINR_{out} = \frac{SINR_{out}}{SINR_{in}} = \frac{|\mathbf{w}^H \mathbf{a}(\theta_s)|^2}{\mathbf{w}^H \mathbf{R}_{c+n} \mathbf{w}} \quad (4.2.20)$$

4.3 Proposed Detector

Previously mentioned, optimum and sub-optimum detectors are obtained by assuming the interference matrix is perfectly known. Unfortunately, it is not available in most cases. A homogeneous environment is assumed. Thus, received clutter signals will be independent and identically distributed.

$$\hat{\mathbf{R}}_x = \hat{\mathbf{R}}_y + \sigma_n^2 \mathbf{I} \quad (4.3.1)$$

If more than one clutter signal is received, a composite clutter case, a superposition approach is applied. If N is the number of clutter signals and $\hat{P}_i, \hat{w}_i, \hat{\sigma}_i^2$'s are estimated moments;

$$\hat{\mathbf{R}}_y = \sum_{i=1}^N \hat{\mathbf{R}}_{y_i}(\hat{P}_i, \hat{w}_i, \hat{\sigma}_i^2) \quad (4.3.2)$$

For each clutter;

$$\hat{\mathbf{R}}_{y_i}(\hat{P}_i, \hat{w}_i, \hat{\sigma}_i^2) = \hat{P}_i \mathbf{A}(\hat{w}_i) \mathbf{B}(\hat{\sigma}_i^2) \mathbf{A}^*(\hat{w}_i) \quad (4.3.3)$$

in which

$$\mathbf{A}(\hat{w}_i) = \text{diag}\left(\left[1 \quad e^{j\hat{w}_i PRI} \quad e^{j2\hat{w}_i PRI} \quad \dots \quad e^{j(m-1)\hat{w}_i PRI}\right]\right) \quad (4.3.4)$$

and

$$\mathbf{B}_{k,l} = e^{-2\pi^2 \hat{\sigma}_v^2 (k-l)^2 PRI^2} \quad (4.3.5)$$

The transversal clutter suppression filter is calculated using optimal filter equation such that,

$$\mathbf{w} = \hat{\mathbf{R}}_x^{-1} \mathbf{s} \quad (4.3.6)$$

With estimated correlation matrix, the optimal weights are found as;

$$\mathbf{w} = \hat{\mathbf{R}}_x^{-1} \mathbf{a}(\theta_s) \quad (4.3.7)$$

Thus, mismatch case Normalized Output SINR will be

$$NSINR_{out} = \frac{|\mathbf{w}^H \mathbf{a}(\theta_s)|^2}{\mathbf{w}^H \mathbf{R}_{c+n} \mathbf{w}} \quad (4.3.8)$$

The clutter signals are generally far stronger than signals received from targets of interest.

4.4 Simulations

To compare the performance of different adaptive algorithms with each other, some standard benchmarks are required. Normalized Output SINRs are measured for various scenarios. Common values and assumptions in all scenarios listed below;

- A Coherent Processing Interval (CPI) consists of 16 pulses.
- In a CPI, PRI does not change and it is 0.0001 sec.
- Target Doppler frequency is known and remains at the same value during a CPI. In other words, the target Doppler frequency is constant during the dwell time.
- Number of snapshots (secondary range cells) is 5.
- $\sigma_{v_{sea}}$ is 1.512 m/sec and $\sigma_{v_{rain}}$ is selected as 3.108 m/sec.

- $\sigma_{v_{init}}$'s are 0.5 m/sec.
- Fletcher-Reeves optimization algorithm with Three Point Line Search method with 50 iterations and 10^{-7} tolerance is implemented to find the maximum of Log-likelihood function.
- Sufficiently large number (≥ 50) of Monte Carlo samples are generated for each test case.
- The pulse train is uniformly spaced in time.

Note that, except investigation of spread knowledge effect on detection performance, the Normalized Output SINRs of detectors are obtained when Doppler mismatch exist between filter and target steering vectors. While target can take any Doppler value, Doppler steering vector, used during calculation of filter coefficients, is quantized at the normalized Doppler frequencies $\{\frac{k}{P}\}_{k=0}^{P-1}$. In other words, detectors assume target Doppler is located at the center of each Doppler bin which is a mismatch. While target Doppler can take infinite number of values, filter is finite.

4.4.1 Distant Clutters

As the distinction between clutter Doppler frequencies increases, the estimation performance is expected to be improved. Thus, first of all, spectrum composed of two relatively isolated clutters is estimated. Sea and rain velocities are selected as 3.872 and 26.178 (m/sec). CNR's are taken as 50-50 dB.

4.4.1.1 The Importance of Spread Knowledge on Detector Performance

In order to observe the effect of spread estimation on detection performance, filter coefficients, calculated from several spread values and Burg estimated frequencies, are used to filter received signal for non-contaminated target scenario. Only Burg method is used to estimate Doppler frequencies. Spreads are selected as 0, 0.5, 1 and 1.5 (m/sec). The detector performances can be seen in Figure 4.1. Note that, quantization of filter coefficients are not implemented for this scenario.

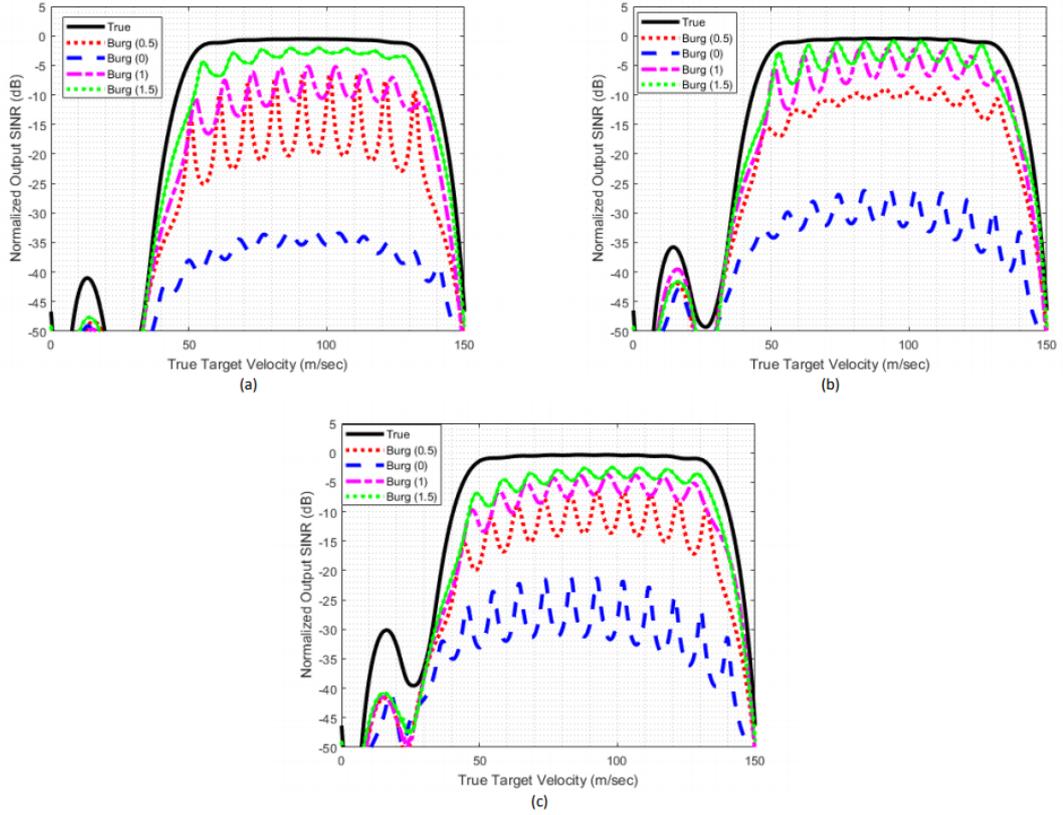


Figure 4.1: Normalized Output SINR for different spread values assumptions
(a) $CNR_{rain} = CNR_{sea} = 50\text{dB}$ (b) $CNR_{rain} = 40\text{dB}$ $CNR_{sea} = 50\text{dB}$
(c) $CNR_{rain} = 30\text{dB}$ $CNR_{sea} = 50\text{dB}$

As it can be observed, the difference between optimal detector, designed with true values of parameters, and detectors working with assigned spreads is remarkable. Thus, the key behind optimum detection is the spread knowledge.

4.4.1.2 Number of Turbo Iterations

The adaptive detector performance is measured when target contamination does not included in received echo. The covariance matrix is calculated from parameters estimated with Turbo having different number of iterations. Two and three iterations include recursive estimation which can be seen in Figure 3.3.

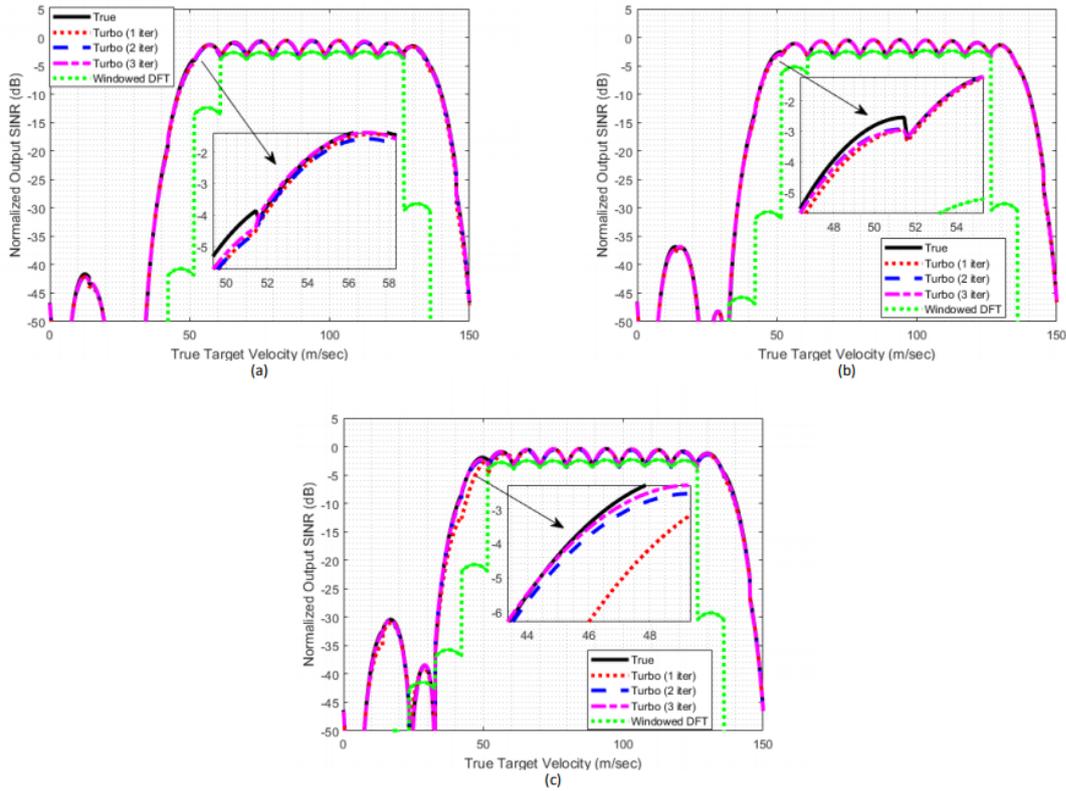


Figure 4.2: Normalized Output SINR of different number of Turbo iterations
(a) $CNR_{rain} = CNR_{sea} = 50\text{dB}$ (b) $CNR_{rain} = 40\text{dB}$ $CNR_{sea} = 50\text{dB}$
(c) $CNR_{rain} = 30\text{dB}$ $CNR_{sea} = 50\text{dB}$

When power values of clutters are relatively close, the performances of one, two and three iterations are similar. However, when the power difference is 20 dB, the Normalized Output SINR of three iterations Turbo is much closer to the optimal case than the other ones. Thus, three Turbo iteration estimation method is selected as Turbo algorithm for clutters, located separately in power spectrum.

4.4.1.3 Performances of Algorithms When Target Contamination is Absent

After selecting the number of Turbo iterations, the overall suggested algorithms are compared when target contamination is absent. Optimal case is jointly plotted with SML and Turbo methods having initial frequencies estimated by Burg or taken as

zero. Target Doppler is included during Normalized Output SINR calculation of adaptive filter. Moreover, the conventional detector performance is compared with implemented adaptive methods.

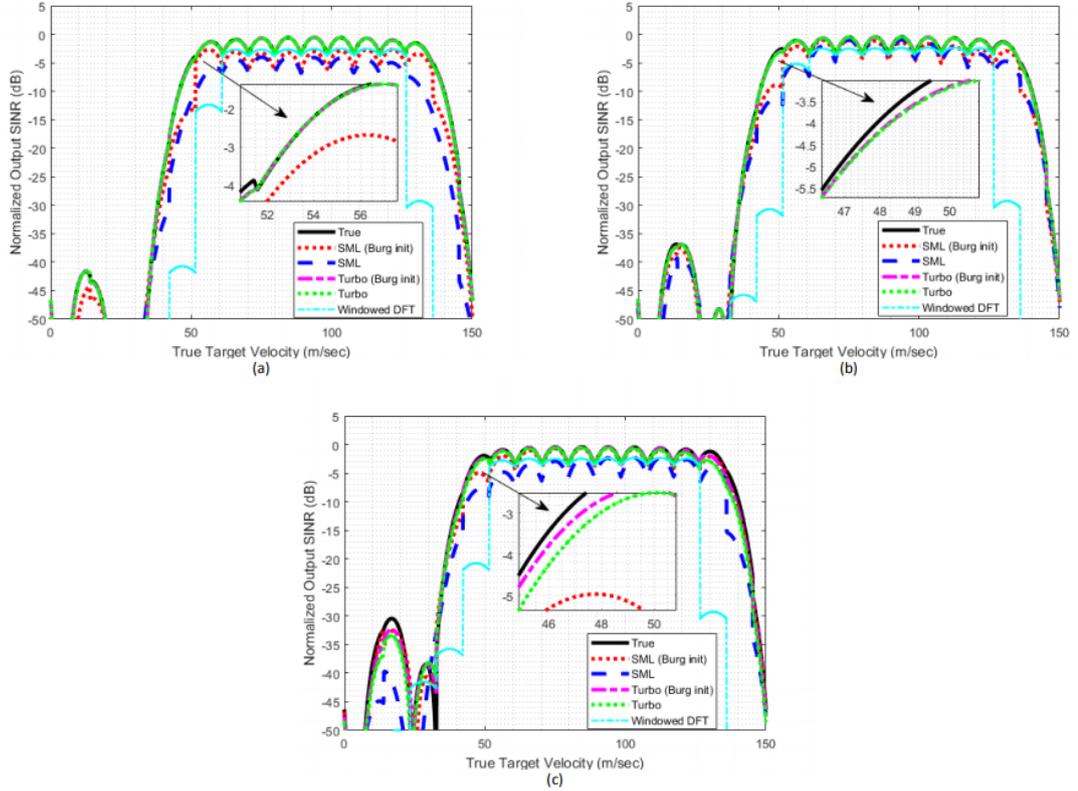


Figure 4.3: Normalized Output SINR for estimation methods (a) $CNR_{rain} = CNR_{sea} = 50\text{dB}$ (b) $CNR_{rain} = 40\text{dB}$ $CNR_{sea} = 50\text{dB}$ (c) $CNR_{rain} = 30\text{dB}$ $CNR_{sea} = 50\text{dB}$

From Figure 4.3(a) and (b), it is observed that zero initiation of Turbo frequencies gave as compromise performance as wisely (Burg) initiated one. However, 20 dB CNR difference illustrated that Turbo initial frequencies must be selected by Burg. While initiation of Turbo frequencies become crucial in noticeable power difference scenarios, SML accuracy is affected positively with wise initiation in all scenarios. Generally, the conventional detector performance stands between SML methods. For all cases, Turbo methods have better detector performance than SML methods. Thus, from now on Turbo method, initiated with Burg frequency estimates, is chosen as the estimation algorithm.

4.4.1.4 Performance of the Designed Algorithm with Target Signal Contamination Presence

Finally, the performance of the selected algorithm is observed, when target contamination is included in received echo. Received radar signal also consists of two clutters having relatively distinct Doppler values. Target SNR is varied between 0 and 20 dB. The averaged performance is compared at each SNR value and the conventional radar detector.

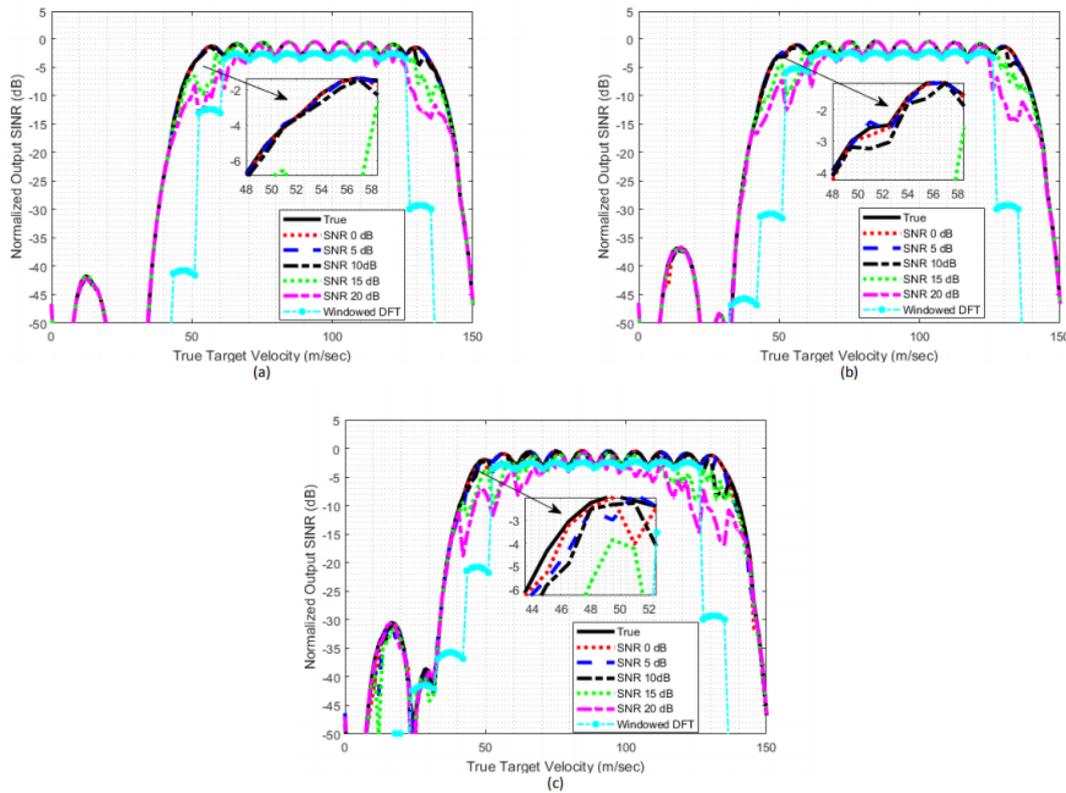


Figure 4.4: Normalized Output SINR with target contamination (a) $CNR_{rain} = CNR_{sea} = 50\text{dB}$ (b) $CNR_{rain} = 40\text{dB}$ $CNR_{sea} = 50\text{dB}$ (c) $CNR_{rain} = 30\text{dB}$ $CNR_{sea} = 50\text{dB}$

The difference between optimal and estimated case become noticeable when target contamination signal will have SNR, greater than 15dB. That is to say, if the range sidelobes after fast time matched filtering (when target is present) due to non-zero autocorrelation lags is 15 dB above the noise level, the performance of proposed

detector starts to degrade. By using suggested filter in [48], target free snapshots can be obtained. It is a fast time pre-processing technique, aiming the minimization of loss due to target presence at secondary range cells. In the absence of target, detector performances are given in previous results.

4.4.2 Closely Spaced Clutters

The clutter velocities are selected as relatively close to each other. Sea and rain velocities are taken as 3.872 and 12.345 (m/sec).

4.4.2.1 The Importance of Spread Knowledge on Detector Performance

Similar to the distant clutter case, the effect of spread estimation on detection performance is observed for closely spaced clutter scenario when no target is present. The detector performances can be seen in Figure 4.5. Note that, quantization of filter coefficients are not implemented for this scenario.

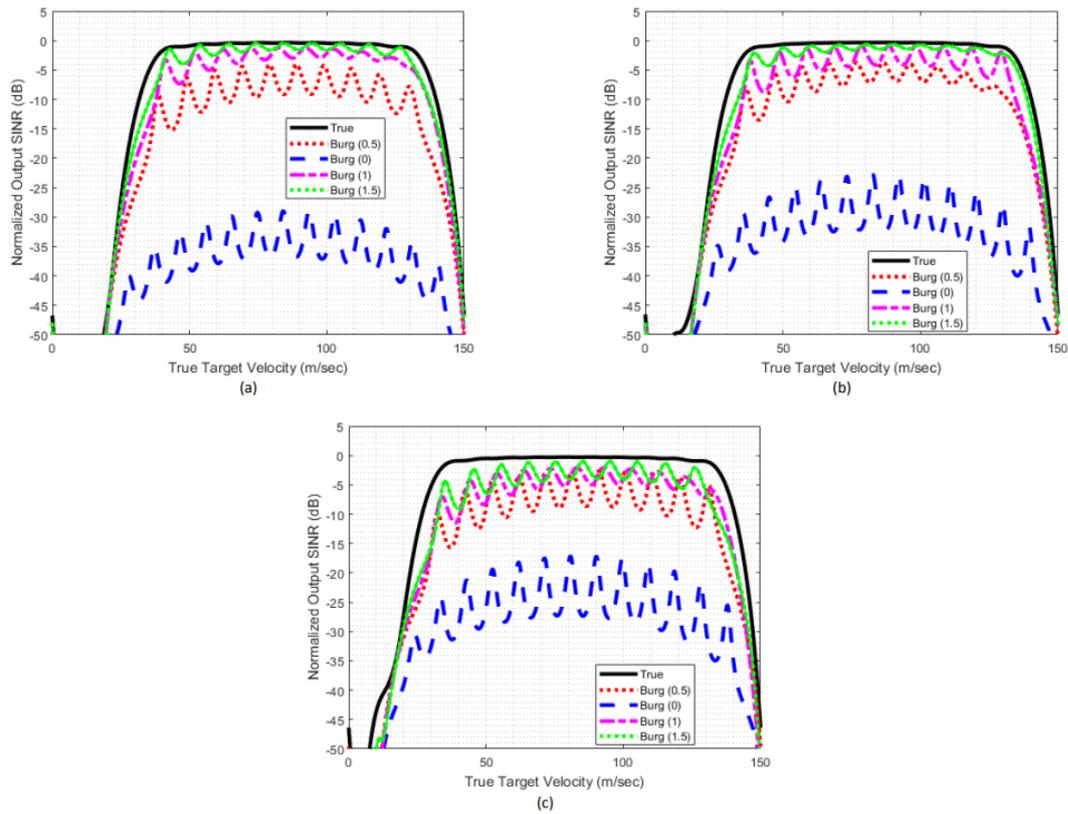


Figure 4.5: Normalized Output SINR for different spread values assumptions
 (a) $CNR_{rain} = CNR_{sea} = 50\text{dB}$ (b) $CNR_{rain} = 40\text{dB}$ $CNR_{sea} = 50\text{dB}$
 (c) $CNR_{rain} = 30\text{dB}$ $CNR_{sea} = 50\text{dB}$

The importance of spread knowledge is also observed for closely spaced clutters from significant difference between Normalized Output SINRs. The accurate spread knowledge requirement is verified.

4.4.2.2 Number of Turbo Iterations

Similar to distinct velocity clutter case, the effect of Turbo iteration on adaptive detection is observed in the absence of target.

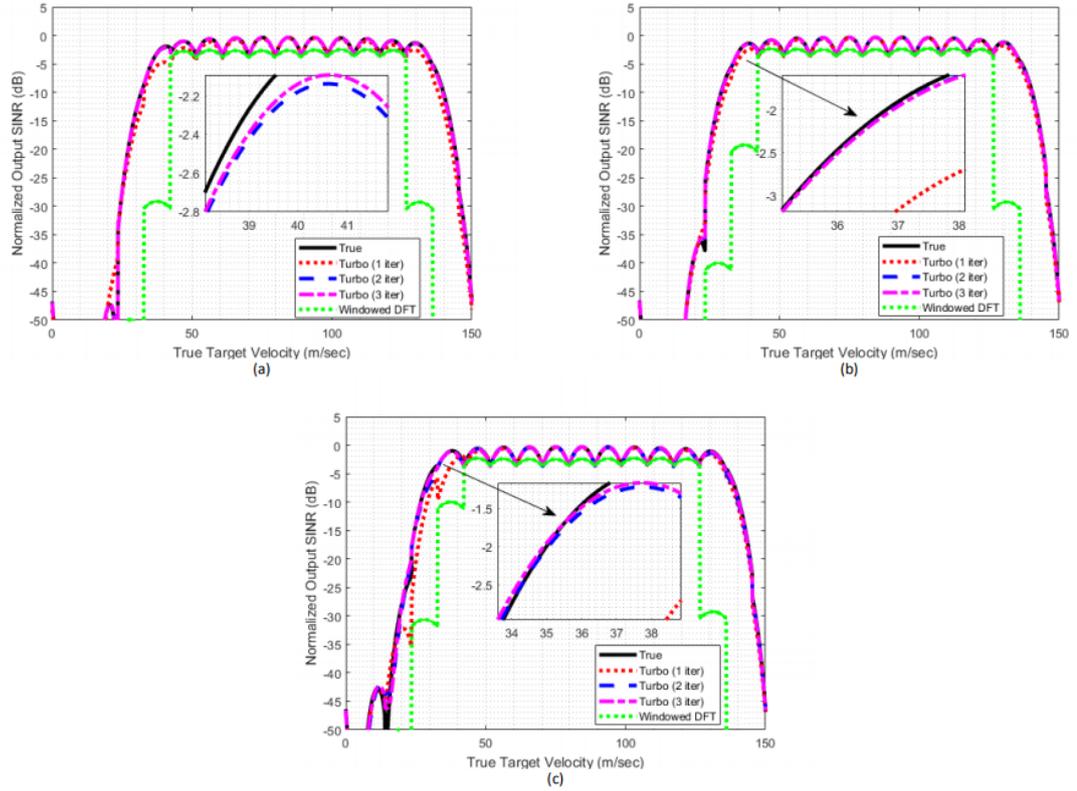


Figure 4.6: Normalized Output SINR of different number of Turbo iterations
 (a) $CNR_{rain} = CNR_{sea} = 50\text{dB}$ (b) $CNR_{rain} = 40\text{dB}$ $CNR_{sea} = 50\text{dB}$
 (c) $CNR_{rain} = 30\text{dB}$ $CNR_{sea} = 50\text{dB}$

While 1 Turbo iteration cannot give sufficient performance, 2 and 3 iteration performances are similar. For coherency and accuracy, Turbo iteration is selected as 3.

4.4.2.3 Performances of Algorithms When Target Signal Contamination is Absent

As the second case, Turbo and SML methods are compared with the conventional detector. The effect of initiation on detection performance is also observed with each estimation algorithm.

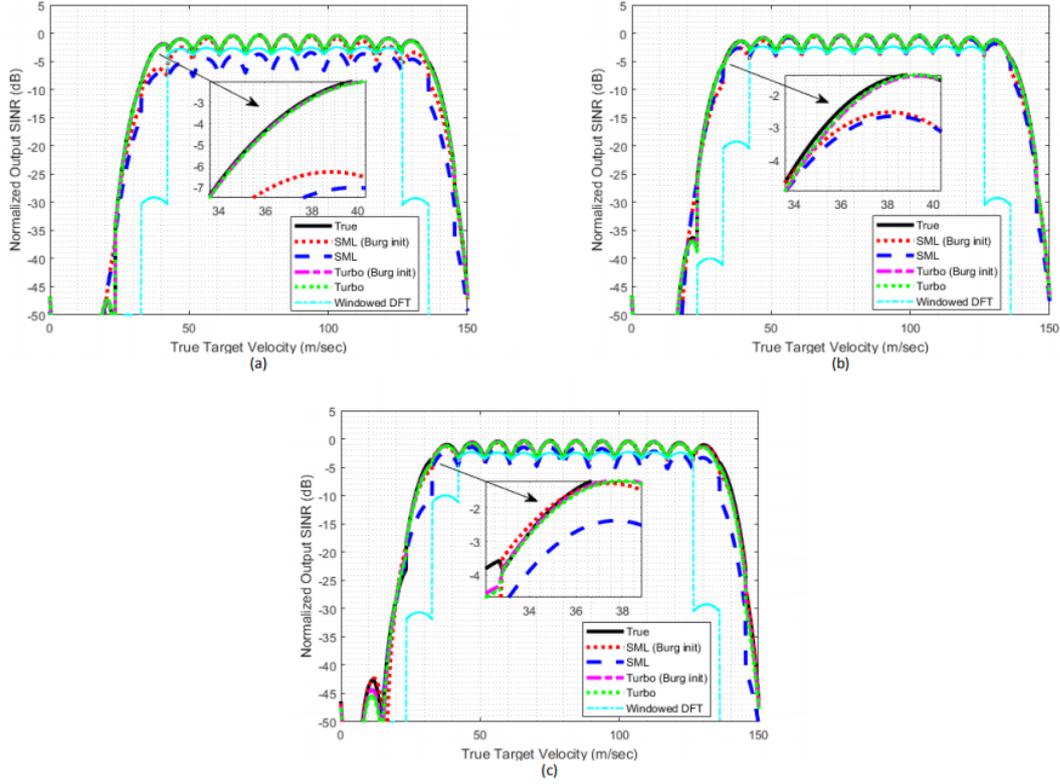


Figure 4.7: Normalized Output SINR for estimation methods (a) $CNR_{rain} = CNR_{sea} = 50\text{dB}$ (b) $CNR_{rain} = 40\text{dB}$ $CNR_{sea} = 50\text{dB}$ (c) $CNR_{rain} = 30\text{dB}$ $CNR_{sea} = 50\text{dB}$

From Figure 4.7 (a) and (b), the superiority of Turbo method over SML can be seen clearly. While, initiation has no effect on detection performance of Turbo methods, SML shows important improvement with Burg initiation. All estimation methods are more effective than the conventional approach. The best choice as an estimation algorithm is Turbo method with 3 iterations initiated by Burg. Additionally, it is observed that when clutters are closely spaced the degraded parameter estimation performance in Figures 3.12 and 3.13 does not affect the Normalized Output SINR.

4.4.2.4 Performance of the Designed Algorithm with Target Signal Contamination Presence

As a final case, target contamination is included in received echo. The detector performances of the selected algorithm are observed. Two clutters in received echo have relatively close velocities. Target SNR is increased step by step from 0 to 20 dB. The averaged performance is compared at each SNR and the conventional radar detector.

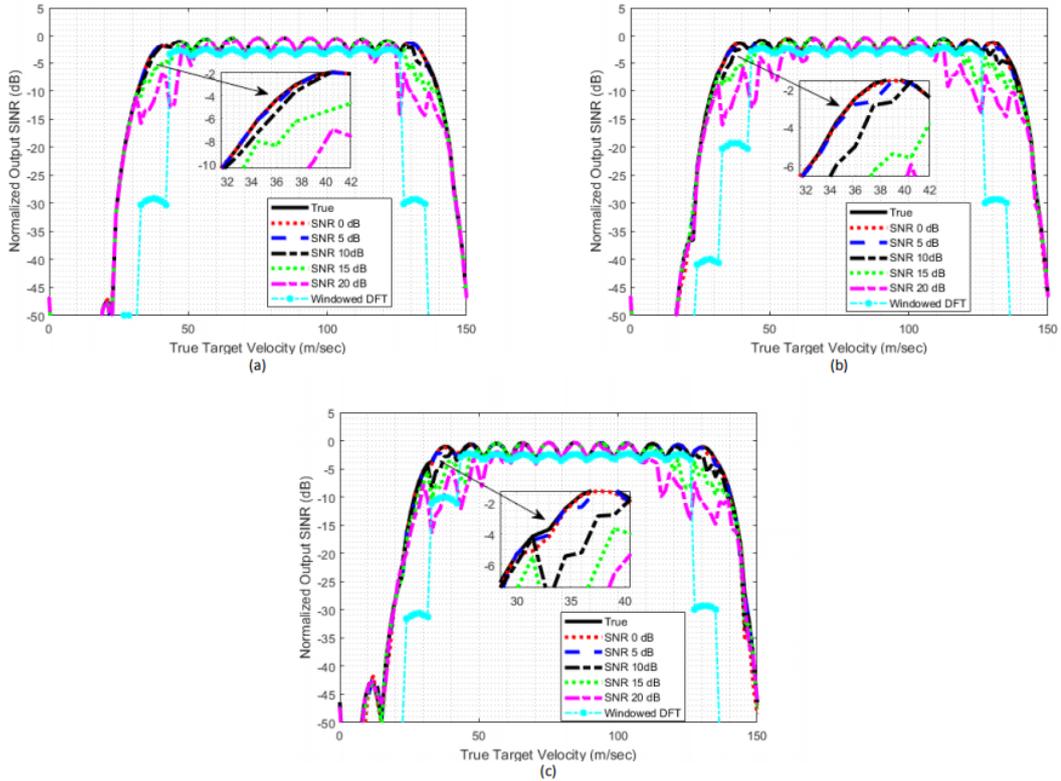


Figure 4.8: Normalized Output SINR with target contamination (a) $CNR_{rain} = CNR_{sea} = 50\text{dB}$ (b) $CNR_{rain} = 40\text{dB}$ $CNR_{sea} = 50\text{dB}$ (c) $CNR_{rain} = 30\text{dB}$ $CNR_{sea} = 50\text{dB}$

Until SNR reaches 10 dB, the detector performances are higher than the conventional detector. 15 and 20 dB SNRs deteriorates the adaptive detector performance. Same fast-time processing technique with distant clutter case can be used to obtain target free cells.

CHAPTER 5

CLUTTER CLASSIFICATION METHODS

In recent years, the potential capabilities of the modern radars have become tremendous, with rapid advances in electronic and software technologies. For instance, optimization of the detection and track performance is brought to the agenda by obtaining high signal interference ratios with high processing speeds. Following these developments, the expectations from radars have increased in parallel.

Most of the conventional radars work with predetermined beam designs and waveforms. However, working environment cannot be predetermined. Consequently, the radar must make use of resources partially autonomous. Therefore, Artificial Intelligence (AI) concept comes to scene in order to rule the working of radar by determining resource allocation. The use of AI in radars applications is not limited with the resource management. For example, it can be used to develop more intuitive methods during beam and waveform selection or to classify radar targets [35]. In the scope of this work, Artificial Neural Networks will be investigated for clutter classification.

The identification of moving targets and clutter signals improves radar detection performance and increases the control over environment. In order to classify clutters, Neural Network structures are studied. The design procedure of the neural networks is explained with theoretical background information. Their performance is illustrated with experimental results.

To summarize, chapter 5 starts with background information about Artificial Intelligence and related concepts. Secondly, proposed Neural Network structure for the classification problem is explained. The design is tested with generated data sets. Finally, the proposed Neural Network architecture is experimented with estimated clutter parameters.

5.1 Artificial Intelligence Background

Artificial Intelligence (AI) is a general field that encompasses machine learning and deep learning.

Deep Learning concepts enable to design computer systems having the ability of learning and improving itself from experience without being explicitly programmed. In other words, computers gain some human like intelligence. It is an application of AI. Deep stands for the idea of successive layers of representations so that deep learning is consecutive. How many layers contribute to a data model is called the *depth* of model. Depth of neural network can be increased to improve approximation quality. Deep learning deals with the problem of learning hierarchical representations with a single algorithm. Traditional approach is to use shallow networks which are simple, generic and not hierarchical. Unlike traditional machine learning algorithms such as shallow neural networks or Support Vector Machine (SVM), deep learning approaches require no human intervention during the training process. Additionally, convenient representation of a system must be hierarchical.

Deep learning is based on Artificial Neural Networks (ANN), in short only Neural Networks (NN). Thus, it is better to use Neural Networks. In other words, deep learning models are based on multi-layer neural networks, and each layer usually learns a set of features at a different scale or complexity. Deep learning methods are remarkably successful because of their high computational power and ability to deal with large datasets.

In short, while classical programming uses data and rules to get answers of the problems, machine learning approaches gives rules to solve problem by using data and answers.

5.1.1 Artificial Neural Networks

Artificial Neural Networks are inspired from neuroscience. They imitate the neural function of the human brain. As it can be understood from its name, it is a network composed of artificial neuron units. Units have connections in between.

Perceptron As stated, ANNs are inspired from biological neural networks such that the perceptron is an abstract model of a single neuron. The perceptron is a linear classifier defined by *weights*, w_i 's and bias, b . It was suggested by Rosenblatt in 1958 [49]. A neuron's dendrites are modelled by weights. Weights are multiplied with input values and summed up to get output, z . The corresponding perceptron model can be seen in Figure 5.1.

$$z = \sum_{i=0}^n x_i w_i + b \quad (5.1.1)$$

Network uses perceptron learning rule to adjust its weights such that weight adjustment is done by calculating the error between desired and actual output of the network.

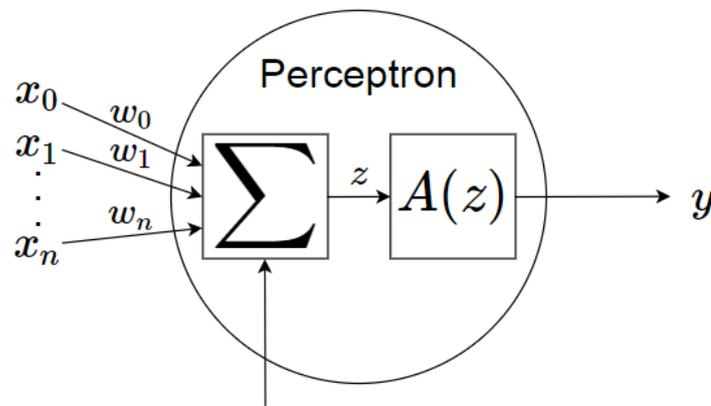


Figure 5.1: Perceptron with n inputs, one output and an activation function

Activation Functions and Nonlinearity After weighting, output will be fed into the activation function $A(z)$. Since the weighting is same as computing linear combination of inputs, the model will be linear independently from number of layers. If neural network units have no activation function or only a linear activation function, then only linear problems can be solved [50]. Thus, non-linearity must be achieved by activation functions. In other words, activation function must not be linear so that NN has the capability to solve non-linear problems.

Briefly, the activation of a neuron in layer k is a linear mapping of the neuronal activations of layer $k - 1$, followed by a non-linear function. The most common activation function is sigmoid.

Sigmoid Function Sigmoid function output is restricted between 0 and 1, which can be seen in Figure 5.2.

$$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}} \quad (5.1.2)$$

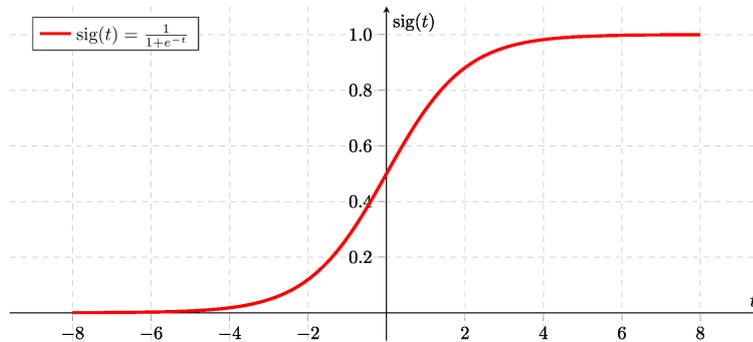


Figure 5.2: Sigmoid Function

Sigmoid function may give zero gradient values which increases the convergence time of weight adjustments. Thus, additional to the non-linearity requirement, activation function must be differentiable. Exponential Linear Unit (ELU), Rectified Linear Unit (ReLU) and Softmax are other alternatives for activation function selection. If sigmoid will not be efficient, one of them can be implemented.

Feed-Forward Neural Network (Multilayer Perceptron) As stated, the typical ANN has a structure with layers. Each layer of the architecture contains some amount of neural units. A variety of networks can be listed depending on the kind of interconnection. If connections at same layer or with the back ones, it is a *recurrent neural network*. If the connection is on one direction to the output layer, then the network is called *feed-forward neural network*. Backward loops are not used in feed-forward networks.

Feed-forward neural networks (FF NN), or Multilayer Perceptrons (MLP); are networks having series connected perceptrons. It consists of layers with specific names.

- Input layer: holds the values and distribute them to the next layer
- Output layer: final state of the NN is read

- Hidden layers: layers between input and output, connected using weighted links to the higher levels

In Figure 5.3, a feed-forward neural network having 3 nodes in input layer, 4 nodes at output layer and 4 nodes at one hidden layer is illustrated.

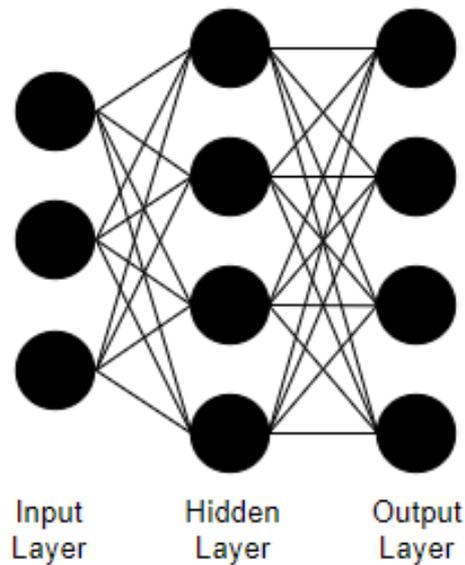


Figure 5.3: A Feed-forward Neural Network with one hidden layer

The perceptrons located at hidden and output layers are called as *units*. The *widths* of layers are defined by number of units. A feedforward neural network involves multiple layers of hidden neurons, with activations. Thanks to non-linearity of activation functions, the network can approximate every continuous function with high accuracy, which is a potent property. As the network become deeper, its performance generally increases. The numbers of input and output nodes are determined according to the application data. Moreover, if number of feed forward neural network hidden layers is greater than one, it is called a *deep* neural network.

Feature Extraction As data sizes enhance, direct feeding of data into network become impractical or even impossible for some problems. Thus, some kind of preprocessing must be performed. Feature extraction is preprocessing of overall data into a smaller set. Extracted features will be the inputs of machine learning algorithm.

Feature is any data extractable from measurements. The selection of features is crucial since classification, the assignment of input into one of the pre-defined classes, will be based on the selected features. While selecting features, values having much more information about input data are desired. A classifier partitions feature space into class labelled decision regions. If class assignments are done uniquely, decision regions must cover all feature space and be disjoint. Generally as a classification strategy, feature vectors are assigned according to decision regions.

Training Training is the learning phase of the neural network. During training, the weights will be adapted and learned. Different learning approaches in AI are known as supervised, unsupervised and reinforcement.

First one is *supervised learning*, in which the machine learning training set is labelled. In other words, each pair of input data mapped to a labelled output. The goal is to find function or rule, which maps the input to the desired output label. Even data encountered for the first time must be correctly mapped. Classification and regression problems are sub-field of supervised learning. In classification problems, input of an example is divided into two or more classes such that mapping will be onto these classes.

In *unsupervised learning*, training set has no label. The algorithm finds a structure in the data. Clustering models are necessary. Clustering is grouping values with respect to similar properties.

In *reinforcement learning*, the goal is pre-specified. Unlike supervised learning, answer about how to accomplish the goal is not known. The system learns the way depending on its own previous experiences and outcomes while doing a similar kind of a job. It is all about making decisions sequentially.

Feed-forward neural networks are sensitive to training parameter selection, which must be comprehensive. Moreover, over-training may occur if a hidden layer is larger than necessary. Thus, Curse of Dimensionality and Dimensionality Reduction concepts are studied in order to obtain optimal training data.

Curse of Dimensionality The number of training data is an exponential function of feature array dimension [51]. Bellman gives the phenomenon a name as “curse of

dimensionality” [52]. Partitioning the input space and labelling each partition is an inefficient technique. As the dimension of data increases, finding practical training data set, fulfilling the requirement, will become critical. Most common classifiers make estimation of unknown parameters and use them in the class-conditional densities as if they are true parameters. If the sample size is fixed, the accuracy of parameter estimates decreases with increasing number of features. Thus, the performance of classifier may degrade because of the corresponding increase in the number of unknown parameters. Enhancement of the sample size can be a solution, but it will not be practical generally. On the other hand, data dimensionality reduction can give optimal results.

Dimensionality Reduction Dimensionality reduction is possible by searching for the optimal features. The optimality criterion is to decrease classification error. The most informative subset of the data must be selected, *feature selection*. Alternatively, new features can be generated with transformations or combinations of the original data set, *feature extraction*. The choice of which one will be used depends on application. For clutter classification, feature extraction is utilized.

Loss Functions The learning of neural network is actually a loss reduction. Loss (cost) functions are used to represent the quality of estimation. Classification problem performance measure is accuracy, which describes the ratio of correct classified examples with respect to all results. A loss function maps variables to a single output, loss. Loss stands for the discrepancy between the function output and the expected value. Choice of the loss function depends on the learning problem. The most commonly used loss function is Root Mean Square Error (RMSE) or MSE. For example, MSE loss function is a useful for regression problems, since MSE punishes the difference in outputs. Thus, for tasks like regression which represent exact values in output nodes, RMSE or MSE is better.

$$L = RMSE(\hat{y}_n, y_n) = \sqrt{\frac{\sum(\hat{y}_n - y_n)^2}{N}} \quad (5.1.3)$$

However, for classification problems, MSE is not convenient. Cross-entropy loss is more advantageous. First of all, calculation of N number of class probabilities is

performed. Then, compared with expected output such that loss can be written as

$$L = CE(\hat{y}_n, y_n) = - \sum_{i=1}^N y_{n,i} \log(p(i|\hat{y}_n)) \quad (5.1.4)$$

Minimizing Loss Neural network design parameters must be optimal in order to provide better approximations. Thus, again an optimization problem must be solved by using an algorithm minimizing the loss function. Popular choices are Stochastic Gradient Descent (SGD), Root Mean Square (RMSProp) and Adam. SGD is used, an iterative learning method that starts with some initial randomly selected parameter values. It is same as method in A, Steepest Descent Method. Given $\theta \in (W \cup B)$ is the parameter that will be optimized such that it will give lowest loss function value. If μ is *learning rate* and $\nabla_{\theta} L(f(x), y)$ is partial derivative of loss function with respect to parameter θ , the learning rule assigning the new value of parameter θ for a simple example would be

$$\theta^* = \theta - \mu \nabla_{\theta} L(f(x), y) \quad (5.1.5)$$

By performing adequate iteration, SGD aims to find a global minimum for the loss function, with given data and initial values of parameters. This involves computing the gradient of the error term with respect to the parameters of each layer, a procedure known as *back propagation*. It is a popular algorithm because it is basic and effective. The aim is to find a function mapping of input data to the output data during training. The average loss over the complete training set is computed by the loss function. Back propagation algorithm consists of four stages,

1. Calculation of network outputs
2. Computation of error between desired outputs and the network outputs with respect to objective function
3. Computation of update values for error minimization
4. Update of parameters with previously updated values [53]

The choice of learning rate, μ , is important. If it is too small, the algorithm will take too much time to find optimum point and may stuck in a local minima. On the other

hand, if it is too large, the optimal solution can be missed. Thus, it must be chosen wisely. Additionally, starting point is crucial, since it determines the start of path. During training, iteration is defined as learning step. Each of the iterations describes one update with the gradient descent. Epoch is one passing of the algorithm over the complete training set.

Regularization The ability of classifying unseen data is done by generalization of the data. The training data set cannot include every possible instance of the inputs, because of that limitation the learning algorithm must be able to generalize in order to handle unseen data points [50]. During testing, generalization error must be measured. If it is high, two reasons must come to mind. First one is *under-fitting*, resulting from capacity deficiency. Thus, both of training and test phases will have high errors. Second one is *over-fitting*, over-much capacity, in which training error will be low, but test error will be significant. In order to use the right capacity model for the problem, regularization is performed.

Most common regularization methods aim to prevent over-fitting of neural networks. Over-fitting occurs when the weights converge for the training dataset. In other words, the network performance for the training dataset is high, but its generalization cannot be performed to work with any other data. Generally, L1 or L2 regularizations are used for regularization. L2 regularization is implemented when necessary by adding an extra term to the cost function.

5.2 Proposed Neural Network

In many applications, Artificial Neural Networks (ANNs) are preferred because of their ability of modelling non-linear statistical data. According to [54], they have shown satisfying performance in the field of classification. As stated, discrimination and classification of radar signal will significantly improve radar detection performance. Additionally, the clutter identification is an important knowledge for radar operation. Thus, ANNs are implemented to classify clutter data.

Classifiers use pattern matching to determine a closest match in other words they estimate the owner class of the data. A neural network classifier is designed and used

to distinguish several common types of radar clutter returns including weather, sea, birds.

ANNs are composed of 3 entities: network topology, characteristic of each neuron and training strategy.

An artificial neural network is simulated to observe its working principle and ensure that it gives preferable results in the aspect of clutter examination. In the scope of work, an optimal feed-forward neural network design is aimed because of following advantages:

- Learning ability from input data
- Non-parametric solution
- Ability to solve non-linear problems
- Comparing to classical methods, its clutter classification success is 80% higher [54].

The work flow for the general neural network design process consists of five primary steps:

1. Collection of data
2. Creating the network
3. Configure the network and Initialize the weights and biases
4. Training of the network
5. Utilization of the network

Note that, final validation must be carried out with an independent data set. System design cycle is shown in Figure 5.4.

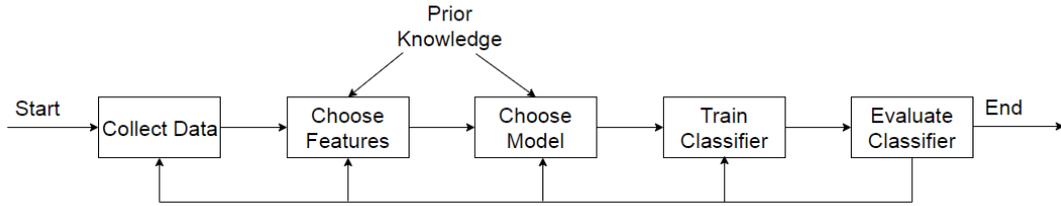


Figure 5.4: Classification System Design Cycle

A class is defined as a set of patterns known to originate from the same source. In our case, sources are clutter types. Input will be features extracted from radar data. Thus, overall problem schema can be seen in Figure 5.5.

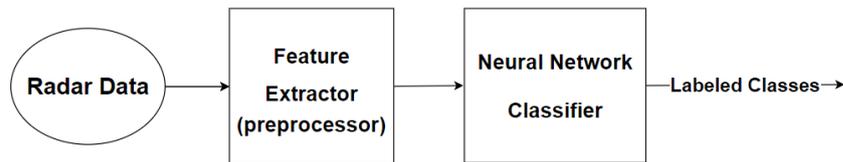


Figure 5.5: Classifier of Radar Signals

5.2.1 Pre-Processing and Feature Extraction

First of all, training and test data sets must be generated. Since the handling of whole radar signal will be time-consuming, a feature extraction approach must be implemented. For feature selection, one of papers from S. Haykin[37] is used. Features giving most information about received signal are selected.

As stated in Chapter 2, radar clutter may be modelled as a relatively low order AR process [46]. One can generate AR parameters of included clutter types which are investigated in Chapter 2, in detail. A multi-dimensional PEF will be used to estimate the characteristics of data. As the dimensionality of data increases, prediction capacity will improve. One of papers of Haykin [46] shows that; a decent feature set for a statistical Bayesian classification can be generated with the help of reflection coefficients. Reflection coefficients are resulting from the lattice implementation of the prediction-error filter (PEF). Moreover, radar features were extracted based

on second-order statistics. However, useful information may be contained in higher-order statistics. Thus, an effective feature selection is performed.

Phase Variations of Sample Sequence as Features Angles of reflection coefficients will be used as features. Yet mean Doppler shift, ϕ , is generally unpredictable. Thus, it must be removed to reduce randomness of estimation. Removal is done by Doppler frequency normalization, applied to all reflection coefficients.

$$\rho'_m = \rho_m \exp(-jm\phi) \text{ where } \phi = \arg(-\rho_1) \quad (5.2.1)$$

By using above equation, the neural classifier input is cleaned from the Doppler information. However, it can be used in the post-processing procedure. Additionally, magnitude of first reflection coefficient is normalized and then used.

$$u_0 = 10 \log \frac{|\rho_1|}{1 - |\rho_1|} \quad (5.2.2)$$

Note that as the $|\rho_1|$ become closer to unity, the larger value of u_0 will be obtained.

Magnitude Variations of Sample Sequence as Features With the help of Gaussian clutter assumption, second-order spectral parameters will be sufficient to obtain all statistical properties of the input data. If distribution differs from Gaussian, higher order spectral characteristics become necessary to deal with the data. For instance, aircraft (target) and ground echoes exhibit significant deviations from a Gaussian distribution. Thus, higher order statistics must be used to generate new features in order to obtain more robust clutter classification method. For example, skewness and kurtosis can also be used as features.

The *skewness* measures the asymmetry of a distribution around its mean. If distribution skewed to the right of mean, skewness will be positive. On the other hand, if the distribution decline towards left of mean, skewness will be negative. Skewness of a vector \mathbf{x} having length M ;

$$Skewness(\mathbf{x}) = \frac{1}{M-1} \sum_{i=1}^M \frac{(x_i - \mu(\mathbf{x}))^3}{\sigma^3} \quad (5.2.3)$$

Kurtosis measures the flatness of a distribution comparing to a Gaussian distribution. If it is positive, distribution is peakier around its mean with respect to a Gaussian

with same parameters. On the other hand, negative kurtosis indicates that the density is more flat around its mean than a Gaussian distribution.

$$Kurtosis(\mathbf{x}) = \frac{1}{M-1} \sum_{i=1}^M \frac{(x_i - \mu(\mathbf{x}))^4}{\sigma^4} \quad (5.2.4)$$

Since calculation of higher order statistics with video signal will be waste of time, a new time series, $\mathbf{z}(k)$; is generated by normalizing magnitude of radar video signal,

$$\mathbf{z}(k) = \frac{\mathbf{x}_{mag}(k) - E\{\mathbf{x}_{mag}(k)\}}{E\{\mathbf{x}_{mag}(k)\}} \quad (5.2.5)$$

Based on the new time series, $\mathbf{z}(k)$; skewness and kurtosis are calculated. Signal strength related features are also used. If L is the number of disjoint spatial zones and K is the number of samples taken from each zone, P_C is the average power of the center samples:

$$P_C = \frac{1}{L} \sum_{i=1}^L |x_i[\frac{K+1}{2}]|^2 \quad (5.2.6)$$

and the average value of the center-sample magnitudes:

$$X_C = \frac{1}{L} \sum_{i=1}^L |x_i[\frac{K+1}{2}]| \quad (5.2.7)$$

and P_0 is the estimated average signal power:

$$P_0 = \frac{1}{LK} \sum_{i=1}^L \sum_{k=1}^K |x_i(k)|^2 \quad (5.2.8)$$

Thus, first signal strength related feature can be selected as the normalized sample magnitude variance, P_{var} . Second one is the power difference between the center and edges of a time series, P_{dif} . The P_{var} and P_{dif} are defined as

$$P_{var} = 1 - \frac{X_C^2}{P_C} \quad P_{dif} = \frac{P_C}{P_0} - 1 \quad (5.2.9)$$

Signal to Noise Ratio (SNR) is used as the last feature.

$$SNR = 10 \log \frac{P_0}{N_0} \quad (5.2.10)$$

To sum up, feature selection plays an important role in classifying systems. First of all, Gaussian distribution parameters are found. After that, AR modelling of spectrum is derived. Magnitudes and phase of the poles and reflection coefficients will be used as inputs. They are calculated by Burg's algorithm. Then higher order statistics are considered in order to work with data deviating from Gaussian.

5.2.2 Neural Network Architecture

Since search volume of radar data is vast, the selection of NN architecture is important. Neural network structure can have multiple hidden layers with a number of nodes. Input layer is considered as layer 0. At each layer the linear combinations of previous layer's output is computed. The computation results are delivered as outputs after applying activation function. Two crucial questions come to mind: how many hidden layers and how many units of them are needed?

Numbers and distribution of neurons used at hidden layers directly affects the performance of an NN. Several topologies can be employed to find the ideal one to solve the problem, empirically. Since perfect design for application cannot be foreseen, only an approximately optimum architecture can be found after repeated tests. A Python code is implemented to simulate the proposed architecture.

If a 5 order PEF is used as the feature extractor, phase variations will be measured with $u_0, Re\{\rho_i\}, Im\{\rho_i\}$ $i = 2, 3, 4, 5$. Magnitude variations are represented by skewness, kurtosis, P_{var} and P_{dif} . The final signal-strength-related feature is SNR . Thus, in total 14 features, $[u_0, Re\{\rho_2\}, Im\{\rho_2\}, Re\{\rho_3\}, Im\{\rho_3\}, Re\{\rho_4\}, Im\{\rho_4\}, Re\{\rho_5\}, Im\{\rho_5\}, Skewness, Kurtosis, P_{var}, P_{dif}, SNR]$, will be used. Therefore, the input layer of the multilayer network consists of 14 nodes. As number of hidden layers increases, topology becomes complex. Two hidden layers will be satisfactory at the beginning of tests. Numbers of nodes in hidden layers are selected as 20 and 10. This can also be changed if it cannot give promising results. Finally, 3 output nodes are defined. The class label at the output is determined by recognizing the neuron at output layer, generating the largest value among others. The overall structure can be seen in Figure 5.6.

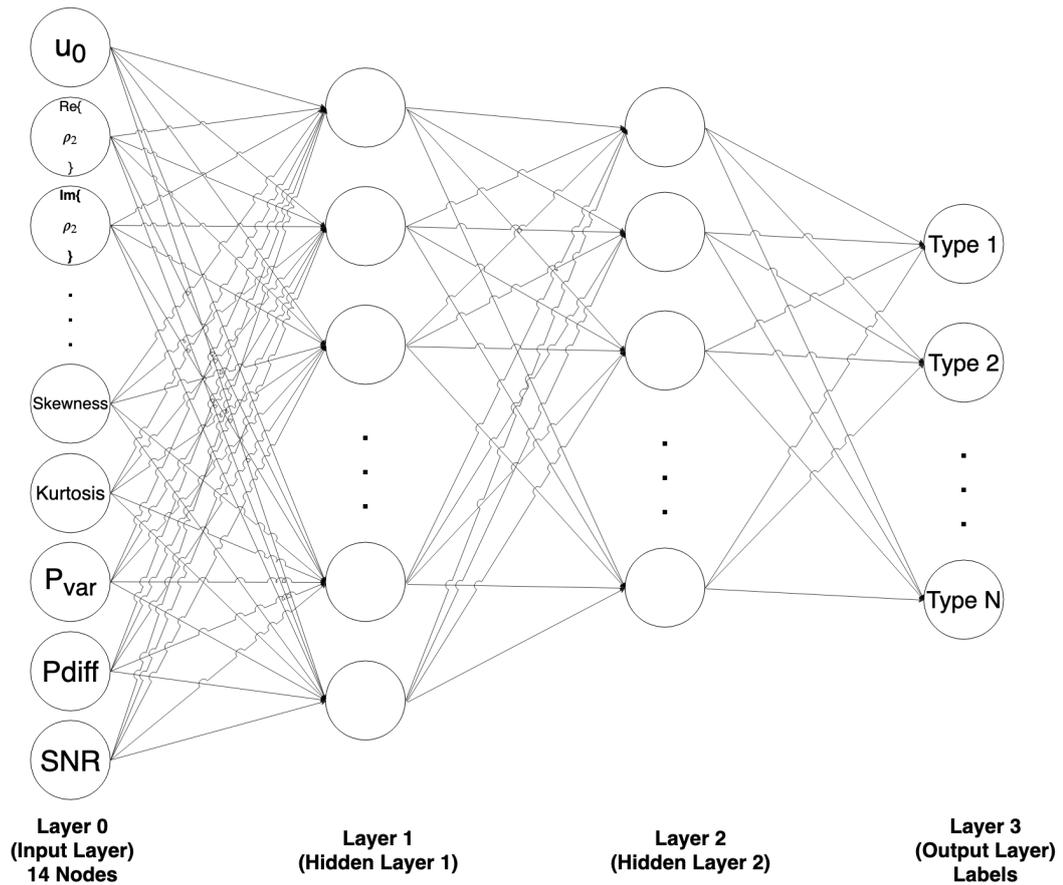


Figure 5.6: Designed Feed-forward Neural Network structure

Supervised learning without feedback is used. After initialization of weights, a modified back-propagation algorithm is used to update them. Rate of convergence is improved with the help of the modified back-propagation algorithm.

Since FF NN has slow training comparing to some others classifiers in literature, it is favorable to use gradient descent as training algorithm. Hyper-parameters must be selected before training. These parameters will not be learned during the training. Some examples of hyper-parameters are learning rate, number of layers or width of layers. To find the optimum ones, several models with different hyper-parameters can be tested such that the model with the lowest error can be chosen.

Four main strategies are available for searching for the optimal configuration. Babysitting (aka Trial and Error), Grid Search, Random Search and Bayesian Optimization. Babysitting is the most popular one among researchers, students, and hobbyists.

If this approach implemented during training, highest capacity model will be the

choice. Yet, it may result in over-fitting. Thus, training set will be further split in a validation set so that the model giving lowest generalization error will be selected one. Thus, three data sets are used, training, validation and a test set. Validation set will not be used during training. Actually, performance of validation set depends on NN selection will have a bias because of model dependence on validation set. Thus, the performance must be measured with test data set.

5.3 Experimental Results

Tests will be performed by selecting one or two of the clutter given in 5.1. The most distinct features among the spectral parameters are selected.

In 2, it is seen that three forms of clutter have more or less definite spectral characteristics.

1. Ground clutter has a impulse-like spectrum centered at zero Doppler
2. Weather clutter has a wider spread, with a possible moderate shift in center Doppler frequency
3. Clutter due to returns from birds is widely spread in frequency, with the center frequency possibly shifted noticeably away from zero Doppler

For each clutter type; velocity, spread and CNR intervals are approximately defined shown in 5.1.

Table 5.1: Statistical Parameters

Type	Velocity (m/sec)	Spread (m/sec)	CNR (dB)
Ground	0	0-1	1-30
Sea	0-15	0.7-1	1-50
Rain	0-30	1.8-4	1-50
Bird	5-30	0-1	3-50
Target	0-300	0-0.2	1-50

The mean of phase difference between signals give information about radial velocity

of target reference to radar. A multi-segment Burg algorithm for reflection coefficients is used. By taking a reference phase Doppler information is removed from input of classifier (ρ_1). As the order of spectra increases, the ability to show deviations from Gaussian increases. Combined skewness and kurtosis is considered for performance evaluation. [37] The performance of neural network architecture was examined by following listed steps.

1. Choose training pairs, architecture, learning rate, initial values for weights, activation function, batch size and cost functions
2. Shuffle training pairs
3. Run neural net on each pair in the first minibatch or each pair
4. Calculate cost results
5. Calculate gradient of cost with respect to weight
6. Update weights
7. Repeat steps 3-7 on next minibatch
8. Repeat steps 2-7 for as many epochs as required

5.3.1 Train the Network

Network connections are formed via training. In other words, internal structure will be adapted such that it will have ability to classify similar patterns. The main goal is to infer the statistical distribution of the data and its parameters. Many training methods are suggested, in literature. A back propagation algorithm is used for training, which can be used on supervised learning problems. Classification problem outputs are known, such that clutter labelling is a supervised learning problem.

The selection of training data set has considerable effects on the performance of the network. As mentioned before, it is trained to distinguish the moving object classes. Training data set contains feature parameters computed by the feature extractor. Training data set must be comprehensive. Features are used as classifier input. The desired outputs are clutter labels. The computed class label is determined

from neuron having the largest output value among others. Thus, 100% true classification results are obtained for widely different velocity and spread valued clutter parameters. Thus, to obtain accuracy limits several cases are tested with approximate clutter parameters.

The goal in training is to find weights and biases in order to minimize the quadratic cost function such that an optimization algorithm is needed. During learning Gradient Descent algorithm is used.

If it is trained to distinguish 2 moving objects, one can use 1 or 2 types in one range cell. 2 types and one type in one range cell cases are selected for tests. By exploiting the Doppler information, ground clutter can be distinguished.

A neural network with enough elements (called neurons) can fit any data with accuracy. If a feature is useful during training in some locations, detectors of the feature will be available in all locations during testing. Regression, the process of fitting models to data, is used. As a result of this, different neural networks, trained with the same input data set, can give varied outputs on the same problem. To ensure that a neural network of decent accuracy has been designed, it must be retrained for several times.

5.3.2 Test Neural Network

The performances are measured in terms of classification rate of each class with Average Classification Rate (ACR) in percent.

5.3.2.1 Some Possible Problems

Vanishing Gradient Problem When more layers using certain activation functions are added to the neural network, the gradient of loss function may become zero. This is an instability problem of neural network models using Gradient based optimization techniques. If it occurs, network will become challenging to train. For example, a large change in input of sigmoid activation function will result in small change at output. Thus, the derivative of corresponding case will be small.

If vanishing gradient problem occur ReLU can be used as an activation function. ReLU does not cause small derivative. If neuron is active, the gradient is positive and not tending to zero at all. If neuron is not active, it will basically die. If neuron death is not desired leaky ReLU can be used.

Exploiting Gradients If gradients become larger, back propagation will be unstable. If gradient become enormous at some point, even if later layers can learn properly, overall training will be ineffective because of the corrupted learning in earlier layers. Earlier layers carry the initial abstraction which affects the behavior of following ones. Gradient Scaling or Gradient clipping can be used to remove exploiting gradient problem.

5.3.2.2 Hyper-parameter Selection

Hyper-parameters are all the training variables set manually with a pre-determined value before starting the training.

Effect of Training Data Size Neural networks can be trained with large amounts of data. The amount of data must be chosen depending on complexity of problem and selected algorithm. There is no one size fits all answer. The training data size is found for two moderately distinct clutters (ground, bird) classification with a sigmoid activation function for a two hidden layered neural network. After several simulations, training size is selected as 21000, while test and validation data sizes are 6000 data sets.

Effect of Number of Hidden Layers and Nodes Number of layers must be chosen wisely. If it is too high some problems such as over-fitting and vanishing and exploding gradient may occur. On the other hand, lower number of layers may cause high bias and low potential model. The value generally depends on the size of training data set. Number of hidden units per layer must be chosen, adequate to find a sweet spot between high bias and variance. First hidden layer must have more nodes than input layer. By considering training data size and problem complexity, number of nodes

in hidden layers are selected as 20 and 10. Resulting training accuracy is shown in Figure 5.7, which satisfies expectations.

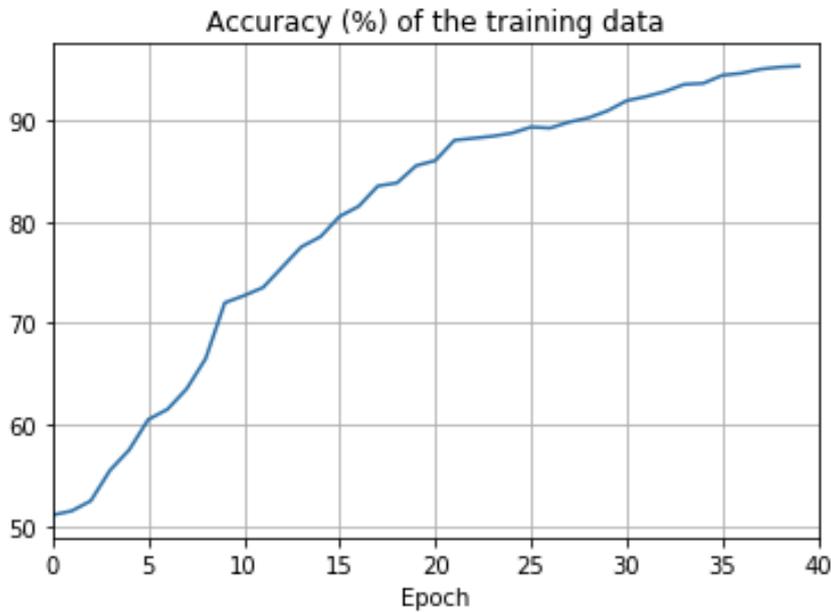


Figure 5.7: Training accuracy

Effect of Activation function The most common ones are ReLU, LeakyReLU, Sigmoid and Tanh (only for shallow networks). Sigmoid is preferable because of simplicity. If vanishing problem occur, ReLU can be implemented.

Effect of Learning Rate Learning rate is used for updating the weights and biases. It is responsible for the core learning characteristic. As learning rate increases, training time will decrease. As it can be interpreted from its name, it determines how fast a network updates its own beliefs with new ones. During training, at each of iterations the derivative of loss function is calculated with respect to each weight and bias, and subtracted from that weight and bias. If this process repeated too much, weights become over-correct such that the loss will actually increase or diverge. Thus, selection of learning rate is substantial. Bigger learning rates can prevent convergence to the minima; similarly algorithm cannot converge to a minimum point in case of too small steps. Trials with powers of 10 is recommended, specifically 0.001, 0.01, 0.1, 1. Comparison of different learning rate is shown in Figure 5.8.

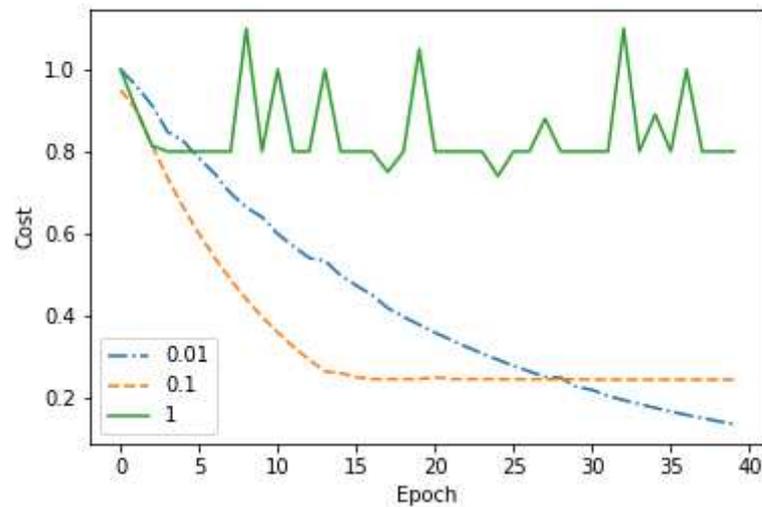


Figure 5.8: Learning rate

After multiple trials, for test scenario learning rate is selected as 0.005.

Effect of Mini batch size A Mini-batch is a group of training dataset samples. Neural network updates its parameters after computing loss with a subset of the training set. It is applied to prevent local minimum problem. If batch size is too small, learning will be slow and loss may oscillate. The gradient descent will not be smooth. If it is too high, one training iteration time will be long with relative small returns. It is generally taken as 8, 16, 32 or 64. It can also be chosen manually. If training iterations take too much time, the value is increased. If it oscillates too much, it is decreased. Once batch size is chosen, it become locked, such that no need to change with the variation of other hyper parameters. Thus, Mini Batched Gradient Descent is the algorithm, used to update weights of every layer at each of iterations.

Number of epochs Number of epochs represents how many times entire training data will be examined by training algorithm. It is important during model fitting to train data. One epoch can contain more than one mini-batch. High number of epochs may result in over fitting such that probability of generalization problems of the test and validation set increases. Lower number of epochs may restrict the model

potential. A variety of values must be tried to find the optimal one.

Effect of Weight Initialization Weight Initialization is the selection of weights of perceptrons for initial iteration. Zero or constant weight initializations are not popular. It is wise to use a Gaussian distribution to initialize weights and biases. A normal distribution is selected, having zero mean and 1 as variance.

Effect of Regularization In regularization, an extra term is added to the cost function. Regularization parameter λ denotes the degree of regularization. If it is zero, no regularization is performed. On the other hand, large values of λ correspond to high regularization. Regularization only adjust the weights at each layer. Generally, biases are untouched. It is selected as 1 when necessary.

Effect of Distinction of Classes As the distinction between classes of interest increases, classification accuracy will improve. Both training and test accuracy of two scenarios are shown below.

Table 5.2: Train and Test Accuracies for Two Scenarios

	Train Accuracy (%)	Test Accuracy (%)
Bird - Groud	100	100
Sea - Rain	94.1	93.8

In which, the accuracy of sea and rain classification is lower because their Power Spectrum Densities are close to each other. In other words, while bird and ground clutters form relatively more distinct spectrum, sea and rain will have coinciding parameter values, datasets.

5.3.2.3 Test Scenarios

Clutter types in train data have equal probability. As weather condition get harder probability of target detection decreases and weather clutter classification rate increases, as expected.

Three Moving Object In these scenarios, three classes are defined; sea, rain and target. The designed neural network architecture is tested such that it will classify echoes reflected from three moving objects. Training data is generated using Table 5.1. The training accuracy is 93.6%. Data sets are generated such that each class will have a similar amount of data. Test accuracy is measured in order to justify the performance of designed neural network.

Calm Day In a calm day, sea and rain velocities will be moderately lower. Target power will be dominant. 6000 data sets are produced by using 5.3.

Table 5.3: Calm Day Scenario

Type	Velocity (m/sec)	Spread (m/sec)	CNR (dB)	Test Accuracy
Sea	0-5	0.7	10-20	87.2 %
Rain	0-10	1.8-2.5	10-20	
Target	0-300	0.001	40-50	

A confusion matrix can be used to illustrate the results of classification. Unlike accuracy which simply puts everything into single number. The confusion matrix is of n x n dimension, where n is the number of classes. Confusion matrix entries are given in percentage with respect to test data size. The calm day scenario Confusion matrix is given as,

Table 5.4: Calm Day Test Results

		Predicted		
		Sea	Rain	Target
Actual	Sea	32.8%	2.1%	0.1%
	Rain	10.6%	24.4%	0.0%
	Target	0.0%	0.0%	30.0%

In the confusion matrix, it is possible to see where the miss-classifications occur. Ideally, a confusion matrix shows a diagonal from the top-left corner to the bottom-right corner.

Moderate day, light wind In a regular day, sea and rain velocities will be moderate. Target power will be similar to ones of clutters. 6000 data sets are produced by using 5.5.

Table 5.5: Moderate Day Scenario

Type	Velocity (m/sec)	Spread (m/sec)	CNR (dB)	Accuracy
Sea	5-10	0.85	20-35	93.4%
Rain	10-20	2-3	20-35	
Target	0-300	0.001	25-35	

Confusion matrix let to distinguish between different types of success and errors made by the classifier, which is given as,

Table 5.6: Moderate Day Test Results

		Predicted		
		Sea	Rain	Target
Actual	Sea	31.8%	3.2%	0.0%
	Rain	3.4%	31.6%	0.0%
	Target	0.0%	0.0%	30.0%

Rough/Stormy day In harsh weather conditions, sea and rain velocities will be high. Target power will be lower than ones of clutters. Test data sets are produced by using 5.7 randomly.

Table 5.7: Rough Day Scenario

Type	Velocity (m/sec)	Spread (m/sec)	CNR (dB)	Accuracy
Sea	10-15	0.95	35-50	95.2%
Rain	20-30	3-4	35-50	
Target	0-300	0.001	10-25	

In order to better identify the classes which is not generalized well by the model, a confusion matrix is implemented based on the results of test dataset. The detailed Confusion matrix is given as,

Table 5.8: Rough Day Test Results

		Predicted		
		Sea	Rain	Target
Actual	Sea	30.3%	4.5%	0.2%
	Rain	0.1%	34.9%	0.0%
	Target	0.0%	0.0%	30.0%

In all scenarios, target classification was successfully performed because of its distinct velocity and spread values. As powers of clutters increases, the classification performance improves.

Two Objects Two clutter classification is aimed. The training data is generated for most general scenario. First of all, two classes is classified by assuming 1 clutter exist in each data set, specified at a range bin.

Table 5.9: One Clutter in Each Dataset

Type	Velocity (m/sec)	Spread (m/sec)	CNR (dB)	Train Accuracy	Test Accuracy
Sea	0-15	0.7-1	1-50	94.1%	93.8%
Rain	0-30	1.8-4	1-50		

The detailed Confusion matrix is also given as,

Table 5.10: One Clutter Test Results

		Predicted	
		Sea	Rain
Actual	Sea	47.6%	2.4%
	Rain	3.8%	46.2%

Secondly, three classes are classified. First and second ones are two clutters, and third class is the case when both of clutters exist in a range cell. The probabilities of three

classes are equal. The power spectrum parameter intervals and accuracy results are shown in 5.11.

Table 5.11: One or Two Clutters in Each Dataset

Type	Velocity (m/sec)	Spread (m/sec)	CNR (dB)	Train Accuracy	Test Accuracy
Sea	0-15	0.7-1	1-50	78.7%	76.8%
Rain	0-30	1.8-4	1-50		

The detailed Confusion matrix is also given as,

Table 5.12: Classification Test Results

		Predicted		
		Sea	Rain	Sea & Rain
Actual	Sea	29.7%	1.8%	0.6%
	Rain	1.1%	26.6%	5.4%
	Sea & Rain	3.8%	10.5%	20.5%

As it can be seen, as closeness between classes increases, the classification performance worsens. It is observed that increasing the powers of clutters makes slight improvement on the performance. Furthermore, if power difference increases between clutters, the classification gets worse.

Classification Tests after Turbo Parameter Estimation Methods In Chapter 3, suggested novel parameter estimation method is explained in detail. From received radar echo, first of all three moments of clutter components in power spectrum are estimated, $\hat{\mu}_{NN}$ by three loop Turbo method initialized with frequencies estimated by Burg algorithm. Estimated CNRs are also taken from Burg algorithm. After Turbo estimation, approximate moments are used to calculate features, inputs of neural network. Test data is generated by adding 10% error over estimated values such that random parameter choice is performed over interval, $[0.9 \times \hat{\mu}_{NN} \quad 1.1 \times \hat{\mu}_{NN}]$. The test parameters will be selected randomly over an interval, having length 20% of parameter values. Train data will be generated by parameters selected in in-

terval, shown by 5.11. The radar parameters are same as in 3.4. Only values of three moments of clutters in spectrum will change. $\hat{\mu}_{NN}$ represents values $\hat{\mu}_{NN} = [v_{rain} \ \sigma_{rain} \ CNR_{rain} \ v_{sea} \ \sigma_{sea} \ CNR_{sea}]$.

Distant Clutters The estimation is performed over data containing two clutters with velocities 3.872 and 26.178 m/sec, CNR's 50 dB and spreads 1.512 and 3.108 m/sec. After 100 Monte Carlo trials, mean estimated velocity values are 3.908 and 26.309 m/sec. Mean estimated spread values are 1.545 and 3.028 m/sec and CNR's are 49.2 dB. Thus, $\hat{\mu}_{NN}$ is equal to [26.309 3.028 49.2 3.908 1.545 49.2]

Table 5.13: Distant Clutters Spectrum Classification

Type	Velocity (m/sec)	Spread (m/sec)	CNR (dB)	Test Accuracy
Sea	3.517-4.3	1.391-1.7	44.28-54.12	86.7%
Rain	23.678-28.94	2.725-3.33	44.28-54.12	

The detailed Confusion matrix is also given as,

Table 5.14: Distant Clutters Test Results

		Predicted		
		Sea	Rain	Sea & Rain
Actual	Sea	21.7%	9.2%	2.5%
	Rain	0.1%	33.3%	0.6%
	Sea & Rain	0.1%	0.8%	31.7%

Closely Spaced Clutters The estimation is performed over data containing two clutters having same spread and CNR values with distant case. However, velocities are closer such that 3.872 and 12.345 (m/sec). After 100 Monte Carlo trials, mean $\hat{\mu}_{NN}$ is equal to [12.294 3.142 49.32 3.423 1.624 49.32].

Table 5.15: Closely Spaced Clutters Spectrum Classification

Type	Velocity (m/sec)	Spread (m/sec)	CNR (dB)	Test Accuracy
Sea	3.081-3.765	1.46-1.786	44.39-54.25	70.1%
Rain	11.065-13.523	2.83-3.46	44.39-54.25	

The detailed Confusion matrix is also given as,

Table 5.16: Closely Spaced Clutters Test Results

		Predicted		
		Sea	Rain	Sea & Rain
Actual	Sea	21.2%	10.3%	0.8%
	Rain	0.8%	20.0%	12.9%
	Sea & Rain	0.4%	3.7%	29.9%

CHAPTER 6

CONCLUSIONS

The main goal of this thesis is the optimization of radar detection performance in difficult operating environments. Additionally, real time operation is a requirement for radar systems. Thus, instantaneous characterization of the received echo is aimed by adaptive estimation and detection. Unwanted signals in received echo must be eliminated to optimize radar detection performance.

In order to diminish clutter components, first of all they must be identified accurately so that clutter characteristic determination is set as primary objective. Thus, clutter power spectrum estimation methods are put under the scope. Both non-parametric and parametric estimation methods are searched in detail. Since high performance is necessary, parametric methods are preferred, requiring a parametric Power Spectrum Density model. With Gaussian assumption, zeroth, first and second moments of spectrum are selected as estimated parameters in order to obtain full clutter characterization. As a primary step first and zeroth power estimation is aimed by implementing suggested methods in literature; MUSIC, ESPRIT and Burg. Burg method gives comparatively accurate estimates. Thus, first and zeroth moment can be estimated successfully. Yet, second moment, indicator of Doppler spread, cannot be estimated with implemented algorithms. In order to estimate all three moments of spectrum, Stochastic Maximum Likelihood (SML) method is implemented, which is suggested in the paper of Boyer [22]. The method is computationally heavy and not adequately robust to the value of initial point. Thus, an original algorithm is designed which uses Burg method to select initial parameter values and use SML to guess all moments with a less complex method. The computational load is lightened, and nearly optimal accuracy is obtained. The proposed algorithm performance is also verified in different Pulse Repetition Frequency regions.

Conventional detection methods used in literature are not appropriate for all scenarios. For example, radars may be moving or stationary, and working environment may be varying. In order to advance detection in such scenarios, after obtaining characterization of clutter components, the effect of them on detection must be diminished with an optimal filtering. In order to obtain maximum information about target, estimated signal model is used to design a parametric adaptive filter. While Sample Matrix Inversion based adaptive filters suggested in literature requires large number of secondary data, this filter can work with small number of snapshots. Thus, detection performance is improved significantly. By making use of novel estimation method, the estimated clutter parameters are nearly accurate. Thus, the detection filter performances reached nearly to the genie-aided detector case. Genie-aided detector performance is used as a benchmark since its detection filter is found by true parameter values. The proposed filter can eliminate clutter signals in order to obtain target information accurately with necessitating small number of secondary cells even in non-stationary environments.

Finally, the classification of clutter signals is performed. Since classification is a Pattern Recognition problem, one of Artificial Intelligence methods must be utilized. A feed-forward Neural Network structure is the best architecture for classification problems with its superior performance in real time operation. Thus, a problem specific feed-forward Neural Network is designed, aiming to obtain types of clutter echoes in received radar signal. The feature set giving the most information about received signal is selected as in [37], in which Prediction Error Filter reflection coefficients, second and higher order statistics of received signal are used as inputs of Neural Network. The feature set is the most comprehensive one suggested in literature, yet the proposed architecture was not sufficient when thinking the current improvements on Neural Networks. Thus, a novel architecture is designed and tested. After making sure the potency of classifier performance, it is integrated with the proposed estimation algorithm. The Neural Network architecture, combined with a parameter estimator, is a novel contribution to the literature. After evaluating the classifier performance, it is concluded that combination of estimator improves the accuracy of classifier.

6.1 Future Works

According to the conclusions drawn from experimental results, possible improvements in methods can be listed as:

- Investigating parameter estimation of other clutter types such as ground, bird etc.,
- Proposed algorithm can be tested when clutter texture shows some fluctuations,
- Non-Gaussian clutter models in literature can be tested with proposed algorithm, Turbo method,
- The feature set can be chosen such that a radar data image will be produced, which can be classified by Convolutional Neural Networks (CNN),
- An adaptive filter can be designed, which uses estimated parameters directly,
- Some pre-defined optimal filters can be defined, in which the selection will be done with the classification results of Neural Network.

REFERENCES

- [1] K. D. Ward, S. Watts, and R. J. Tough, *Sea clutter: scattering, the K distribution and radar performance*, vol. 20. IET, 2006.
- [2] M. Richards, *Fundamentals of Radar Signal Processing*, vol. 1st Ed. McGraw-Hill, 2005.
- [3] D. S. Zrnic, “Estimation of spectral moments for weather echoes,” *IEEE Transactions on Geoscience Electronics*, vol. 17, no. 4, pp. 113–128, 1979.
- [4] F. Wong and I. G. Cumming, “A combined SAR Doppler centroid estimation scheme based upon signal phase,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 34, no. 3, pp. 696–707, 1996.
- [5] J. Capon, “High-resolution frequency-wavenumber spectrum analysis,” *Proceedings of the IEEE*, vol. 57, no. 8, pp. 1408–1418, 1969.
- [6] K. Miller and M. Rochwarger, “A covariance approach to spectral moment estimation,” *IEEE Transactions on Information Theory*, vol. 18, no. 5, pp. 588–596, 1972.
- [7] J. M. Dias and J. M. Leitão, “Nonparametric estimation of mean Doppler and spectral width,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 38, no. 1, pp. 271–282, 2000.
- [8] S. S. Abeysekera, “Performance of pulse-pair method of Doppler estimation,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 34, no. 2, pp. 520–531, 1998.
- [9] R. Frehlich and M. Yadlowsky, “Performance of mean-frequency estimators for Doppler radar and lidar,” *Journal of atmospheric and oceanic technology*, vol. 11, no. 5, pp. 1217–1230, 1994.
- [10] R. Schmidt, “Multiple emitter location and signal parameter estimation,” *IEEE Transactions on Antennas and Propagation*, vol. 34, no. 3, pp. 276–280, 1986.

- [11] B. Friedlander, "A sensitivity analysis of the music algorithm," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, no. 10, pp. 1740–1751, 1990.
- [12] O. A. Oumar, M. F. Siyau, and T. P. Sattar, "Comparison between music and esprit direction of arrival estimation algorithms for wireless communication systems," in *The First International Conference on Future Generation Communication Technologies*, pp. 99–103, IEEE, 2012.
- [13] R. Roy and T. Kailath, "Esprit-estimation of signal parameters via rotational invariance techniques," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 7, pp. 984–995, 1989.
- [14] J. P. Burg, "Maximum entropy spectral analysis," in *37 Annual International Meeting, Soc. of Explor. Geophys., Oklahoma City, Okla., Oct. 31, 1967*, 1967.
- [15] Chih-Kuang Yeh and Pai-Chi Li, "Doppler angle estimation using the ar spectrum model," in *2000 IEEE Ultrasonics Symposium. Proceedings. An International Symposium (Cat. No.00CH37121)*, vol. 2, pp. 1513–1516 vol.2, Oct 2000.
- [16] R. O. Schmidt, "A signal subspace approach to multiple emitter location and spectral estimation.," 1982.
- [17] M. Wax, *Detection and estimation of superimposed signals*. University Microfilms, 1995.
- [18] H. A. Cirpan and M. K. Tsatsanis, "Maximum likelihood blind channel estimation in the presence of Doppler shifts," *IEEE Transactions on Signal Processing*, vol. 47, no. 6, pp. 1559–1569, 1999.
- [19] J. Dias and J. Leita, "Maximum likelihood estimation of spectral moments at low signal to noise ratios," in *1993 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, pp. 149–152, IEEE, 1993.
- [20] M. Levin, "Power spectrum parameter estimation," *IEEE Transactions on Information Theory*, vol. 11, pp. 100–107, January 1965.
- [21] P. Waldteufel, "An analysis of weather spectra variance in a tornadic storm," 1976.

- [22] E. Boyer, P. Larzabal, C. Adnet, and M. Petitdidier, "Parametric spectral moments estimation for wind profiling radar," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 41, no. 8, pp. 1859–1868, 2003.
- [23] J. B. Billingsley, "Exponential decay in windblown radar ground clutter doppler spectra: Multifrequency measurements and model.," tech. rep., Massachusetts Inst Of Tech Lexington Lincoln Lab, 1996.
- [24] E. J. Kelly, "An adaptive detection algorithm," tech. rep., Massachusetts Inst Of Tech Lexington Lincoln Lab, 1986.
- [25] E. J. Kelly and K. M. Forsythe, "Adaptive detection and parameter estimation for multidimensional signal models," tech. rep., Massachusetts Inst Of Tech Lexington Lincoln Lab, 1989.
- [26] D. R. Fuhrmann, E. J. Kelly, and R. Nitzberg, "A CFAR adaptive matched filter detector," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 28, no. 1, pp. 208–216, 1992.
- [27] I. S. Reed, J. D. Mallett, and L. E. Brennan, "Rapid convergence rate in adaptive arrays," *IEEE Transactions on Aerospace and Electronic Systems*, no. 6, pp. 853–863, 1974.
- [28] S. Kraut and L. L. Scharf, "The CFAR adaptive subspace detector is a scale-invariant GLRT," *IEEE Transactions on Signal Processing*, vol. 47, no. 9, pp. 2538–2541, 1999.
- [29] E. Conte, M. Lops, and G. Ricci, "Asymptotically optimum radar detection in compound-gaussian clutter," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 31, no. 2, pp. 617–625, 1995.
- [30] N. B. Pulsone and C. M. Rader, "Adaptive beamformer orthogonal rejection test," *IEEE Transactions on Signal Processing*, vol. 49, no. 3, pp. 521–529, 2001.
- [31] A. Sheikhi, M. Nayebi, and M. Aref, "Adaptive detection algorithm for radar signals in autoregressive interference," *IEE Proceedings-Radar, Sonar and Navigation*, vol. 145, no. 5, pp. 309–314, 1998.

- [32] S. Bose and A. O. Steinhardt, "A maximal invariant framework for adaptive detection with structured and unstructured covariance matrices," *IEEE Transactions on Signal Processing*, vol. 43, no. 9, pp. 2164–2175, 1995.
- [33] O. Besson and D. Orlando, "Adaptive detection in nonhomogeneous environments using the generalized eigenrelation," *IEEE Signal Processing Letters*, vol. 14, no. 10, pp. 731–734, 2007.
- [34] C. D. Richmond, "Performance of a class of adaptive detection algorithms in nonhomogeneous environments," *IEEE Transactions on Signal Processing*, vol. 48, no. 5, pp. 1248–1262, 2000.
- [35] I. Jouny, F. D. Garber, and S. C. Ahalt, "Classification of radar targets using synthetic neural networks," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 29, pp. 336–344, April 1993.
- [36] R. P. Lippmann, "Pattern classification using neural networks," *IEEE Communications Magazine*, vol. 27, pp. 47–50, Nov 1989.
- [37] S. Haykin and C. Deng, "Classification of radar clutter using neural networks," *IEEE Transactions on Neural Networks*, vol. 2, pp. 589–600, Nov 1991.
- [38] M. A. Rico-Ramirez and I. D. Cluckie, "Classification of ground clutter and anomalous propagation using dual-polarization weather radar," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 46, pp. 1892–1904, July 2008.
- [39] K. Ng and R. Lippmann, "Practical characteristics of neural network and conventional patterns classifiers," in *Advances in Neural Information Processing Systems 3, [NIPS Conference, Denver, Colorado, USA, November 26-29, 1990]*, pp. 970–976, 1990.
- [40] M. A. Darzikolaei, A. Ebrahimzade, and E. Gholami, "Classification of radar clutters with artificial neural network," in *2015 2nd International Conference on Knowledge-Based Engineering and Innovation (KBEI)*, pp. 577–581, Nov 2015.
- [41] C. Bouvier, L. Martinet, G. Favier, H. Sedano, and M. Artaud, "Radar clutter classification using autoregressive modelling, k-distribution and neural net-

- work,” in *1995 International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, pp. 1820–1823 vol.3, May 1995.
- [42] A. Farina and F. A. Studer, “A review of CFAR detection techniques in radar systems,” *Microwave Journal*, vol. 29, p. 115, 1986.
- [43] P. Stoica, R. L. Moses, *et al.*, “Spectral analysis of signals,” 2005.
- [44] S. M. Kay, *Modern spectral estimation*. Pearson Education India, 1988.
- [45] “Autoregressive model.” http://en.wikipedia.org/wiki/Autoregressive_model.
- [46] S. Haykin, B. W. Currie, and S. B. Kesler, “Maximum-entropy spectral analysis of radar clutter,” *Proceedings of the IEEE*, vol. 70, pp. 953–962, Sep. 1982.
- [47] D. de Ridder, D. M. J. Tax, B. Lei, G. Xu, M. Feng, Y. Zou, and F. van der Heijden, *Classification, Parameter Estimation and State Estimation An Engineering Approach Using MATLAB*.
- [48] G. M. Güvensen and Ç. Candan, “On the impact of fast-time and slow-time preprocessing operations on adaptive target detectors,” in *2018 IEEE Radar Conference (RadarConf18)*, pp. 1183–1188, IEEE, 2018.
- [49] F. Rosenblatt, “The perceptron: a probabilistic model for information storage and organization in the brain,” *Psychological review*, vol. 65, no. 6, p. 386, 1958.
- [50] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.
- [51] C. M. Bishop *et al.*, *Neural networks for pattern recognition*. Oxford university press, 1995.
- [52] R. E. Bellman, *Adaptive control processes*. Princeton, NJ: Princeton University Press, 1961.
- [53] L. V. Fausett *et al.*, *Fundamentals of neural networks: architectures, algorithms, and applications*, vol. 3. prentice-Hall Englewood Cliffs, 1994.
- [54] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.

- [55] I. Daubechies and E. Weinan, *Princeton Series in Applied Mathematics*. Princeton, NJ: Princeton University Press, 2014.
- [56] D. G. Luenberger and Y. Ye, *Linear and Nonlinear Programming*. Springer, 2008.
- [57] R. BALDICK, *Applied Optimization, , Formulation and Algorithms for Engineering Systems*. Cambridge University Press, New York, 2006.
- [58] W.-T. Chu, “Chapter 11 quasi-newton methods,” 2014. https://www.cs.ccu.edu.tw/~wtchu/courses/2014s_OPT/Lectures/Chapter%2011%20Quasi-Newton%20Methods.pdf.

APPENDIX A

APPENDIX

A.1 Derivatives of \mathbf{R}_x

$$\left(\frac{\partial \mathbf{R}_x}{\partial w_i}\right)_{kl} = jPRI(k-l)P_i e^{jw_i(k-l)PRI - 2\pi^2\sigma_i^2(k-l)^2PRI^2} \quad (\text{A.1.1})$$

$$\left(\frac{\partial \mathbf{R}_x}{\partial \sigma_i^2}\right)_{kl} = -2\pi^2PRI^2(k-l)^2P_i e^{jw_i(k-l)PRI - 2\pi^2\sigma_i^2(k-l)^2PRI^2} \quad (\text{A.1.2})$$

$$\frac{\partial \mathbf{R}_x}{\partial P_i} = \mathbf{A}(w_i)\mathbf{B}(\sigma_i^2)\mathbf{A}^*(w_i) \quad (\text{A.1.3})$$

$$\left(\frac{\partial^2 \mathbf{R}_x}{\partial w_i^2}\right)_{kl} = -PRI^2(k-l)^2P_i e^{jw_i(k-l)PRI - 2\pi^2\sigma_i^2(k-l)^2PRI^2} \quad (\text{A.1.4})$$

$$\left(\frac{\partial^2 \mathbf{R}_x}{\partial (\sigma_i^2)^2}\right)_{kl} = 4\pi^4PRI^4(k-l)^4P_i e^{jw_i(k-l)PRI - 2\pi^2\sigma_i^2(k-l)^2PRI^2} \quad (\text{A.1.5})$$

$$\left(\frac{\partial^2 \mathbf{R}_x}{\partial P_i \partial w_i}\right)_{kl} = jPRI(k-l)e^{jw_i(k-l)PRI - 2\pi^2\sigma_i^2(k-l)^2PRI^2} \quad (\text{A.1.6})$$

$$\left(\frac{\partial^2 \mathbf{R}_x}{\partial P_i \partial \sigma_i^2}\right)_{kl} = -2\pi^2PRI^2(k-l)^2e^{jw_i(k-l)PRI - 2\pi^2\sigma_i^2(k-l)^2PRI^2} \quad (\text{A.1.7})$$

$$\left(\frac{\partial^2 \mathbf{R}_x}{\partial w_i \partial \sigma_i^2}\right)_{kl} = -2j\pi^2PRI^3(k-l)^3P_i e^{jw_i(k-l)PRI - 2\pi^2\sigma_i^2(k-l)^2PRI^2} \quad (\text{A.1.8})$$

All other terms are zero.

A.2 Line Search and Optimization Method Selection

A.2.1 Line Search Methods

After which direction to move determined, it remains only to decide how far will be moved along that direction. Selection of the best step size, α^* , depends on the situation. General idea of search methods:

1. Start with the interval (“bracket”) $[\alpha_L, \alpha_U]$ such that α^* lies inside.
2. Evaluate objective function at two points inside the bracket.
3. Reduce the bracket.
4. Repeat the process from step two.

A.2.1.1 Uniform Search

Uniform search is the most basic search technique. It is a simultaneous search technique. In other words, functional evaluation points will be chosen beforehand. Initial interval $[\alpha_L, \alpha_U]$ will be divided smaller ones by using grid points such that $\alpha_L + k\delta$. The function will be evaluated at each of the grid points and the point giving small function value, $\hat{\lambda}$; will determine the new interval of search. The search will continue until interval length will reach the tolerance or number of iterations will exceed a predefined limit.

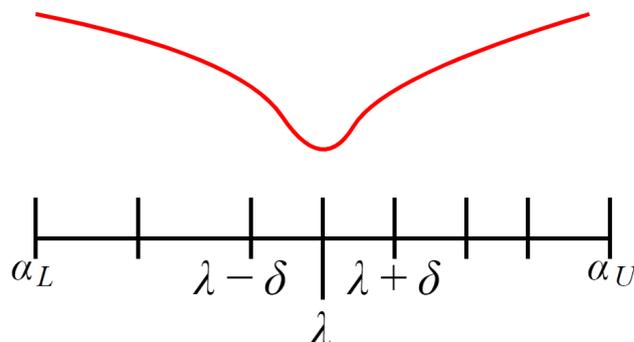


Figure A.1: Uniform Search

A.2.1.2 Dichotomous Search

Dichotomous line search technique is generally used for unimodal functions, functions having single local maximum. During dichotomous search, two points closer to the center of the interval of uncertainty are selected. Almost half of the interval of uncertainty can be eliminated by comparing relative values of the objective function at the two points. The positions of two points are where the two experiments give significantly different results. Two points are separated by a small positive value, δ .

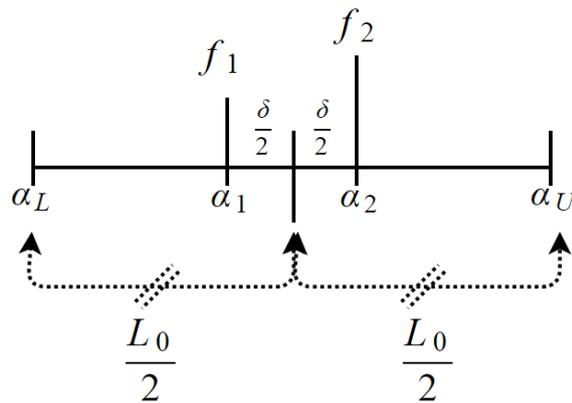


Figure A.2: Dichotomous Search

After comparing two function values, the new interval of uncertainty become $\frac{L_0}{2} + \frac{\delta}{2}$. All in all, the main idea of dichotomous search is to conduct a pair of experiments at the center of the current interval of uncertainty. After interval reduction, the next pair of experiment will be conducted with same logic. The interval of uncertainty reduction is nearly a factor of two.

A.2.1.3 Fibonacci Search

The Fibonacci's line search method uses Fibonacci numbers to achieve maximum interval reduction in a given number of steps. It is the best line search method, when the number of function evaluations is specified [55]. At each step, the size of search interval will be reduced. Fibonacci search method chooses the function evaluation points wisely. Each new point is selected such that they will be symmetrical with

respect to the point already in the remaining search interval. Fibonacci search is optimal one, since it guarantees to reduce the length of the final interval. It also uses the fewest function evaluations, among all search methods. The interval selection is shown in A.3

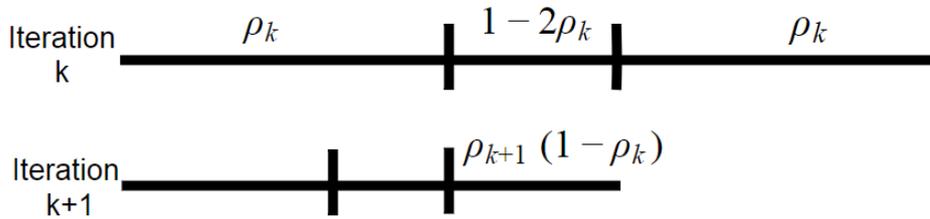


Figure A.3: Fibonacci Search

Unfortunately, before choosing the location of initial two function evaluation points, the knowledge of final interval length is necessary. In practice, a priori determination of interval may be inconvenient. Thus, golden section method is used as an alternative to this method.

A.2.1.4 Golden Section Search

The golden section method is similar to the Fibonacci method except the necessity of a priori information. In the Fibonacci method, the total number of experiments has to be specified before beginning the search, whereas this is not a requirement for the golden section method [55]. The idea is to narrow down the interval that contains the local minimum until the length of the remaining interval is less than a pre-determined tolerance level. During the Fibonacci search, the locations of first two experiments are determined from the total number of experiments. The golden section method starts with an assumption, suggesting a large number of experiments will be conducted. Of course, the total number of experiments must be decided during the computation with respect to a tolerance.

To sum up, divide an interval $[\alpha_L, \alpha_U]$ in the ratio of golden section first from right (point α_1) and then from left (point α_2). Then point α_2 divides the interval $[\alpha_L, \alpha_1]$ in the ratio of golden section and point α_1 does the same for $[\alpha_2, \alpha_U]$. The algorithm illustrated A.4

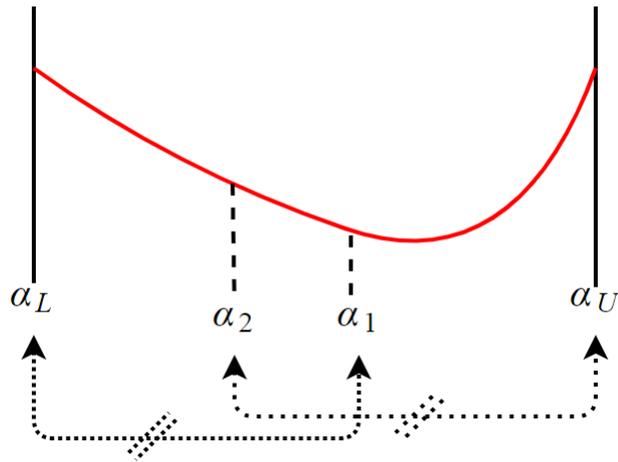


Figure A.4: Golden Search

A.2.1.5 Three Point Interval

The three point equal interval method partitions a closed and bounded interval into four closed subintervals with disjoint interiors and evaluates the function at each endpoint of the subintervals. The interval or unions of the two intervals which contain the extreme value are retained and the process is repeated using the retained closed and bounded interval. The process is terminated when the length of final interval is less than or equal to a preassigned tolerance.

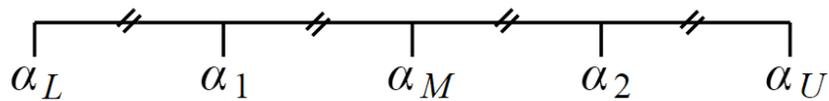


Figure A.5: Three Point Search

A.2.1.6 Quadratic Search

The key idea is to approximate objective function with a quadratic polynomial whose minimizer is known. In other words, in order to approximate the minimum of cost function on the interval $[\alpha_L, \alpha_U]$ numerically, a "quadratic interpolative" search is employed. Proceed with the method only if the subject function is a unimodal function

over the given interval. Three points selected as starting point, end point and middle of interval. After that, a quadratic interpolant is fitted to the arc contained in that interval. Then, the interpolant is minimized, and the new interval is determined based on the relation of minimizer to the original endpoints of the interval. Below thick red line stands for objective function and thin black one is the fitted quadratic function.

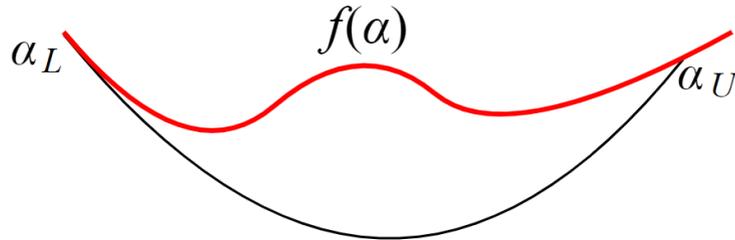


Figure A.6: Quadratic Search

A.2.1.7 Parabolic Fit

Parabolic fit search has similar idea with quadratic interpolation. Differently, Lagrange interpolation technique is used. In Lagrange Interpolation for parabolic fit, minimum value is found by one dimensional line search. The Lagrange interpolating is a tool enabling to construct a polynomial, going through any desired set of points. The problem of constructing a continuous function from specific points is called *data fitting*. The goal is to approximate a linear combination of known functions in order to fit a set of data that imposes constraints. A unique solution that fits the data exactly is tried to be guaranteed. When constructing interpolating polynomials, a trade-off occurs between a better fit and a smooth well-behaved fitting function. As the more data points included in the interpolation, degree of the resulting polynomial will become higher. Therefore, the greater oscillation will occur between the data points. Thus, high degree interpolation functions may predict the actual function between selected values poorly; although the accuracy at data points will be nearly perfect [56].

For comparison of line search methods, one clutter frequency and spread estimation is selected. As initial values of parameters; values, close to the solution, are selected. Moreover, as an optimization algorithm steepest descent with newton algorithm, ex-

plained at A.2.2.3, is chosen. For that scenario, iteration, snapshot, CNR sensitivity and spent time are shown in Table. In the scope of this work, three point interval line search method is selected as line search algorithm.

Table A.1: Line Search Methods Test Results

	Number of Iterations for Convergence 100 Snapshots , 40 dB CNR	Number of Snapshots for Convergence 40 Iterations, 40 dB CNR	CNR for Convergence 40 Iterations 40 Snapshots	Time Spent (msec) 30 Iterations 30 Snapshots 30 dB CNR
Uniform Search	27	26	8	157
Fibonacci Search	26	30	7	94
Dichotomous Search	38	41	8	47
Golden Search	22	25	6	16
Three Point Interval Search	23	19	6	46
Quadratic Search	35	41	10	62
Parabolic Fit	31	38	15	94

A.2.2 Optimization Algorithms

Each algorithm has its own advantages and disadvantages. Depending on the problem, their performance may vary. Additionally, there is a trade-off between number of iterations to satisfy stopping criterion and average effort per iteration for each algorithm [57]. Thus, choice of optimization algorithm is important.

A.2.2.1 Steepest Descent

The minimization of a general nonlinear functions is a problem, tried to be solved with many algorithms. The classical steepest descent method is the oldest and easiest method among all implemented ones. Steepest descent method is also called as *gradient descent* method. It is a first-order optimization algorithm, assuming that gradient of objective function can be computed. The nearest local minimum is found by following a line, starting from an initial point and goes towards the negative of gradient direction. Geometrically, the step direction is perpendicular to the function contour. Suppose that we would like to find the minimum of a function $L(\boldsymbol{\mu})$, $\boldsymbol{\mu} \in R_n$, and $f : R^n \rightarrow R$. We will denote the gradient of f by $\mathbf{g}_k = \mathbf{g}(x_k) = \nabla(\boldsymbol{\mu}_k)$. Then, the direction will be $\mathbf{d}_k = -\nabla(\boldsymbol{\mu}_k)$. For the steepest descent method, the formula for iterations is given by

$$\boldsymbol{\mu}_{k+1} = \boldsymbol{\mu}_k + \alpha_k \mathbf{d}_k \quad k = 0, 1, 2, \dots \quad (\text{A.2.1})$$

where the step length, α_k , is chosen so that

$$\alpha_k = \arg \min_{\alpha} L(\boldsymbol{\mu}_k + \alpha \mathbf{d}_k) \quad k = 0, 1, 2, \dots \quad (\text{A.2.2})$$

The $\alpha_k > 0$ is a selected small number. It forces the algorithm to make small jumps in order to maintain stability of algorithm. Its optimal value depends on the function. In order to find α_k ; line search method is the one chosen in previous section.

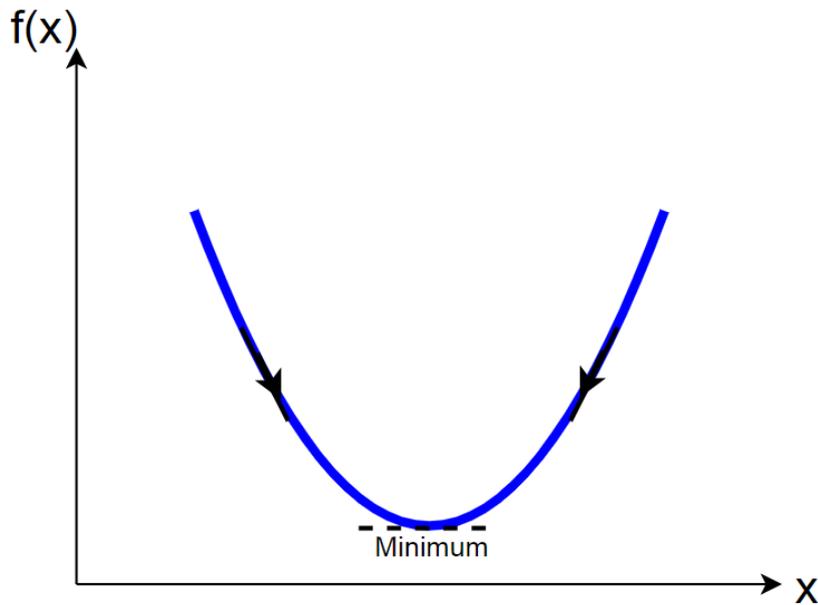


Figure A.7: Steepest Descent

As one can see, steepest descent algorithm has straight trajectory towards the minimum. Its main advantage is availability of a solution, approximate or exact. Additionally, in theory, it is guaranteed to converge to the global minimum if the loss function is convex. Similarly, if the loss function is not convex, it converges to a local minimum. The only disadvantage encountered during tests is its slowness. If the contour sets of the function are eccentric, progress towards solution is extremely slow [57]. Thus, the Newton method is implemented to obtain a fast solution.

A.2.2.2 Newton-Raphson Method

Newton's method uses first and second derivatives of the objective function. The main idea of the Newton method is to *linearize*. It indeed performs better if good initial

point selection is accomplished. However, calculating matrix of partial derivatives with related Hessian and gradient requires considerable effort. In short, Newton's method is complex to compute but fast to converge.

$$\boldsymbol{\mu}_{k+1} = \boldsymbol{\mu}_k + \alpha_k \mathbf{H}(\boldsymbol{\mu}_k)^{-1} \nabla(\boldsymbol{\mu}_k) \quad k = 0, 1, 2, \dots \quad (\text{A.2.3})$$

It is observed that Newton's Method can fail to converge the solution if initial point of iteration is not chosen wisely. Thus, Newton's method is highly related to the initial point. Newton's method for optimization in n-dimensions requires the inversion of the Hessian and therefore it can be computationally expensive for large number of variables. In addition to that, Newton's method works well if Hessian is positive everywhere. However, if Hessian is negative at some points, Newton's method may fail to converge. To eliminate the dependence on positive definiteness of Hessian matrix Levenberg-Marquart modification can be used which is shown A.2.4 .

$$\boldsymbol{\mu}_{k+1} = \boldsymbol{\mu}_k + \alpha_k \{\mathbf{H}(\boldsymbol{\mu}_k) + \epsilon \mathbf{I}\}^{-1} \nabla(\boldsymbol{\mu}_k) \quad k = 0, 1, 2, \dots \quad (\text{A.2.4})$$

However, because of its negative effect on accuracy as number of parameters increase, it did not used for solving the problem.

A.2.2.3 Newton with Steepest Descent

As it is stated, Newton method is a fast method, but it can also be unreliable because of its dependence on initial point selection. Therefore, an algorithm combining Newton Method with Steepest descent algorithm is suggested. This algorithm will start with steepest descent method. After reaching tolerance or half of maximum number of iterations, it will continue with Newton's method.

$$\boldsymbol{\mu}_{k+1} = \boldsymbol{\mu}_k + \alpha_k \mathbf{d}_k \quad k = 0, 1, 2, \dots, M \quad (\text{A.2.5})$$

$$\boldsymbol{\mu}_{k+1} = \boldsymbol{\mu}_k + \alpha_k \mathbf{H}(\boldsymbol{\mu}_k)^{-1} \nabla(\boldsymbol{\mu}_k) \quad k = M, M + 1, M + 2, \dots \quad (\text{A.2.6})$$

Convergence to the solution will be rapid, since initial guess is improved toward the solution. Additionally, computational effort is decreased.

A.2.2.4 Modified Newton Method (Rank 1 Correction)

Newton algorithm convergence is highly dependent on initial point selection and positive definiteness of Hessian. Additionally, calculation of Hessian during iteration has computational cost so instead of that Hessian is found by using Hessian of previous iteration and gradients. These methods are called as *quasi-newton methods* having form.

$$\boldsymbol{\mu}_{k+1} = \boldsymbol{\mu}_k + \alpha_k \mathbf{H}_k(\boldsymbol{\mu}_k)^{-1} \nabla(\boldsymbol{\mu}_k) \quad k = 0, 1, 2, 3, \dots \quad (\text{A.2.7})$$

Quasi Newton methods involve successive update of Hessian. A typical selection of initial Hessian matrix is generally identity matrix. The first one of the Quasi Newton Algorithms is called as Rank 1 Correction Algorithm such that

$$\mathbf{H}_{k+1} = \mathbf{H}_k + \alpha_k \mathbf{z}_k \mathbf{z}_k^T \quad (\text{A.2.8})$$

and,

$$\text{rank}(\mathbf{z}_k \mathbf{z}_k^T) = \text{rank} \left(\begin{array}{c} \left[\begin{array}{c} z_{1k} \\ \cdot \\ \cdot \\ \cdot \\ z_{nk} \end{array} \right] \left[\begin{array}{cccc} z_{1k} & \cdot & \cdot & z_{nk} \end{array} \right] \\ \end{array} \right) = 1 \quad k = 0, 1, 2, 3, \dots \quad (\text{A.2.9})$$

Thus, the name is rank one correction method is also called as Single Rank Symmetric (SRF) algorithm. The Hessian for k+1 iteration is calculated as shown below

$$\mathbf{H}_{k+1} = \mathbf{H}_k + \frac{(\Delta \boldsymbol{\mu}_k - \mathbf{H}_k(\boldsymbol{\mu}_k) \Delta(\nabla(\boldsymbol{\mu}_k))) (\Delta \boldsymbol{\mu}_k - \mathbf{H}_k(\boldsymbol{\mu}_k) \Delta(\nabla(\boldsymbol{\mu}_k)))^T}{\nabla(\boldsymbol{\mu}_k)^T (\Delta \boldsymbol{\mu}_k - \mathbf{H}_k(\boldsymbol{\mu}_k) \Delta(\nabla(\boldsymbol{\mu}_k)))} \quad (\text{A.2.10})$$

Unfortunately, the rank one correction algorithm is not the best choice for several reasons. First of all, Hessian may not be positive definite. Moreover, denominator may become close to zero. Because of these disadvantages, rank 2 correction algorithm is developed [58]

A.2.2.5 Davidon Fletcher Powell Method

It is the second Modified Newton, employing Rank 2 Correction. This algorithm was developed by Davidon (1959), Fletcher, and Powell (1963) [58]. It is also called as

the variable metric algorithm. The DFP method is one of the most powerful iterative methods, known for minimizing a general unconstrained function of n variables or parameters. Its formula is similar to the rank 1 correction method.

$$\mathbf{H}_{k+1} = \mathbf{H}_k + \frac{\Delta\boldsymbol{\mu}_k \Delta\boldsymbol{\mu}_k^T}{\Delta\boldsymbol{\mu}_k^T \Delta\nabla(\boldsymbol{\mu}_k)} + \frac{(\mathbf{H}_k(\boldsymbol{\mu}_k) \Delta\nabla(\boldsymbol{\mu}_k))(H_k(\boldsymbol{\mu}_k) \Delta\nabla(\boldsymbol{\mu}_k))^T}{\nabla(\boldsymbol{\mu}_k)^T \mathbf{H}_k(\boldsymbol{\mu}_k) \Delta\nabla(\boldsymbol{\mu}_k)} \quad (\text{A.2.11})$$

Hessian is definitely positive definite for nonzero gradient. It is superior to rank 1 correction method. However, in the case of larger non-quadratic problems, since Hessian is nearly singular, the algorithm may not converge. Thus, as a last alternative Fletcher Reeves is implemented.

A.2.2.6 Fletcher-Reeves Method

The Fletcher-Reeves (FR) method aims to solve unconstrained optimization problems. Given a function $f : R^n \rightarrow R$ the Fletcher-Reeves method tries to locate local minimum of function. It belongs to a group of methods called *conjugate gradient methods*. The FR method is the first non-linear conjugate gradient method. It has some nice properties such as the finite quadratic termination and the global convergence properties. It is a kind of line search method. Hessian is not used in the algorithm. In quadratic function case it is identical to the original conjugate direction algorithm. The algorithm steps shown below;

$$\boldsymbol{\mu}_1 = \boldsymbol{\mu}_0 - \alpha_0 \nabla(\boldsymbol{\mu}_0) \quad (\text{A.2.12})$$

$$\boldsymbol{\mu}_{k+1} = \boldsymbol{\mu}_k + \alpha_k \mathbf{d}_k \quad (\text{A.2.13})$$

in which

$$\mathbf{d}_{k+1} = -\nabla(\boldsymbol{\mu}_{k+1}) + \frac{(\nabla(\boldsymbol{\mu}_{k+1}))^T \nabla(\boldsymbol{\mu}_{k+1})}{(\nabla(\boldsymbol{\mu}_k))^T \nabla(\boldsymbol{\mu}_k)} \mathbf{d}_k \quad (\text{A.2.14})$$

Algorithm comparison is done with the same scenario in line search comparison. In other words, one clutter frequency and spread are estimated with initial values close to the solution. Three point interval line search method is used. For the case, iteration, snapshot, CNR sensitivity and spent time are shown in A.2 Fletcher Reeves method is selected as optimization algorithm while searching for optimum estimate.

Table A.2: Optimization Algorithms Test Results

	Number of Iterations for Convergence	Number of Snapshots for Convergence	CNR for Convergence	Time Spent
	100 Snapshots, 40 dB CNR	40 Iterations, 40 dB CNR	40 Iterations 40 Snapshots	30 Iterations 30 Snapshots 30 dB CNR
Steepest Descent	13	25	8	84
Newton Raphson	32	30	16	122
Newton with Steepest Descent	20	38	7	143
Rank 1 Correction	40	27	10	104
Davidon- Fletcher- Powell	20	25	15	137
Fletcher Reeves	12	20	5	38

A.3 Cramer Rao Lower Bound Calculation

The difference between covariance matrix of any unbiased estimator $\hat{\boldsymbol{\mu}}$ and CRLB must be positive definite. In other words, under certain conditions, no other unbiased estimator of the parameter $\boldsymbol{\mu}$ can have a variance smaller than CRLB. CRLB is found as the inverse of FIM (Fisher Information Matrix). FIM measure the existing total information about the parameters in observations.

The FIM is calculated as given in A.3.1. Since, SML estimator variance will reach the CRLB asymptotically; the expression for CRLB also gives theoretical SML variance.

$$[\mathbf{FIM}]_{(i,j)} = Tr\left\{\mathbf{R}_x^{-1} \frac{\partial \mathbf{R}_x}{\mu_i} \mathbf{R}_x^{-1} \frac{\partial \mathbf{R}_x}{\mu_j}\right\} \quad (\text{A.3.1})$$

Overall, CRB equation can be written as follows;

$$\frac{\mathbf{FIM}^{-1}}{K} = \mathbf{CRB}(\boldsymbol{\mu}) \leq E\{(\hat{\boldsymbol{\mu}} - \boldsymbol{\mu})(\hat{\boldsymbol{\mu}} - \boldsymbol{\mu})^T\} \quad (\text{A.3.2})$$

The performance of algorithm is investigated with respect to the CRB.