



PARTICLE FILTER BASED TRACK BEFORE DETECT ALGORITHM FOR  
TRACKING OF DIM MOVING TARGETS

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

BY

MURAT SABUNCU

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

FEBRUARY 2012

Approval of the thesis:

**PARTICLE FILTER BASED TRACK BEFORE DETECT ALGORITHM FOR  
TRACKING OF DIM MOVING TARGETS**

submitted by **MURAT SABUNCU** 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. Canan Özgen \_\_\_\_\_  
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. İsmet Erkmen \_\_\_\_\_  
Head of Department, **Electrical and Electronics Engineering**

Prof. Dr. Mübeccel Demirekler \_\_\_\_\_  
Supervisor, **Electrical and Electronics Engineering Department, METU**

**Examining Committee Members:**

Prof. Dr. Mustafa Kuzuoğlu \_\_\_\_\_  
Electrical and Electronics Engineering Dept., METU

Prof. Dr. Mübeccel Demirekler \_\_\_\_\_  
Electrical and Electronics Engineering Dept., METU

Prof. Dr. Seyit Sencer Koç \_\_\_\_\_  
Electrical and Electronics Engineering Dept., METU

Assoc. Prof. Dr. Afşar Saranlı \_\_\_\_\_  
Electrical and Electronics Engineering Dept., METU

Dr. Ülkü Çilek Doyuran \_\_\_\_\_  
Senior Design Leader, ASELSAN

**Date:** \_\_\_\_\_

**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, Last Name: MURAT SABUNCU

Signature :

# ABSTRACT

## PARTICLE FILTER BASED TRACK BEFORE DETECT ALGORITHM FOR TRACKING OF DIM MOVING TARGETS

Sabuncu, Murat

M.S., Department of Electrical and Electronics Engineering

Supervisor : Prof. Dr. Mübeccel Demirekler

FEBRUARY 2012, 89 pages

In this study Track Before Detect (TBD) approach will be analysed for tracking of dim moving targets. First, a radar setup is presented in order to introduce the radar range equation and signal models. Then, preliminary information is given about particle filters. As the main algorithm of this thesis, a multi-model particle filter method is developed in order to solve the non-linear non-Gaussian Bayesian estimation problem. Probability of target existence and RMS estimation accuracy are defined as the performance parameters of the algorithm for very low SNR targets. Simulation results are provided and performance analysis is presented as a conclusion.

Keywords: track before detect, particle filter, dim target tracking

# ÖZ

## HAREKETLİ SÖNÜK HEDEFLERİN PARÇACIK FİLTRE TABANLI İZLE BUL ALGORİTMASI İLE TAKİP EDİLMESİ

Sabuncu, Murat

Yüksek Lisans, Elektrik-Elektronik Mühendisliği Bölümü

Tez Yöneticisi : Prof. Dr. Mübeccel Demirekler

ŞUBAT 2012, 89 sayfa

Bu çalışmada, hareketli sönük hedeflerin izle bul algoritması ile takip edilmesi incelenecektir. İlk olarak radar menzil denklemi ve sinyal modellerini anlatmak üzere bir radar ortamı tanımlanmıştır. Daha sonra parçacık filtreler ile ilgili gerekli ön bilgiler verilmiştir. Bu tezin ana algoritması olarak, Gaussian dağılıma sahip olmayan ve doğrusal olmayan bir Bayesian kestirim problemini çözmek üzere çoklu model parçacık filtresi geliştirilmiştir. Çok düşük SNR değerleri için, hedef varlık olasılığı ve RMS kestirim doğruluğu performans parametresi olarak belirlenmiştir. Sonuç olarak simülasyon sonuçları verilmiş ve performans analizi gerçekleştirilmiştir.

Anahtar Kelimeler: izle bul, parçacık filtresi, sönük hedef izleme

*To the Memoirs of my Aunt and to My Lovely Family*

## **ACKNOWLEDGMENTS**

First of all, I would like to specify my grateful feeling of my supervisor Prof. Dr. Mübeccel Demirekler for her guidance and assistance throughout all the thesis preparation period.

I would like to thank Prof. Dr. Seyit Sencer Koç for his help to modelling and data generation used within this thesis.

I would like thank to my family for their great support and confidence that they have provided to me through all my life.

Finally, I would like to indicate my appreciation to ASELSAN Inc. for her supports during the preparation of this thesis.

# TABLE OF CONTENTS

ABSTRACT . . . . .	iv
ÖZ . . . . .	v
ACKNOWLEDGMENTS . . . . .	vii
TABLE OF CONTENTS . . . . .	viii
LIST OF TABLES . . . . .	xi
LIST OF FIGURES . . . . .	xii
CHAPTERS	
1 INTRODUCTION . . . . .	1
1.1 Radar History . . . . .	1
1.2 Track Before Detect . . . . .	2
1.3 Thesis Motivation and Objective . . . . .	4
1.4 Thesis Outline . . . . .	5
2 RADAR AND TRACK BEFORE DETECT CONCEPTS . . . . .	6
2.1 Radar Background . . . . .	6
2.1.1 Radar Fundamentals . . . . .	6
2.1.2 Radar Range Equation . . . . .	7
2.1.3 Signal and Noise Models used in Radar . . . . .	9
2.1.4 Radar Receiver . . . . .	11
2.1.4.1 The Matched Filter . . . . .	11
2.1.4.2 Doppler Processing . . . . .	12
2.2 Detection . . . . .	12
2.3 Particle Filters . . . . .	13
2.3.1 Sequential Importance Sampling . . . . .	14
2.3.2 Degeneracy and Resampling . . . . .	18

	2.3.3	Sequential Importance Resampling . . . . .	19
	2.3.4	Improving Sample Diversity . . . . .	21
	2.4	Tracking with Particle Filters . . . . .	22
3		DETECTION AND TRACKING OF DIM TARGETS USING TBD APPROACH	27
	3.1	Problem Statement . . . . .	27
	3.2	Target and Sensor Model . . . . .	27
	3.2.1	The Mode Variable . . . . .	29
	3.2.2	Target State Model . . . . .	30
	3.2.3	Measurement Model . . . . .	31
	3.3	Bayesian Solution of the TBD Problem . . . . .	34
	3.4	Particle Filtering . . . . .	36
	3.4.1	Initialization . . . . .	37
	3.4.2	Time Update . . . . .	37
	3.4.2.1	Existing Particles . . . . .	38
	3.4.2.2	Newborn Particles . . . . .	38
	3.4.3	Measurement Update . . . . .	40
	3.4.4	Normalize . . . . .	41
	3.4.5	Output . . . . .	41
	3.4.6	Resample . . . . .	42
	3.4.7	MCMC Move Step . . . . .	43
	3.5	Detection . . . . .	43
4		RESULTS AND SIMULATIONS . . . . .	47
	4.1	Simulation Environment . . . . .	47
	4.2	Scenario and Parameters . . . . .	47
	4.2.1	Target Scenario . . . . .	48
	4.2.2	Surveillance Region . . . . .	49
	4.2.3	The Mode Variable . . . . .	50
	4.2.4	Target State Model . . . . .	51
	4.2.5	Target Measurement Model . . . . .	53
	4.2.6	Detection . . . . .	53

4.3	Simulation Results . . . . .	53
4.3.1	Incoming Target with Constant Amplitude . . . . .	53
4.3.2	Incoming Target with Fluctuating Amplitude . . . . .	65
4.3.3	Maneuvering Target with Fluctuating Amplitude . . . . .	76
4.3.4	Noise Only Detection . . . . .	84
4.3.5	Summary of the Results . . . . .	85
5	CONCLUSION . . . . .	86
5.1	Thesis Summary . . . . .	86
5.2	Future Work . . . . .	87
	REFERENCES . . . . .	88

## LIST OF TABLES

### TABLES

Table 2.1	Radar Equation Parameters . . . . .	9
Table 4.1	Surveillance Region Parameters . . . . .	50
Table 4.2	Declaration Times for Different SNR's (Swerling-0) . . . . .	64
Table 4.3	Declaration Times Observed for Different SNR's (Swerling-1) . . . . .	76
Table 4.4	Declaration Times Observed for Different SNR's (Swerling-1) . . . . .	83
Table 4.5	Declaration Times Observed . . . . .	85

## LIST OF FIGURES

### FIGURES

Figure 1.1 Illustration of Raw Data together with Constant and Adaptive Thresholds . . .	2
Figure 1.2 Sequence of a Classical Radar Tracker vs Track Before Detect Method . . .	3
Figure 2.1 Target Trajectory . . . . .	23
Figure 2.2 RMS Tracking Error . . . . .	24
Figure 3.1 Illustration of Transition Probability Matrix . . . . .	30
Figure 3.2 Sample Range Doppler Matrix . . . . .	32
Figure 3.3 Histogram of Newborn Particles . . . . .	40
Figure 4.1 Incoming Target Trajectory . . . . .	48
Figure 4.2 Maneuvering Target Scenario . . . . .	49
Figure 4.3 Initial Position of Particles . . . . .	50
Figure 4.4 Distributions of Particles in R-D Plane . . . . .	54
Figure 4.5 Probability of Target Existence and Target SNR (Swerling-0, 10dB Initial SNR) . . . . .	55
Figure 4.6 Range and Doppler Estimate (Swerling-0, 10dB Initial SNR) . . . . .	56
Figure 4.7 RMS Tracking Error (Swerling-0, 10dB Initial SNR) . . . . .	56
Figure 4.8 Effective Number of Particles (Swerling-0, 10dB Initial SNR) . . . . .	57
Figure 4.9 SNR Estimate (Swerling-0, 10dB Initial SNR) . . . . .	57
Figure 4.10 Probability of Target Existence and Target SNR (Swerling-0, 8dB Initial SNR) . . . . .	58
Figure 4.11 Range and Doppler Estimate (Swerling-0, 8dB Initial SNR) . . . . .	59
Figure 4.12 RMS Tracking Error (Swerling-0, 8dB Initial SNR) . . . . .	59

Figure 4.13 SNR Estimate (Swerling-0, 8dB Initial SNR) . . . . .	60
Figure 4.14 Effective Number of Particles (Swerling-0, 8dB Initial SNR) . . . . .	60
Figure 4.15 Probability of Target Existence and Target SNR (Swerling-0, 6dB Initial SNR) . . . . .	61
Figure 4.16 Range and Doppler Estimate (Swerling-0, 6dB Initial SNR) . . . . .	62
Figure 4.17 RMS Tracking Error (Swerling-0, 6dB Initial SNR) . . . . .	62
Figure 4.18 SNR Estimate (Swerling-0, 6dB Initial SNR) . . . . .	63
Figure 4.19 Effective Number of Particles (Swerling-0, 6dB Initial SNR) . . . . .	63
Figure 4.20 Probability of Target Existence and Target SNR (Swerling-0, 4dB Initial SNR) . . . . .	64
Figure 4.21 Demonstration of Target Plus Noise Signal . . . . .	66
Figure 4.22 Probability of Target Existence and Target SNR (Swerling-1, 10dB Initial SNR) . . . . .	67
Figure 4.23 Range and Doppler Estimate (Swerling-1, 10dB Initial SNR) . . . . .	67
Figure 4.24 RMS Tracking Error (Swerling-1, 10dB Initial SNR) . . . . .	68
Figure 4.25 SNR Estimate (Swerling-1, 10dB Initial SNR) . . . . .	68
Figure 4.26 Effective Number of Particles (Swerling-1, 10dB Initial SNR) . . . . .	69
Figure 4.27 Probability of Target Existence and Target SNR (Swerling-1, 8dB Initial SNR) . . . . .	70
Figure 4.28 Range and Doppler Estimate (Swerling-1, 8dB Initial SNR) . . . . .	70
Figure 4.29 RMS Tracking Error (Swerling-1, 8dB Initial SNR) . . . . .	71
Figure 4.30 SNR Estimate (Swerling-1, 8dB Initial SNR) . . . . .	71
Figure 4.31 Effective Number of Particles (Swerling-1, 8dB Initial SNR) . . . . .	72
Figure 4.32 Probability of Target Existence and Target SNR (Swerling-1, 6dB Initial SNR) . . . . .	72
Figure 4.33 Range and Doppler Estimate (Swerling-1, 6dB Initial SNR) . . . . .	73
Figure 4.34 RMS Tracking Error (Swerling-1, 6dB Initial SNR) . . . . .	73
Figure 4.35 SNR Estimate (Swerling-1, 6dB Initial SNR) . . . . .	74
Figure 4.36 Effective Number of Particles (Swerling-1, 6dB Initial SNR) . . . . .	74

Figure 4.37 Probability of Target Existence and Target SNR (Swerling-1, 4dB Initial SNR) . . . . .	75
Figure 4.38 Probability of Target Existence and Target SNR (Swerling-1, 10dB Initial SNR) . . . . .	77
Figure 4.39 Range and Doppler Estimate (Swerling-1, 10dB Initial SNR) . . . . .	77
Figure 4.40 RMS Tracking Error (Swerling-1, 10dB Initial SNR) . . . . .	78
Figure 4.41 SNR Estimate (Swerling-1, 10dB Initial SNR) . . . . .	78
Figure 4.42 Effective Number of Particles (Swerling-1, 10dB Initial SNR) . . . . .	79
Figure 4.43 Probability of Target Existence and Target SNR (Swerling-1, 8dB Initial SNR) . . . . .	79
Figure 4.44 Range and Doppler Estimate (Swerling-1, 8dB Initial SNR) . . . . .	80
Figure 4.45 RMS Tracking Error (Swerling-1, 8dB Initial SNR) . . . . .	80
Figure 4.46 SNR Estimate (Swerling-1, 8dB Initial SNR) . . . . .	81
Figure 4.47 Effective Number of Particles (Swerling-1, 8dB Initial SNR) . . . . .	81
Figure 4.48 Probability of Target Existence and Target SNR (Swerling-1, 6dB Initial SNR) . . . . .	82
Figure 4.49 Probability of Target Existence and Target SNR (Swerling-1, 4dB Initial SNR) . . . . .	83
Figure 4.50 Target Existence Probability vs Time . . . . .	84

# CHAPTER 1

## INTRODUCTION

### 1.1 Radar History

The history of radar started with some experiments by Heinrich Hertz in 1887. He has discovered that electromagnetic waves can be transmitted and received through different materials. He was the one who first succeeded to generate and detect what were later called 'radio waves'. In 1900, Nikola Tesla demonstrated a concept for electromagnetic detection and velocity measurements. Later, in 1904 Christian Huelsmeyer, introduced his device to detect ships even under non-visible conditions using radio signals. In 1917, in an interview Tesla stated that, by use of the standing electromagnetic waves along with reflected surface waves, it is possible to determine the relative position of the target together with its range and speed. The development of radars accelerated in late 1930s.

In 1934, British scientist Robert A. Watson Watt demonstrated his work investigating possibility of using radars against aircraft attacks. He also demonstrated a working prototype of the system and patented his device. In 1941, U.S Army first standardized medium-range radar for air defence applications and used during World War II. Development of radar systems during the war has gained a lot of speed and various types of anti-aircraft radars are developed such as mobile, long-range and etc. U.S. General George C. Marshall stated that, "Radar equipment developed by the United States and Britain was superior to the electronics devices of either Germany or Japan. Our radar instruments, for example, which tracked aircraft in flight and directed the fire of anti-aircraft guns was more accurate than any possessed by the enemy". The statement proves the critical role committed by radars with their features such as mobility, long range and high accuracy.

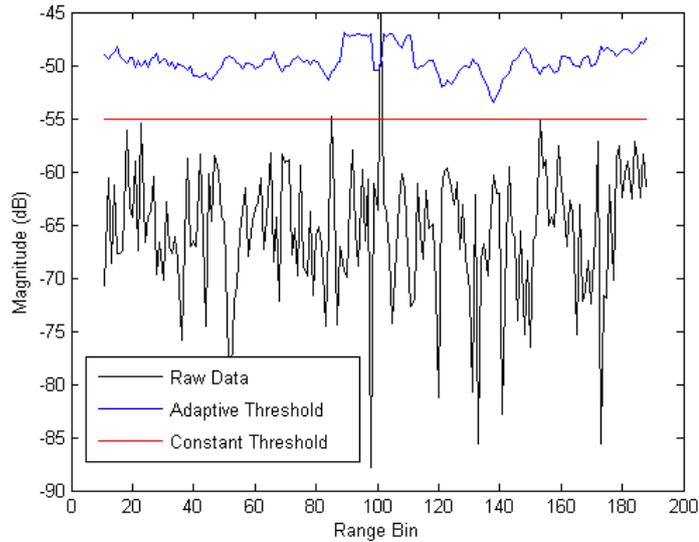


Figure 1.1: Illustration of Raw Data together with Constant and Adaptive Thresholds

## 1.2 Track Before Detect

Ever since the radar has been invented, detection of the targets from the radar data has been an area of interest. Many methods have been developed to decide if there is a target present in the surveillance region. Most of the methods are based on thresholding the radar data and making a hard decision for the target existence. Thresholds are determined considering reflected target power and receiver noise. Whereas, early methods based on constant thresholds, later methods included adaptive thresholding techniques such as CFAR (Constant False Alarm Rate). Figure 1.1 illustrates a received target echo and applied thresholds for constant and adaptive thresholding techniques. Thresholding methods are out of scope of this thesis but detailed information can be found in [18, 22].

It is obvious that determining the threshold is a difficult problem especially for low SNR (Signal-to-Noise Ratio) targets. However, stealth military aircraft's and other small targets such as missiles and UAV's increased the need for detecting low SNR, low observable targets which are so called "dim targets". The early process applied to detect those targets is to lower the thresholds which in contrast result in high false alarm environments.

In the classical approach, there are several methods proposed for dim target tracking in a high false alarm environment such as Multiple Hypothesis Tracking (MHT) and Joint Probabilistic

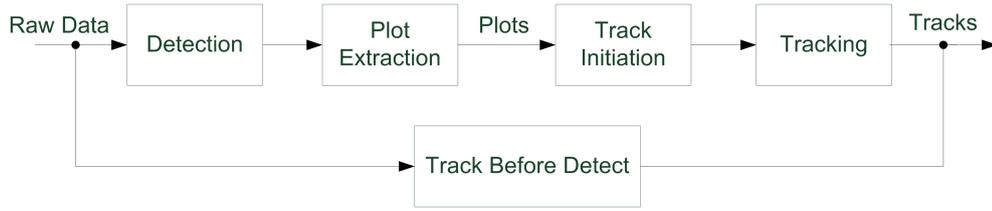


Figure 1.2: Sequence of a Classical Radar Tracker vs Track Before Detect Method

Data Association Filter (JPDAF) [3, 17, 2]. All of these methods are based on processing the output of a detector in order to initiate and update tracks, see Figure 1.2. The main idea of those algorithms is that they account for many possibilities of the target and clutter associations either in terms of plot or track level.

Track Before Detect (TBD), however is an alternative way of tracking dim targets. TBD method is in fact a combination of signal processing and tracking so that target state estimation and detection processes occur simultaneously. In the TBD method, one removes the thresholding process and TBD algorithm uses the raw data as an input, Figure 1.2. In that way, all the available information is used and more crucially it is integrated over time by the tracking filter. This typical advantage of TBD method leads to a better detection and tracking performance especially for dim targets.

There are various algorithms applicable to be used within TBD method. In [25], a Dynamic Programming (DP) based TBD method is developed as a modified Viterbi algorithm. The DP algorithm performs an exhaustive search over all possible state sequences in the state space trying to maximize a merit function. Once the merit function exceeds a given threshold, target is declared at the state sequence which ‘best’ fits to the past measurements. Both [25] and [13] includes performance analysis of DP based TBD algorithm in terms of probability of detection, probability of false alarm and tracking accuracy.

In [26], H-PMHT (Histogram based Probabilistic Multiple Hypothesis Tracking) is introduced as a multi-signal tracking algorithm based on a quantized raw data so called histogram. Each quantization cell is assumed to be associated with targets or noise in a similar way of a classical radar tracker associates plots. In [8], H-PMHT is used as a TBD method and is compared to other algorithms.

In [6], Hough Transform is used to extract detections and tracks simultaneously from multiple scans of data. The aim of Hough Transform is to locate lines in a noisy plane data. It is allowed and encouraged to use as many scans as much to increase the performance which in turn increases the amount of data to be stored. Also in [24] a maximum likelihood approach is defined to detect and track low SNR targets based on raw data.

The above algorithms generally require discretization of the state space, operate on multiple scans of data and in general require enormous computational resources. However, by Salmond and Birch [20], it is shown that a recursive Bayesian TBD algorithm can be implemented using particle filters. The aim of the PF approach is to construct a posteriori density function of the state and the likelihood ratio needed for detection at the same time. In this content, Probability of Target Existence is also calculated within the algorithm. Details of the PF based TBD algorithm structure is developed in [19] for an image based single target tracker. Boers and Driessen extends the single target assumption to two target case in [5, 4].

The potential advantages of particle filter approaches are:

- Since it is a recursive algorithm, there is no need for any data storage
- PF algorithm is capable of working with structured and/or non-Gaussian background noise
- Since target motion is modelled by a stochastic dynamic equation it is not restricted to straight-line trajectories
- Extended target and/or point spread function can be modelled explicitly
- Effects of unknown and/or fluctuating target power (SNR) can be accommodated.

Details of a PF based TBD algorithm will be derived within this thesis. Relevant references will be noted in more detail in later chapters.

### **1.3 Thesis Motivation and Objective**

Depending on the increase of the stealth technology and small missile and UAV structures, detection of low observable targets, so called dim targets become an important role of modern

radar systems. Track Before Detect is a famous way of dealing with the detection and tracking of dim targets. Apart from the classical detection techniques, TBD algorithms primarily aim to use the 'raw data' provided by the radar receiver. There exist many alternative ways of implementing TBD method in the literature. In this thesis, a Particle Filter (PF) based Track Before Detect (TBD) algorithm will be developed in order to obtain a recursive Bayesian solution to the non-linear, non-Gaussian system at hand. The aim of the filter will be to detect and track low SNR targets simultaneously and therefore to emphasize the advantages of TBD method over classical detection and tracking techniques. Finally, simulation results will be demonstrated.

## **1.4 Thesis Outline**

Chapter 1 initially introduced radar history together with detection and track before detect concepts. Later, a literature survey was given based on famous TBD algorithms. Chapter 2 will introduce preliminary information including radar equation, radar signal models, particle filtering and tracking concepts. Chapter 3 will then concentrate on the main problem that this work aims to overcome and will include the proposed algorithm. Chapter 4 will propose the simulation results and demonstrate performance parameters. Finally, Chapter 5 will summarize the work and suggest some future works.

## CHAPTER 2

### RADAR AND TRACK BEFORE DETECT CONCEPTS

#### 2.1 Radar Background

##### 2.1.1 Radar Fundamentals

The word RADAR is an acronym for 'Radio Detection and Ranging'. As it can be emphasized from the acronym, the main purpose of radar is to decide if a target exists at the line of sight, which is called '*Detection*'. Detection is based on transmitting electromagnetic waves and receiving its reflections from targets and/or undesired objects which are called '*Clutter*'. Another part of the received signal is '*Noise*' which can be induced from the receiver itself and/or external sources.

Once the target is detected, radar finds the following properties of the target: Range (how far away is the target from the radar) and Doppler (radial velocity of the target with respect to radar). Range measurement is based on evaluating the time difference between the transmitted signal and the received target echo. Doppler is based on measuring the frequency difference between the transmitted and the received signal (thus Doppler is also called Doppler shift).

Electromagnetic energy in free space travels at the speed of light, which is  $c = 2.9979 \times 10^8 \text{ m/s}$ . Thus the time that the signal travels to the target at range  $R$  and returns back to the radar is  $2R/c$ . Therefore, the range of a target is defined by,

$$R = \frac{c \Delta t}{2} \quad (2.1)$$

If the radar and the scatterer (target and/or undesired object) are moving with respect to each

other, Doppler shift will occur, where the frequency of the received echo,  $F_r$  will differ from the transmitted frequency,  $F_t$ . The difference between transmitted and received frequency is called Doppler shift or Doppler frequency ( $F_D$ ) which is calculated as,

$$F_D = \frac{2v}{c} \cdot F_t = \frac{2v}{\lambda} \quad (2.2)$$

where  $\lambda$  is the wavelength of the transmitted signal,  $v$  is the radial component of the target velocity. Note that, for an approaching target received frequency is higher than the transmitted frequency which causes a positive Doppler shift, and vice versa.

### 2.1.2 Radar Range Equation

The radar range equation (also called the radar equation) relates the range of a radar to the characteristics of the transmitter, receiver, antenna, target and the environment. The radar equation can either be used to determine the maximum range of a radar for a given target or to calculate the Signal to Noise Ratio (SNR) of a target at a specific range. Derivation of a simple form of the range equation is given below.

Assume that the transmitted power from an isotropic antenna is represented with  $P_t$ . Power density at a distance  $R$  is given by,

$$\text{Power density at a distance } R = \frac{P_t}{4\pi R^2} \quad (2.3)$$

However, radars use directive antennas which concentrate the power at a specific direction. The gain of a transmitter antenna  $G_t$ , is a measure of the increased power density radiated in some direction compared to an isotropic antenna. So that, power density at a distance  $R$  with a directive antenna is now given by,

$$\text{Power density at a distance } R = \frac{P_t G_t}{4\pi R^2} \quad (2.4)$$

Once the transmitted electromagnetic waves reaches to a target, radar cross section (RCS)  $\sigma$ , of the target determines the power density reflected back to the radar. Reflected power density is then given by,

$$\text{Reflected Power Density} = \frac{P_t G_t}{4\pi R^2} \frac{\sigma}{4\pi R^2} \quad (2.5)$$

Once the reflected signal returns back to the radar, antenna captures a portion of the echo signal. The amount of the energy being captured by the antenna is called received power  $P_r$  and is related to the antenna effective area  $A_e$ .

$$P_r = \frac{P_t G_t}{4\pi R^2} \frac{\sigma}{4\pi R^2} A_e \quad (2.6)$$

Here  $A_e$  can be related to the receiver antenna gain  $G_r$  and the operating wavelength  $\lambda$  of the antenna according to  $A_e = \lambda^2 G_r / 4\pi$ . Thus the received power can be rewritten as

$$P_r = \frac{P_t G_t G_r \lambda^2 \sigma}{(4\pi)^3 R^4} \quad W \quad (2.7)$$

Equation (2.7) gives the ideal received power if an ideal radar operated in free space conditions is assumed. In order to account for system and atmospheric losses, two loss terms are added to Equation (2.7)

$$P_r = \frac{P_t G_t G_r \lambda^2 \sigma}{(4\pi)^3 R^4 L_s L_a(R)} \quad W \quad (2.8)$$

Here  $L_s$  is the combination of system losses and is assumed to be constant for a given radar design.  $L_a(R)$  is called atmospheric loss which is a function of range and is related to a constant  $\alpha$  [dB/km], called 'atmospheric loss coefficient'. Equation (2.8) is a simple form to calculate the received power from target reflections. Next item affecting the range equation is the received noise power,  $P_n$  which is defined in Equation (2.9).

$$P_n = k T_0 B_n F_n \quad (2.9)$$

Here,  $k = 1.38 \times 10^{-23}$  J/K is called Boltzmann's constant,  $T_0 = 290$  K is the standard temperature,  $B_n$  is the noise-equivalent bandwidth and  $F_n$  is the noise figure of the receiver.

Equations (2.8) can be used to calculate maximum range of a radar,  $R_{max}$  and is obtained by substituting  $P_r$  by minimum detectable signal  $S_{min}$  ( $S_{min} = P_r$ ).

Table 2.1: Radar Equation Parameters

Source	Parameter Name	Symbology
Radar Related Parameters	Transmitted Power	$P_t$
	Transmitter Antenna Gain	$G_t$
	Receiver Antenna Gain	$G_r$
	Noise Equivalent Bandwidth	$B_n$
	Noise Figure	$F_n$
	System Losses	$L_s$
	Atmospheric Loss	$L_a$
	Wavelength	$\lambda$
Target Related Parameters	Radar Cross Section	$\sigma$
	Range	$R$

$$R_{max} = \left[ \frac{P_t G_t G_r \lambda^2 \sigma}{(4\pi)^3 L_s L_a (R) S_{min}} \right]^{1/4} \quad (2.10)$$

Equations (2.8) and (2.9) are used to calculate the Signal to Noise Ratio (SNR) of the target as  $P_r/P_n$ .

$$\frac{P_r}{P_n} = S/N = \frac{P_t G_t G_r \lambda^2 \sigma}{(4\pi)^3 R^4 k T_0 B_n F_n L_s L_a (R)} \quad W \quad (2.11)$$

Table 2.1 summarizes all the parameters of Equations (2.8) and (2.9).

### 2.1.3 Signal and Noise Models used in Radar

In any radar system, received signal consist of superposition of many major components such as; target originated reflections, clutter signals, receiver induced noise and possibly jamming. Aim of radar signal processing is to extract the useful information out of this composite signal. In this work, target and noise signals will be used to model the receiver output of a radar whereas, others will be neglected.

Assume a single pulse transmitted by a radar system is given by

$$x(t) = a(t) e^{j[2\pi F_t t + \theta(t)]} \quad (2.12)$$

where  $a(t)$  is the amplitude of the pulse envelope,  $F_t$  is the carrier frequency and  $\theta(t)$  is the possible phase modulation of the transmitted pulse.  $a(t)$  is assumed to be an ideal rectangle

with amplitude of  $A$  and duration of pulse length.  $A$  is related to the transmitted power  $P_t$  described in the previous section.

Target echo is assumed to be the same signal as the transmitted pulse which is delayed in time and shifted in frequency. Time delay corresponds to range of the target whereas frequency shift is related to the relative radial velocity between the target and the radar. Receiver noise, which is also called thermal noise is an additive random signal induced by the receiver itself. So the received echo from the transmitted pulse can be written as the sum of target echo and receiver noise as follows,

$$y(t) = b(t - t_0) e^{j2\pi F_t(t-t_0)+\phi(t)} + \eta(t) \quad (2.13)$$

where  $b(t)$  is target echo amplitude,  $\phi(t)$  is echo phase modulation and  $\eta(t)$  is receiver noise.  $t_0$  is the time delay of the transmitted pulse so that, for a target located at  $R$ ,  $t_0 = 2R/c$ . Considering that  $a(t)$  is an ideal rectangle,  $b(t)$  is also an ideal rectangle with amplitude  $B$ . The value of  $B$  is proportional to  $P_r$ , which is already given in the previous section by Equation (2.8).

It is assumed that thermal noise is the dominant part of other possible noises such as sun originated noise or galactic noise. Thermal noise is assumed to be a zero-mean Gaussian process which also has a white power spectrum. In a radar receiver, after demodulation of 'I' and 'Q' channels, it is shown that both channels contain zero-mean Gaussian noises with equal power [21]. Furthermore, since I and Q channels noise processes are Gaussian and uncorrelated, they are also independent. As a result  $\eta(t)$  can be written as

$$\eta(t) = \sqrt{P_n/2} N(0, 1) + j \sqrt{P_n/2} N(0, 1) \quad (2.14)$$

Considering that the transmitted and received signals are on a carrier of  $F_t$ , received signal must be first demodulated. After demodulation received signal is given by

$$y(t) = B e^{j\phi(t)} + \eta(t) \quad (2.15)$$

$y(t)$  here is called '*Complex Envelope*' of the received signal. It models the received echo of a single transmitted pulse and will be used by the radar signal processing algorithms defined

in the following sections.

#### 2.1.4 Radar Receiver

Radar signal processing algorithms introduced in this section are simple and common examples of a classical radar receiver. Although there are many more complex algorithms in literature, the following ones are selected to derive the raw video in the output of a radar receiver.

A Matched filter is first applied to the complex envelope of the received signal. Then, a bank of received pulses are processed together to obtain the Range-Doppler matrix.

##### 2.1.4.1 The Matched Filter

The aim of the Matched Filter, which is also called the Waveform Matched Filter, is to obtain the 'similar' part of the received echo to the transmitted waveform and thus to maximize Signal-to-Noise Ratio (SNR). The matched filter as given in [18] is defined in time-domain and frequency domain as follows

$$H(\omega) = X^*(\omega) e^{-j\omega t} \quad (2.16)$$

$$h(t) = x^*(t - t_0) \quad (2.17)$$

Here  $X(\omega)$  indicates the spectrum of the waveform,  $H(\omega)$  is the frequency response of the receiver. Thus, the output of the matched filter  $Z(\omega)$  is given by

$$Z(\omega) = H(\omega) Y(\omega) \quad (2.18)$$

So that, in time domain the output of the matched filter is given by the convolution

$$z(t) = \int y(s) h(t - s) ds \quad (2.19)$$

Then using Equations (2.17) and (2.19) together one can obtain

$$z(t) = \int y(s) x^*(s + t - t_0) ds \quad (2.20)$$

Therefore, as given in Equation (2.20) the matched filter implements a correlation between the received signal and the transmitted waveform which in turn maximizes the SNR.

#### 2.1.4.2 Doppler Processing

After matched filtering Doppler processing is applied to a bank of received pulses over a period of time. Doppler processing is implemented utilizing discrete Fourier transform (DTF).

Let  $\tilde{z}_k^{rd}$  denote the complex observation made from range bin of  $r$  and Doppler bin of  $d$  at time  $k$ . The matrix including  $\tilde{z}_k^{rd}$  for  $r = 1, \dots, r_m$  and  $d = 1, \dots, d_m$  forms the so called '*Range - Doppler (R-D) Matrix*'. Each cell size of R-D matrix is determined by the range and Doppler resolutions respectively.

Doppler processing is the final basic function of a classical radar receiver. The output of Doppler processing is the so called '*raw data*' (raw video). From now on classical approaches may apply detection techniques such as CFAR or track- before-detect algorithms as this work will concentrate on in later chapters.

## 2.2 Detection

In a classical radar design, detection is applied to the raw data received. The output of the detection process is the so called 'plots' and they are used for precise angle measurement and tracking of targets. Detection process aims to 'decide' if the signal under test consists of target reflections or noise only. Those decisions are made by using statistical hypothesis testing so that for any measurement under test for the presence of a target, one of two hypotheses ( $H_0$  or  $H_1$ ) is assumed to be true

1. The measurement is the result of noise only :  $H_0$
2. The measurement is a combination of target echo and noise :  $H_1$

Making a hard decision of target existence introduces two concepts: Probability of Detection ( $P_D$ ) and Probability of False Alarm ( $P_{FA}$ ).  $P_D$  is defined as the probability that a target is declared (i.e.  $H_1$  is chosen) when a target is in fact present.  $P_{FA}$  is the probability that a target is declared again (i.e.  $H_1$  is chosen) when a target is not present.

For radar applications, a commonly used detection rule is called Neyman-Pearson criterion. According to this criterion, the goal is to maximize  $P_D$  where  $P_{FA}$  does not exceed a given threshold. The Neyman-Pearson detection rule is realized by the ‘likelihood ratio test (LRT)’ given below, [18]

$$\frac{p_z(z|H_1)}{p_z(z|H_0)} \underset{H_0}{\overset{H_1}{>}} \lambda \quad (2.21)$$

where  $p_z(z|H_1)$  is the probability density function (pdf) of measurement  $z$  given that a target is present,  $p_z(z|H_0)$  is the pdf of  $z$  given that a target is not present. In likelihood ratio test method calculation of the likelihood ratio and defining the threshold  $\lambda$  are both crucial problems. In this thesis a particle filter based method will be proposed in order to calculate the LRT in the later section’s.

### 2.3 Particle Filters

Particle Filter (PF) is a Sequential Monte Carlo (SMC) method in order to solve the Bayesian estimation problem for non-linear, non-Gaussian systems. The Bayesian approach is to construct the pdf of the state at time  $k$  using on all available information up to the time  $k$  such that

$$x_k|Z_k \sim p(x_k|Z_k) \quad (2.22)$$

where  $x_k|Z_k$  denotes the posterior state vector,  $Z_k$  denotes the available measurements up to the time  $k$ ,  $Z_k = \{z_j : j = 1, \dots, k\}$ .

For linear-Gaussian systems the required pdf (2.22) remains Gaussian for every time step and the Kalman filter propagate and update the mean and the covariance of the pdf in an optimal

way. Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF) are two most popular sub-optimal approaches to recursive Gaussian but non-linear estimation problems. EKF is based on linearization of the system about the predicted state and thus Kalman filter can be applied [12]. On the other hand UKF depends on the idea of representing a Gaussian distribution by a set of samples which are called 'Sigma Points'. Sigma points are chosen such that mean and covariance of the distribution are protected. The non-linear function is then applied to each sigma point instead of the distribution itself. Finally, Gaussian representation is reconfigured based on transformed sigma points [14].

The main idea of particle filtering however, is to represent the required probability density function (pdf) by a set of random samples. As the number of samples becomes very large they effectively provide an exact, equivalent representation of the required pdf. Thus, PF algorithm approaches to the optimal Bayesian estimator [10]. The most powerful property of PF is that, being dependent on random samples, it is able to handle any non-linear function and/or any system or measurement noise densities of any distribution.

### 2.3.1 Sequential Importance Sampling

Importance sampling is a general Monte Carlo integration method which is applied to perform non-linear filtering. Sequential Importance Sampling (SIS) is a Monte Carlo method based on the idea of importance sampling. Let us assume that the system (state) model is of the form

$$x_{k+1} = f_k(x_k, w_k) \quad (2.23)$$

where  $x_k \in R^n$  is the target state,  $f_k : R^n \times R^m \rightarrow R^n$  is a known system transition function and  $w_k \in R^m$  is a zero mean, white noise sequence with known pdf.

The measurement model is given by

$$y_k = h_k(x_k, v_k) \quad (2.24)$$

where  $y_k \in R^p$  is the measurement vector which becomes available at time  $k$ .  $h_k : R^n \times R^r \rightarrow R^p$  is the measurement function and  $v_k \in R^r$  is a zero mean white noise sequence with known

pdf. Note that, both  $v_k$  and  $w_k$  are assumed to be independent. In order to develop the SIS algorithm it is assumed that the initial pdf of the state vector  $p(x_0) = p(x_0|Z_{-1})$  is known.

Let  $X_k = \{x_j : j = 1, \dots, k\}$  denote the sequence of all target states upto the time  $k$ . The joint posterior pdf is then given by  $p(X_k|Z_k)$  and its marginal density is denoted as  $p(x_k|Z_k)$ . Let  $\{X_k^i, w_k^i\}_{i=1}^N$  denote a random measure that characterizes the probability density function  $p(X_k|Z_k)$  where  $\{X_k^i, i = 1, \dots, N\}$  are the support points and  $\{w_k^i, i = 1, \dots, N\}$  are the associated normalized weights.

Then the joint posterior pdf at time  $k$  can be approximated as follows [9]

$$p(X_k|Z_k) \approx \sum_{i=1}^N w_k^i \delta(X_k - X_k^i) \quad (2.25)$$

where  $w_k^i, i = 1, \dots, N$  are the normalized weights of the discrete weighted approximation of the true posterior such that

$$\sum_{i=1}^N w_k^i = 1 \quad (2.26)$$

Notice that in Equation (2.25) each random sample (i.e. particle)  $X_k^i$  together with  $w_k^i$  represents a random sample of the joint posterior density belonging to a time sequence  $k = 1, \dots, k$ .

Ideally, one would like to draw particles from the true posterior density. However, since the true posterior is not known exactly, particles  $\{X_k^i, w_k^i\}_{i=1}^N$  cannot be generated directly from the posterior but instead they can be generated from another density called the '*Importance (Proposal) Density*',  $q(x)$ .

It is shown that if we have a set of samples taken from  $q(x)$  as  $\{X_k^i\}_{i=1}^N$ , then the following set of the importance weights will assure that the particles  $\{X_k^i, w_k^i\}_{i=1}^N$  is distributed according to the true posterior  $p(X_k|Z_k)$ .

$$\tilde{w}_k^i \propto \frac{p(X_k^i|Z_k)}{q(X_k^i|Z_k)} \quad (2.27)$$

where  $\tilde{w}_k^i$  is called the unnormalized importance weight. Then the '*importance weights*' are defined as follows

$$w_k^i = \frac{\tilde{w}_k^i}{\sum_{i=1}^N \tilde{w}_k^i} \quad (2.28)$$

This principle is called '*Importance Sampling*' in statistics and is applied in the Bayesian framework in order to obtain random samples distributed as the true posterior [9].

In the SIS algorithm, the requirement is to construct the pdf of the current state given all the available information with two stages: Prediction and Update. Therefore, one should start the derivation by assuming that the posterior density at time  $k - 1$  is known and given by the distribution  $p(X_{k-1}|Z_{k-1})$ .

Let us apply Bayes' rule to the posterior density as

$$p(X_k|Z_k) = \frac{p(z_k|X_k, Z_{k-1}) p(X_k|Z_{k-1})}{p(z_k|Z_{k-1})} \quad (2.29)$$

$$p(X_k|Z_k) = \frac{p(z_k|X_k, Z_{k-1}) p(x_k|X_{k-1}, Z_{k-1}) p(X_{k-1}|Z_{k-1})}{p(z_k|Z_{k-1})} \quad (2.30)$$

$$p(X_k|Z_k) = \frac{p(z_k|x_k) p(x_k|x_{k-1})}{p(z_k|Z_{k-1})} p(X_{k-1}|Z_{k-1}) \quad (2.31)$$

and since  $p(z_k|Z_{k-1})$  is a normalizing denominator we can write

$$p(X_k|Z_k) \propto p(z_k|x_k) p(x_k|x_{k-1}) p(X_{k-1}|Z_{k-1}) \quad (2.32)$$

In Equation (2.32) we have obtained the posterior pdf at time  $k$  is related to the known pdf's defined as follows

- $p(X_{k-1}|Z_{k-1})$  which is the joint posterior pdf at time  $k - 1$
- $p(x_k|x_{k-1})$  which is the known one step prediction density of the target state defined by the Equation (2.23)
- $p(z_k|x_k)$  which is the known measurement pdf defined by the Equation (2.24)

Next, let us assume that the importance density  $q(\cdot)$  is defined as

$$q(X_k|Z_k) = q(x_k|X_{k-1}, Z_k) q(X_{k-1}|Z_{k-1}) \quad (2.33)$$

In Equation (2.33) we have defined the posterior pdf of the importance density is proportional to

- $q(X_{k-1}|Z_{k-1})$  which is the joint posterior pdf of the importance density at time  $k - 1$ ,
- $q(x_k|X_{k-1}, Z_k)$  which is called the 'importance density function'.

Now in order to obtain the the weight update equation, one can substitute (2.32) and (2.33) into (2.27) as

$$\tilde{w}_k^i = \frac{p(X_k^i|Z_k)}{q(X_k^i|Z_k)} \propto \frac{p(z_k|x_k^i) p(x_k^i|x_{k-1}^i) p(X_{k-1}^i|Z_{k-1})}{q(x_k^i|X_{k-1}^i, Z_k) q(X_{k-1}^i|Z_{k-1})} \quad (2.34)$$

In Equation (2.34) the last term  $p(X_{k-1}^i|Z_{k-1})/q(X_{k-1}^i|Z_{k-1})$  is nothing but the weights from the previous step. Furthermore if we assume that  $q(x_k^i|x_{k-1}^i, z_k) = q(x_k^i|X_{k-1}^i, Z_k)$  the update equation can be written as

$$\tilde{w}_k^i \propto w_{k-1}^i \cdot \frac{p(z_k|x_k^i) p(x_k^i|x_{k-1}^i)}{q(x_k^i|x_{k-1}^i, z_k)} \quad (2.35)$$

Note that with the final assumption, only  $x_{k-1}^i$  needs to be stored and the rest of the sequences  $X_{k-2}$  and  $Z_{k-1}$  are discarded.

Finally after normalization posterior density can be approximated as

$$p(x_k|Z_k) \approx \sum_{i=1}^N w_k^i \delta(x_k - x_k^i) \quad (2.36)$$

It can be shown that as  $N \rightarrow \infty$  the approximation approaches to the true posterior.

Equations (2.35) and (2.36) define a simple SIS filter. A pseudo-code of this filter is given in Algorithm 1. The algorithm defined here is the base of many other types of particle filters. Note that the selection of the importance density  $q(x_k^i|x_{k-1}^i, z_k)$  plays a crucial role on the

algorithm's performance. The best possible choice of the importance density is the true posterior distribution which is not possible for most of the time. Details of selection of importance density will be noted in the following chapter's.

---

**Algorithm 1** SIS Filter

---

**function**  $\left[ \left\{ x_k^i, w_k^i \right\}_{i=1}^N \right] = \text{SIS Filter} \left( \left\{ x_{k-1}^i, w_{k-1}^i \right\}_{i=1}^N, z_k \right)$

- FOR  $i = 1 : N$

- Draw samples from  $x_k^i \sim q(x_k^i | x_{k-1}^i, z_k)$

- Evaluate the 'Importance Weights'

$$\tilde{w}_k^i \propto w_{k-1}^i \cdot \frac{P(z_k | x_k^i) P(x_k^i | x_{k-1}^i)}{q(x_k^i | x_{k-1}^i, z_k)}$$

- END

- FOR  $i = 1 : N$

- Normalize

$$w_k^i = \frac{\tilde{w}_k^i}{\sum_{i=1}^N \tilde{w}_k^i}$$

- END

---

### 2.3.2 Degeneracy and Resampling

The performance of the PF algorithm is tightly dependent on the effective number of particles that are 'active' in approximating the posterior density (2.36). Ideally, one would like to draw samples from the posterior density of the state directly. However, this is not possible for most of the time. Therefore one aims that the importance density is 'close' to the true posterior. In [9], it is shown that the importance sampling of the form of Equation (2.33) can only increase the unconditional variance of the importance weights. That means, after a finite number of steps, depending on how close the importance density to the true posterior is, all particles but a few will have negligible normalized weights. This phenomena is called 'sample degeneracy' and it is unavoidable.

The solution of the degeneracy problem is so called resampling. The basic idea of resampling

is to eliminate the particles that have small weights and to concentrate on the particles with large weights. A measure of the sample degeneracy is the effective number of particles which is calculated in [11] and is approximated as,

$$\hat{N}_{eff} \cong \frac{1}{\sum_i (w_{k|k}^i)^2} \quad (2.37)$$

This approximation has a property of  $1 \leq \hat{N}_{eff} \leq N$ . Note that if the particles are distributed uniformly, i.e all the weights are equal to  $1/N$ , then  $\hat{N}_{eff} = N$ . On the other hand, if only one of the particles has a weight of 1 and the others are equal to 0, then  $\hat{N}_{eff} = 1$ . Note that, it is a design criteria to apply resampling technique at every time step or when only number of effective particles are below a threshold,  $\hat{N}_{eff} < N_{th}$ .

The resampling algorithm in fact involves a mapping of the random measure  $\{x_k^i, w_k^i\}$  into a random measure  $\{x_k^{*i}, 1/N\}$  with uniform weights. The new set of particles are generated by resampling with replacement  $N$  times from an approximate discrete representation of  $p(x_k|Z_k)$ , so that  $P\{x_k^{*i} = x_k^j\} = w_k^j$ . There are several resampling algorithms in literature, [9, 11]. In Algorithm 2 the resampling algorithm that we have used for our implementation is given as defined in [19].

### 2.3.3 Sequential Importance Resampling

Sequential Importance Resampling (SIR) algorithm which is also known as the Bootstrap filter [10], is a very famous PF algorithm obtained by assuming that the importance density is equal to the transitional prior such that

$$q(x_k^i|x_{k-1}^i, z_k) = p(x_k^i|x_{k-1}^i) \quad (2.38)$$

Then the weight update Equation (2.35) takes the form

$$\tilde{w}_k^i \propto w_{k-1}^i p(z_k|x_k^i) \quad (2.39)$$

Although the selection of the transitional prior as the importance density is not the best way of

---

**Algorithm 2** Resampling Algorithm

---

**function**  $\left[ \{x_k^{*j}, w_k^{*j}, i^j\}_{j=1}^N \right] = \text{Resample} \left( \{x_k^i, w_k^i\}_{i=1}^N \right)$

- Initialize Cumulative Sum (CS) of weights:  $cs^1 = w_k^1$
  - *FOR*  $i = 2 : N$ 
    - Construct CS:  $cs^i = cs^{i-1} + w_k^i$
  - *END*
  - Draw a sample to start iteration:  $u_1 \sim U[0, 1/N]$
  - *FOR*  $j = 1 : N$ 
    - Move on CS:  $u_j = u_1 + N^{-1} \cdot (j - 1)$
    - *WHILE*  $u_j > cs^i$ 
      - \*  $i = i + 1$
    - *END*
    - Assign sample:  $x_k^{*j} = x_k^i$
    - Assign weight:  $w_k^{*j} = w_k^i$
    - Assign parent:  $i^j = i$
  - *END*
- 

defining it, SIR provides a simple and easy to implement way of particle filters. Pseudo-code of the SIR filter is given in Algorithm 3 which applies resampling at every time step. Notice that since for every time step resampling is applied, normalized weights are equal to  $1/N$  at the end of each step. Therefore,  $w_{k-1}^i$  part of Equation (2.39) is omitted in the algorithm and the weights are not propagated.

SIR filter introduced here will be the base of the TBD algorithm to be proposed in the next chapter's. Therefore, the following assumptions of the SIR filter are emphasized

1.  $p(x_k|x_{k-1})$  is a known one step prediction density obtained from the Equation 2.23 and one can draw samples from that distribution. However, there is no limit on the type of

---

**Algorithm 3** SIR Filter

---

**function**  $\left[\{x_k^i\}_{i=1}^N\right] = \text{SIR}\left(\{x_{k-1}^i\}_{i=1}^N, z_k\right)$

- *FOR*  $i = 1 : N$ 
    - Draw samples from:  $x_k^i \sim p(x_k|x_{k-1}^i)$
    - Evaluate the 'Importance Weights':  $\tilde{w}_k^i = p(z_k|x_k^i)$
  - *END*
  - *FOR*  $i = 1 : N$ 
    - Normalize
- $$w_k^i = \frac{\tilde{w}_k^i}{\sum_{i=1}^N \tilde{w}_k^i}$$
- *END*
  - Resample using the Algorithm 2
- 

the function  $f_k$  or on the noise distribution  $w_k$ .

2.  $p(z_k|x_k)$  is a known measurement density obtained from the Equation 2.24 and one can evaluate the likelihood given the state variable  $x_k$ . However again there is no limit on the type of the function  $h_k$  or on the noise distribution  $v_k$ .

### 2.3.4 Improving Sample Diversity

Although resampling is suggested in order to overcome the sample degeneracy it introduces a new problem. This problem is due to the loss of diversity of the particles, i.e. after a finite number of steps almost all particles will occupy the same point in the state space thus they provide a poor representation of the true posterior. This problem is found to be severe especially when the process noise is small and/or a parameter estimation is being made by using particle filters. There are many methods introduced in literature to improve sample diversity. In [11] the problem is analysed under the name of sample depletion. Also in [9] and [1] the loss of diversity problem is investigated in correlation with the choice of importance density. According to [19, 9, 11, 1], a very famous method in order to improve

sample diversity is to move the resampled particles to a new place according to Equation (2.40) based on a kernel density. The move function is given by

$$x_k^{*i} = x_k^i + bD_k\epsilon^i \quad (2.40)$$

where  $b$  is the kernel bandwidth and  $D_k$  is the matrix square root such that  $D_k D_k^T = S_k$  and  $S_k$  is state covariance matrix. Note that Epanechnikov or Gaussian kernels can be used as the kernel density as described in [19].

If the move step (2.40) is applied to each particle without any condition then the method is called '*Regularized Particle Filter (RPF)*'. However, theoretical disadvantage of RPF is that its samples are no longer guaranteed to asymptotically approximate the true posterior [19]. Therefore another method is proposed such that the move step (2.40) is applied based on the Metropolis Hastings acceptance rejection algorithm [7]. This method is so called '*MCMC Move Step*' and its samples are now guaranteed to asymptotically approximate the true posterior.

According to MCMC move method a particle  $x_k^i$  is moved to a new location  $x_k^{*i}$  only if  $u \leq \alpha(x, y)$  where  $u \sim U[0, 1]$  and  $\alpha(x, y)$  is the probability of move from  $x$  to  $y$ . According to the Metropolis Hastings algorithm, the probability of move is defined as [7]

$$\alpha(x_k^i, x_k^{*i}) = \min \left[ \frac{q(z_k | x_k^{*i}) p(x_k^{*i} | x_{k-1}^i)}{q(z_k | x_k^i) p(x_k^i | x_{k-1}^i)}, 1 \right] \quad (2.41)$$

In practice it is expected that MCMC move step increase the performance of SIR filter especially when the process noise is small or a parameter estimation is being made. The SIR algorithm including a MCMC move step is given in the Algorithm 4.

## 2.4 Tracking with Particle Filters

In this section a tracking application based on PF algorithm will be demonstrated. In order to analyze the key parameters and features of PF's such as optimality, effective number of particles and key role of importance density, a linear-Gaussian system will be simulated. Let us assume that the system and measurement models are defined as follows

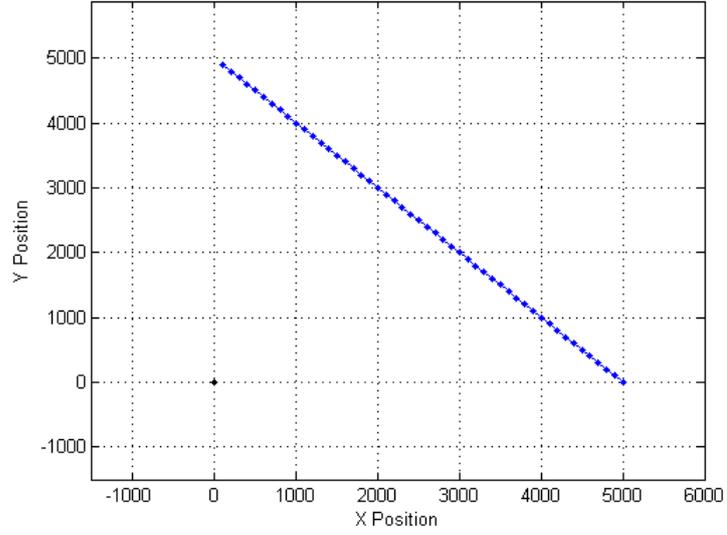


Figure 2.1: Target Trajectory

$$x_{k+1} = A x_k + w_k \quad (2.42)$$

$$z_k = C x_k + v_k \quad (2.43)$$

where  $x_k = [x \ \dot{x} \ y \ \dot{y}]^T$  is the state vector,  $A$  is the state transition function,  $w_k$  is the process noise sequence,  $z_k = [x \ y]^T$  is the measurement vector and  $C$  is the measurement transition function. Assume that,  $w_k$  and  $v_k$  are white Gaussian noise sequences with the following distributions

$$w_k \sim N(0, Q) \quad (2.44)$$

$$v_k \sim N(0, R) \quad (2.45)$$

Let us assume that  $x_0$  is the known initial state and has a Gaussian distribution i.e.  $x_0 \sim N(\bar{x}_0, \Sigma_0)$ . Furthermore  $\{x_0, w_k, v_k, k = 0, \dots, k\}$  is an independent set of random variables. Under these conditions the state vector at time  $k$  becomes conditionally Gaussian.

$$x_{k|z_{0:k}} = x_{k|k} \sim N(\bar{x}_{k|k}, \Sigma_{k|k}) \quad (2.46)$$

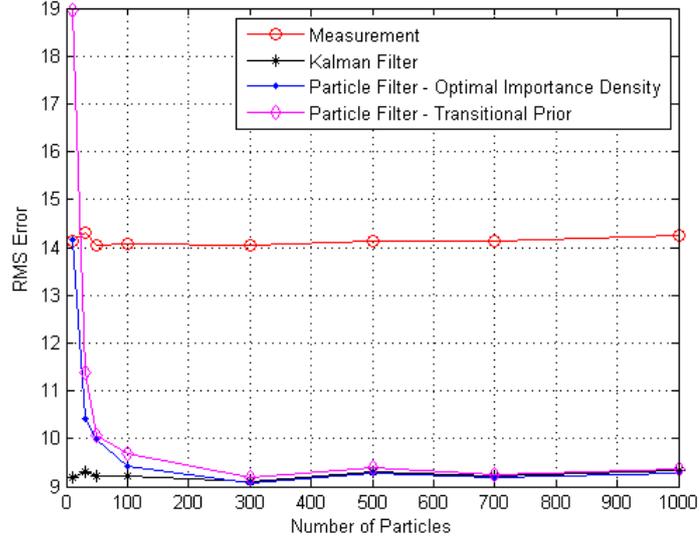


Figure 2.2: RMS Tracking Error

Kalman filter (KF) is the optimal Bayesian estimator for the linear Gaussian system defined by the Equation's (2.42) to (2.46). Our purpose here is to compare the performances of the PF algorithm KF. For this reason we have applied the two algorithms to track a target moving in X-Y plane with constant velocity as shown in Figure 2.1.

As it is noted before the transitional prior is a simple selection of importance density. However, the optimal importance density for linear Gaussian systems is also obtained as Gaussian [19] as follows

$$q(x_k^i | x_{k-1}^i, z_k) \sim N(a_k, I_k) \quad (2.47)$$

where

$$\begin{aligned} a_k &= Ax_{k-1} + I_k C^T R_k^{-1} (z_k - b_k) \\ I_k &= Q_{k-1} - Q_{k-1} C^T S_k^{-1} C Q_{k-1} \\ S_k &= C Q_{k-1} C^T + R_k \\ b_k &= CAx_{k-1} \end{aligned} \quad (2.48)$$

Note that the optimal importance density is actually the posterior density  $p(x_k | Z_k)$  obtained by the KF equations.

Therefore, let us compare the performance of two possible choices of importance density:

1. Transitional prior :  $q(x_k^i | x_{k-1}^i, z_k) = p(x_k^i | x_{k-1}^i)$
2. Optimal Gaussian :  $q(x_k^i | x_{k-1}^i, z_k) \sim N(a_k, I_k)$

Figure 2.2 shows the RMS tracking error versus number of particles for the two different types of importance density function. For the simulations, it is assumed that we have a sensor which is able to locate the target position at every 1 sec with a Gaussian error distribution of  $N(0, 100)$  meters in both  $x$  and  $y$  axes. The results are obtained through 100 Monte Carlo simulations.

---

**Algorithm 4** SIR with MCMC Move Step

---

**function**  $\left[ \{x_k^i\}_{i=1}^N \right] = \text{SIR with MCMC Move} \left( \{x_{k-1}^i\}_{i=1}^N, z_k \right)$

- *FOR*  $i = 1 : N$ 
  - Draw samples from:  $x_k^i \sim p(x_k | x_{k-1}^i)$
  - Evaluate the 'Importance Weights':  $\tilde{w}_k^i = p(z_k | x_k^i)$

• *END*

- *FOR*  $i = 1 : N$

- Normalize

$$w_k^i = \frac{\tilde{w}_k^i}{\sum_{i=1}^N \tilde{w}_k^i}$$

• *END*

- Calculate state covariance matrix

$$\bar{x}_k = \sum_{i=1}^N x_k^i w_k^i$$

$$S_k = \sum_{i=1}^N w_k^i (x_k^i - \bar{x}_k)(x_k^i - \bar{x}_k)^T$$

- Calculate  $D_k$  such that  $D_k D_k^T = S_k$
  - Resample using the Algorithm 2
  - See Algorithm 5
-

Figure 2.2 shows that, as the number of particles increases, PF algorithm approaches to the optimal Bayesian estimator, in this case KF. Moreover, if the optimal importance density can be used, even with less number of particles we can achieve the optimality. It also indicates that as the importance density become 'closer' to the true posterior one can obtain better performance in terms of RMS tracking error.

---

**Algorithm 5** SIR with MCMC Move Step (ctd.)

---

**function**  $\left[ \{x_k^i\}_{i=1}^N \right] = \text{SIR with MCMC Move} \left( \{x_{k-1}^i\}_{i=1}^N, z_k \right)$

- *FOR*  $i = 1 : N$

- Draw  $\epsilon^i$  from Epanechnikov / Gaussian kernel

- Calculate

$$x_k^{*i} = x_k^i + bD_k\epsilon^i$$

- Evaluate probability of move  $\alpha(x_k^i, x_k^{*i})$  according to Equation (2.41)

- Draw a candidate  $u$  such that  $u \sim U[0, 1]$

- \* *IF*  $u \leq \alpha(x_k^i, x_k^{*i})$

- Move to the new location such that:  $x_k^i = x_k^{*i}$

- \* *ELSE*

- Reject move such that:  $x_k^i = x_k^i$

- \* *END*

- *END*

---

## CHAPTER 3

# DETECTION AND TRACKING OF DIM TARGETS USING TBD APPROACH

### 3.1 Problem Statement

The aim of this thesis is to develop an efficient algorithm for detection and tracking of dim targets. For this purpose, a TBD based detection and tracking algorithm is derived in this chapter. Particle Filter (PF) approach will be used to implement the algorithm and the likelihood ratio test (LRT) will be used to obtain the detector.

TBD approach has been an area of interest in recent years, [20, 19, 5, 4]. However, the literature is mostly about constant SNR targets, i.e. target echo amplitude does not change over time as in [19], and also generally it is assumed that the target SNR is known as in [20, 4]. The algorithm derived here include the effects of target amplitude fluctuations according to Swerling models [23] and estimate the target amplitude as well as its kinematics.

### 3.2 Target and Sensor Model

In order to derive the algorithm let us first define the target and sensor models. Consider a target motion is modeled in Cartesian coordinate system according to the following discrete time dynamic model

$$x_{k+1} = f_k(x_k, w_k) \tag{3.1}$$

where  $k$  is the discrete time index,  $x_k \in R^n$  is the target state,  $f_k : R^n \times R^m \rightarrow R^n$  is a known

state transition function and  $w_k \in R^m$  is a white noise sequence with known pdf. Although Equation (3.1) defines a general dynamic system, many systems in practice can be modeled by two or more modes. For tracking applications for example, one mode can model nonmaneuvering targets whereas the other mode may account for the maneuvering targets. These kind of systems are called jump Markov (or Hybrid) systems which have a continuous valued state and a discrete valued mode. For these systems it is assumed that the system behaves according to one of the several modes.

A discrete time jump Markov system is described by the following dynamic and measurement equations respectively

$$x_{k+1} = f_k(x_k, m_k, w_k) \quad (3.2)$$

$$z_k = h_k(x_k, m_k, v_k) \quad (3.3)$$

where  $m_k$  is the discrete system mode,  $z_k \in R^p$  is the measurement vector which becomes available at time  $k$ .  $h_k \in R^n \times R^r \rightarrow R^p$  is the measurement function and  $v_k \in R^r$  is a white noise sequence with known pdf. Note that, both  $w_k$  and  $v_k$  are assumed to be independent of each other and also independent of the initial state. In order to define a complete stochastic system the initial condition  $p(x_0) = p(x_0|Z_{-1})$  is also assumed to be known. Note that, Equation (3.2) is equivalent to the one step prediction density  $p(x_{k+1}|x_k)$  and Equation (3.3) is equivalent to the measurement density  $p(z_k|x_k)$  such that

$$x_{k+1} \sim p(x_{k+1}|x_k, m_k) \quad (3.4)$$

$$z_k \sim p(z_k|x_k, m_k) \quad (3.5)$$

Thus, Equations (3.2) and (3.3) or equivalently (3.4) and (3.5) define the target and sensor models to be used in this thesis.

### 3.2.1 The Mode Variable

The mode variable is modeled by a first order Markov chain which has a transition probability matrix  $\Pi$ , with the following probabilities

$$\Pi = [\pi_{ij}] \quad (3.6)$$

$$\pi_{ij} \triangleq P\{m_k = j | m_{k-1} = i\} \quad (3.7)$$

Transition probability matrix  $\Pi$  also satisfies the following properties

$$\pi_{ij} \geq 0 \quad (3.8)$$

$$\sum_i \pi_{ij} = 1 \quad (3.9)$$

In the TBD approach considered in this work, the modes denote the '*target existence*' in the surveillance region such as

$$m_k = \begin{cases} 1 & ; \text{Target Exists} \\ 0 & ; \text{Target Does Not Exist} \end{cases} \quad (3.10)$$

The transition  $\{m_{k-1} = 0 \rightarrow m_k = 1\}$  is called the '*birth process*' with probability  $P_b$  and the transition  $\{m_{k-1} = 1 \rightarrow m_k = 0\}$  is called the '*death process*' with probability  $P_d$  defined as

$$P_b = P\{m_k = 1 | m_{k-1} = 0\} \quad (3.11)$$

$$P_d = P\{m_k = 0 | m_{k-1} = 1\} \quad (3.12)$$

The Markov Chain related to the target existence is illustrated in Figure 3.1 and the transition probability matrix is given in Equation (3.13).

$$\Pi = \begin{bmatrix} 1 - P_b & P_b \\ P_d & 1 - P_d \end{bmatrix} \quad (3.13)$$

Thus in order to use within a TBD algorithm, a multi-model jump Markov system is defined such that the mode variable indicates the target existence.

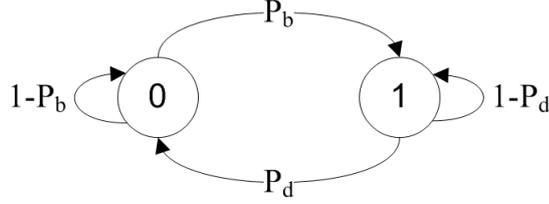


Figure 3.1: Illustration of Transition Probability Matrix

### 3.2.2 Target State Model

The state of the target is defined according to the appropriate motion model of the targets of interest which clarifies the state transition function  $f_k$  defined in Equation 3.2. For tracking applications, the motion model of a target can be best approximated in an absolute Cartesian coordinate system [16]. Therefore the state vector  $x_k$  is usually of the form

$$x_k = [x_k \ \dot{x}_k \ y_k \ \dot{y}_k \ z_k \ \dot{z}_k]^T \quad (3.14)$$

where  $[x_k \ y_k \ z_k]^T$  and  $[\dot{x}_k \ \dot{y}_k \ \dot{z}_k]^T$  denote the 3-D positions and velocities of the target respectively. In this work, in order to estimate the target amplitude the state vector is augmented such that

$$x_k = [x_k \ \dot{x}_k \ y_k \ \dot{y}_k \ z_k \ \dot{z}_k \ I_k]^T \quad (3.15)$$

where  $I_k$  is the target amplitude (intensity) variable.

Now, state Equation (3.2) can be rewritten depending on the mode variable such that

$$x_{k+1} = \begin{cases} f_k(x_k, w_k) & ; \text{Target Exists } (m_k = 1) \\ \text{undefined} & ; \text{Target Does Not Exist } (m_k = 0) \end{cases} \quad (3.16)$$

Equivalently transition density  $p(x_{k+1}|x_k)$  depending on the mode can be rewritten as

$$x_{k+1} \sim \begin{cases} p(x_{k+1}|x_k) & ; \text{Target Exists } (m_k = 1) \\ \text{undefined} & ; \text{Target Does Not Exist } (m_k = 0) \end{cases} \quad (3.17)$$

Note that the state transition density  $p(x_{k+1}|x_k)$  is assumed to be known although it is not restricted to any distributions or dynamics like linear-Gaussian case.

Thus, Equation (3.16) (or equivalently Equation (3.17)) defines the state model of the system at hand.

### 3.2.3 Measurement Model

In a TBD setup using a radar, it is assumed that the measurements are the so called Range Doppler (R-D) matrices described earlier in Chapter (2.1.4). For a fixed bearing and elevation angle the R-D matrix is calculated from the received echo of the transmitted signal. For completeness, let us define again the complex Range Doppler matrix for a fixed bearing and elevation angle at time  $k$  as  $\{z_k^{rd}\}$  where  $r = 1, \dots, r_m$  and  $d = 1, \dots, d_m$ .

The R-D matrix  $z_k^{rd}$  is a complex valued matrix including both phase and amplitude information. However, use of that complex matrix in the detection scheme is usually of the form of magnitude or power measurements [18, 19, 4]. In this thesis, we choose to use the power of the measurement matrix such that

$$z_k^{rd} = |z_k^{rd}|^2 \quad (3.18)$$

Figure 3.2 shows a sample R-D matrix which includes both target and noise originated reflections. The measurement equation is defined earlier in Equation (3.3). It is expected that if there is a target present in the radar line of sight, the received signal includes both target originated reflections and receiver induced noise. On the other hand, if the target is absent, it is expected that the received signal consists of the receiver noise only. So, the measurement function for each range and Doppler cell is written as

$$z_k^{rd} = \begin{cases} h_k(x_k, v_k) & ; \text{ Target Exists} \\ v_k & ; \text{ Target Does Not Exist} \end{cases} \quad (3.19)$$

Equivalently the measurement function can be defined by the following probability distributions

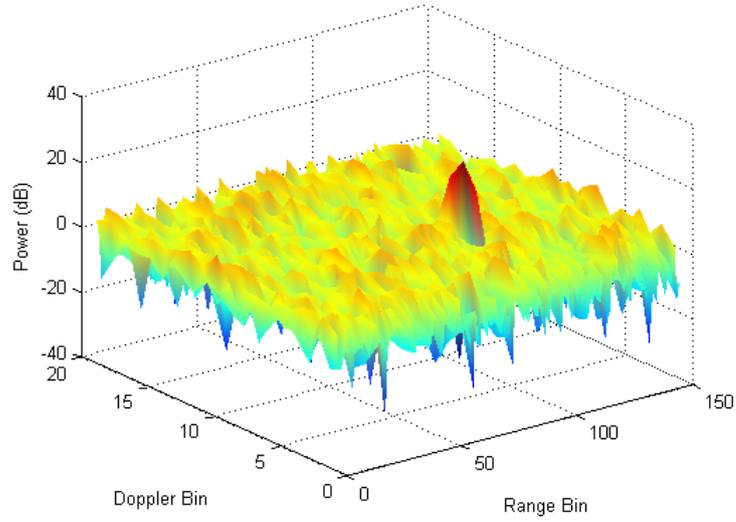


Figure 3.2: Sample Range Doppler Matrix

$$z_k^{rd} \sim \begin{cases} p_{S+N}(z_k|x_k) & ; \text{Target Exists} \\ p_N(z_k) & ; \text{Target Does Not Exist} \end{cases} \quad (3.20)$$

where  $p_N(z_k)$  denotes the pdf of the measurement noise and  $p_{S+N}(z_k|x_k)$  denotes the probability distribution of the target plus noise signal given that the target location is defined by state vector  $x_k$ .

The probability distribution of the measurement noise which is the magnitude squared of a complex Gaussian random variable  $\eta_k$  such that

$$v_k = |\eta_k|^2 \quad (3.21)$$

$$\eta_k = \sqrt{P_n/2} N(0, 1) + j \sqrt{P_n/2} N(0, 1) \quad (3.22)$$

where  $P_n$  is the noise power as defined earlier in Chapter 2.1.3.

$v_k$  is an 'exponentially' distributed random variable according to

$$p_N(z_k) = \begin{cases} \frac{1}{P_n} e^{-x/P_n} & ; x \geq 0 \\ 0 & ; x \leq 0 \end{cases} \quad (3.23)$$

Let us now consider the target plus noise signal distribution. According to the Swerling-I model, target plus noise signal power fluctuations from scan to scan is modeled as an 'exponential' distribution [18, 23].

$$p_{S+N}(z_k|x_k) \sim \begin{cases} \frac{1}{P_r} e^{-x/P_r} & ; x \geq 0 \\ 0 & ; x \leq 0 \end{cases} \quad (3.24)$$

Then, Equation (3.20) can be rewritten with the following likelihoods

$$p(z_k|x_k, m_k) \sim \begin{cases} \prod_{r,d} p_{S+N}(z_k^{rd}|x_k) & \text{for } m_k = 1 \\ \prod_{r,d} p_N(z_k^{rd}) & \text{for } m_k = 0 \end{cases} \quad (3.25)$$

Here,  $p_N(z_k^{rd})$  is the pdf of the received noise at range bin of  $r$  and Doppler bin of  $d$ .  $p_{S+N}(z_k^{rd}|x_k)$  is the pdf of target plus noise signal in the  $(r, d)$  bin of the measurement matrix given that the target is at the location defined by  $x_k$ . Note that the target and/or noise distributions for each resolution cell is assumed to be independent of each other hence the product of the pdf's in the above likelihood, see Equation (3.25).

Consider a target present in the surveillance region and its centroid is at the position  $(r_t, d_t)$ . The target may contribute to the resolution cells in its vicinity. This phenomena is called the 'Blurring', [20]. Let us define the contribution of the target to the cell  $(r, d)$  is denoted as  $b(r, d, r_t, d_t)$  where the target is located at  $(r_t, d_t)$ . If the target is a point object blurring function is used to model the 'Point Spread function' of the sensor itself. On the other hand, if the target is an extended object with respect to the resolution cells, blurring function is used to model this effect also.

In practice,  $b(r, d, r_t, d_t)$  is a non zero function only in some of the neighbourhood of  $(r_t, d_t)$ . We denote this neighbourhood area by  $B(x_k)$  and restrict the likelihood computation of Equation (3.25) to this area for target returns.

Let us define the  $B(x_k)$  indicates the affected  $(r, d)$  bins by the target with state vector  $x_k$ . In order to account for the blurring effect, pdf of the target plus noise density can be rewritten as follows

$$\begin{aligned} p(z_k|x_k, m_k = 1) &= \prod_{r,d} p_{S+N}(z_k^{rd}|x_k) \\ &\cong \prod_{r,d \in B(x_k)} p_{S+N}(z_k^{rd}|x_k) \prod_{r,d \notin B(x_k)} p_N(z_k^{rd}) \end{aligned} \quad (3.26)$$

Therefore the probability density function of the measurement matrix depending on the target existence is obtained as the product of the following likelihoods

$$p(z_k|x_k, m_k) = \begin{cases} \prod_{r,d \in B(x_k)} p_{S+N}(z_k^{rd}|x_k) \prod_{r,d \notin B(x_k)} p_N(z_k^{rd}) & \text{for } m_k = 1 \\ \prod_{r,d} p_N(z_k^{rd}) & \text{for } m_k = 0 \end{cases} \quad (3.27)$$

Note that, particle filters can accommodate structured and/or non-Gaussian background noise provided that its distribution is known. That means,  $p_N(z_k^{rd})$  can be used to model undesired background effects such as clutter. For example, it would be possible to learn the probability density of  $p_N(z_k^{rd})$  from a stationary background given that a target is not present during the learning process.

### 3.3 Bayesian Solution of the TBD Problem

The aim of the TBD algorithm developed in this thesis is to construct the joint posterior density of the target state  $x_k$  and it's existence  $m_k$  given all the available information as  $p(x_k, m_k|Z_k)$ . Bayesian approach for the computation of the posterior pdf is solved recursively in two stages: Prediction and Update [19]. Considering that the joint posterior pdf at time  $k - 1$ ,  $p(x_{k-1}, m_{k-1}|Z_{k-1})$  is known, one can obtain the prior pdf of the state with the following equations

$$\begin{aligned}
p(x_k, m_k = 1|Z_{k-1}) &= \int p(x_k, m_k = 1|x_{k-1}, m_{k-1} = 1, Z_{k-1}) \cdot \\
&\quad p(x_{k-1}, m_{k-1} = 1|Z_{k-1}) dx_{k-1} \\
&+ \int p(x_k, m_k = 1|x_{k-1}, m_{k-1} = 0, Z_{k-1}) \cdot \\
&\quad p(x_{k-1}, m_{k-1} = 0|Z_{k-1}) dx_{k-1}
\end{aligned} \tag{3.28}$$

First consider the first term of the integral sum

$$\begin{aligned}
\int p(x_k, m_k = 1|x_{k-1}, m_{k-1} = 1, Z_{k-1}) &= p(x_k|x_{k-1}, m_k = 1, m_{k-1} = 1) \cdot \\
&\quad P\{m_k = 1|m_{k-1} = 1\}
\end{aligned} \tag{3.29}$$

Here  $p(x_k|x_{k-1}, m_k = 1, m_{k-1} = 1)$  is the conditional pdf of the target motion defined by Equation (3.2) and  $P\{m_k = 1|m_{k-1} = 1\}$  is the survival probability from the Markov Chain (3.13). Then it can be written as

$$\int p(x_k, m_k = 1|x_{k-1}, m_{k-1} = 1, Z_{k-1}) = p(x_k|x_{k-1}) \cdot (1 - P_d) \tag{3.30}$$

Now, consider the second term

$$\begin{aligned}
p(x_k, m_k = 1|x_{k-1}, m_{k-1} = 0, Z_{k-1}) &= p(x_k|x_{k-1}, m_k = 1, m_{k-1} = 0) \cdot \\
&\quad P\{m_k = 1|m_{k-1} = 0\}
\end{aligned} \tag{3.31}$$

Here  $p(x_k|x_{k-1}, m_k = 1, m_{k-1} = 0)$  is the probability density of the state during which is denoted as  $p_b(x_k)$  and  $P\{m_k = 1|m_{k-1} = 0\}$  is the birth probability of the Markov Chain (3.13). Then

$$p(x_k, m_k = 1|x_{k-1}, m_{k-1} = 0, Z_{k-1}) = p_b(x_k) \cdot P_b \tag{3.32}$$

As it can be seen from the derivation prediction equations consist of two terms. The first term, Equation (3.29) accounts for the target being exist at time  $k - 1$ , stays existing at time  $k$ . It

declares that the target did not suffer from 'death' process and stays alive with probability  $1 - P_d$ . Second term, Equation (3.31) accounts for the 'birth' process where target did not exist at time  $k - 1$ , but it exists at time  $k$  with probability  $P_b$ . Also note that for the target not existing case ( $m_k = 0$ ), target state  $x_k$  is undefined and did not taken into account for the prediction step. Therefore the prediction step can be summarized as

$$p(x_k|x_{k-1}, m_k = 1, m_{k-1}) = \begin{cases} p(x_k|x_{k-1}) & \text{for } m_{k-1} = 1 \\ p_b(x_k) & \text{for } m_{k-1} = 0 \end{cases} \quad (3.33)$$

Now we can get into the measurement update equation as follows

$$p(x_k, m_k|Z_k) = \frac{p(z_k|x_k, m_k) p(x_k, m_k|Z_{k-1})}{p(z_k|Z_{k-1})} \quad (3.34)$$

Here, the term  $p(z_k|x_k, m_k)$  is the probability density of the measurement and it is assumed to be known.  $p(x_k, m_k|Z_{k-1})$  is the prior density calculated in the prediction step.  $p(z_k|Z_{k-1})$  is a normalizing denominator. Thus the measurement update equation is obtained.

Finally, the posterior probability of target existence at time  $k$  is the marginal density given below

$$P_{E_k} \triangleq p(m_k = 1|Z_k) = \int p(x_k, m_k|Z_k) dx_k \quad (3.35)$$

Thus here we have defined a recursive Bayesian solution of the TBD problem. The implementation of the solution is derived in the next section's using particle filters.

### 3.4 Particle Filtering

The main idea of particle filtering is to represent the required pdf as a set of random samples. As the number of samples become very large, PF algorithm approaches to the optimal Bayesian estimator. Therefore the aim of this section is to use particle filters to realize the iterative Bayesian solution (i.e. prediction and update) of the TBD problem defined in the previous section.

The PF algorithm proposed here is a modified version of the SIR filter. Note that the algorithm is developed on the system and measurement models developed earlier in this chapter.

### 3.4.1 Initialization

Let us assume that the initial distributions of the state and mode variable are known and are denoted as  $p_{x_0}(x_0^i)$  and  $p_{m_0}(m_0^i)$  respectively. Assume that we have a set of random samples  $\{x_0^i, m_0^i\}_{i=1}^N$  taken from the initial distributions such that

$$x_0^i \sim p_{x_0}(x_0^i) \quad (3.36)$$

$$m_0^i \sim p_{m_0}(m_0^i) \quad (3.37)$$

Considering that there will be no information about the target existence at the initial step,  $m_0^i$  can be selected as a binary random variable with uniform distribution. Then, for those particles who claim a target existence i.e.  $m_0^i = 1$ , one can generate initial state variables according to any importance density function. In this thesis the initial state distribution  $p_{x_0}(x_0^i)$  is assumed to be uniform over the surveillance region of the state space.

### 3.4.2 Time Update

Let us assume that at time  $k - 1$ , the joint state posterior pdf  $p(x_{k-1}, m_{k-1} | Z_{k-1})$  is known and we have a set of random samples  $\{x_{k-1}^i, w_{k-1}^i\}_{i=1}^N$  associated with this pdf. The first step for each particle is to calculate the mode prior  $p(m_k^i | m_{k-1}^i)$  using  $m_{k-1}^i$  and transition probability matrix,  $\Pi$ . This is done by implementing a two state Markov chain with transition probability  $\pi_{ij}$  as described earlier.

The next step for each particle is to calculate the target state prior,  $p(x_k^i | x_{k-1}^i)$ . However, this will be done only for the particles that indicate a target is present, i.e.  $m_k^i = 1$ . As noted before, for others target state is undefined. Two possible cases occur for the prediction step which are introduced earlier in Section 3.3.

### 3.4.2.1 Existing Particles

The first case of time update is so called '*Existing Particles*' where the  $i^{th}$  particle stays alive between the times  $k - 1$  and  $k$ , i.e.  $m_{k-1} = 1$  and  $m_k = 1$ .

It has been shown that the selection of the importance density plays a crucial role in the performance of the PF algorithm. One ideally desire to use the true posterior density as the importance density. However in TBD problem this is not possible. Thus for existing particles we have selected the transitional prior as the importance density such that

$$q(x_k^i | x_{k-1}^i, z_k) = p(x_k^i | x_{k-1}^i) \quad (3.38)$$

Considering that the target dynamic model is known, one can obtain transitional prior density of the state given the previous value. In the particle representation it is obtained by drawing samples from the transitional prior density as

$$x_k^i \sim p(x_k | x_{k-1}^i) \quad (3.39)$$

Note that obtaining state distribution  $x_k^i$  from the transitional prior  $p(x_k | x_{k-1}^i)$  defines a simple but inefficient representation of the true prior.

### 3.4.2.2 Newborn Particles

The second case of the time update is the birth of new particles which are so called '*Newborn Particles*' where the  $i^{th}$  particle borns between the times  $k - 1$  and  $k$ . This phenomena is characterized by  $m_{k-1}^i = 0$  and  $m_k^i = 1$ .

The birth probability density function  $p_b(x_k)$  is assumed to be known. Depending on the application there are several methods to get the birth function for newborn particles

- It may be uniform over the surveillance region
- It may be based on the measurement
- Any a-priori information about target position, velocity or SNR may be used

In this work in order to use the maximum information, newborn particles are drawn using the measurement itself. For this reason, the R-D matrix is considered as a pdf and samples will be drawn from higher return amplitudes.

This technique is sometimes called as the 'likelihood' particle filter, see [1], where the measurement pdf is also used for existing particles. However, it is obvious that for low SNR applications prior density is likely to have more information than the measurement itself. Thus, in this TBD application we offer the measurement likelihood pdf is used only for the newborn particles.

An Acceptance-Rejection (A-R) algorithm is proposed in order to draw more particles from 'high' value (i.e. high SNR) parts of the measurement matrix. Let  $S_k$  denote the 'Surveillance Region',  $z_k$  is the R-D matrix and  $k$  is the discrete time index. The algorithm's pseudo-code is given in Algorithm 6.

Figure 3.3 illustrates the measurement data (on the left) and histogram of drawn particles based on that data (on the right) with A-R technique proposed here.

---

**Algorithm 6** Acceptance Rejection Algorithm Structure

---

**function**  $[x_k^i] = \text{A-R}(z_k, S_k)$

- *FOR*  $i = 1 : \text{max iteration}$ 
    - Draw a sample from  $S_k$  such that:  $x_k^i \sim U[S_{k_{min}}, S_{k_{max}}]$
    - Draw a value  $u$  such that:  $u \sim U[0, 1]$
    - *IF*  $u \leq h(x_k^i) / \text{max}(z_k)$ 
      - \* *Return*:  $x_k^i$
    - *ELSE*
      - \* *Continue*
    - *END*
  - *END*
-

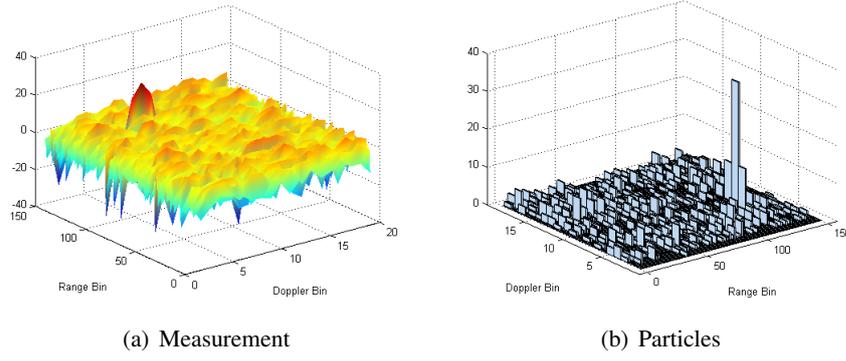


Figure 3.3: Histogram of Newborn Particles

### 3.4.3 Measurement Update

Measurement update of the TBD algorithm is actually a weight update of the PF. The update equation is to calculate 'unnormalized importance weights' defined as

$$\tilde{w}_k^i \propto w_{k-1}^i \cdot \frac{p(z_k | x_k^i, m_k^i) p(x_k^i | x_{k-1}^i, m_k^i)}{q(x_k^i | x_{k-1}^i, z_k)} \quad (3.40)$$

where the importance density  $q(x_k^i | x_{k-1}^i, z_k)$  is selected as the transitional prior density  $p(x_k^i | x_{k-1}^i)$ . Therefore the update equation takes the form

$$\tilde{w}_k^i \propto w_{k-1}^i p(z_k | x_k^i) \quad (3.41)$$

Note that since we apply resampling at each time step normalized weights are equal to  $1/N$  at the end of each time instance and thus unnormalized weights are not dependent on  $w_{k-1}^i$ . Therefore, it is only required to calculate the likelihoods  $p(z_k | x_k^i)$  at each time instance. The likelihood function is derived earlier in Equation (3.27) however for the sake of completeness let us give the equation once more as follows

$$p(z_k | x_k, m_k) = \begin{cases} \prod_{r,d \in B(x_k)} PS+N(z_k^{rd} | x_k) \prod_{r,d \notin B(x_k)} PN(z_k^{rd}) & \text{for } m_k = 1 \\ \prod_{r,d} PN(z_k^{rd}) & \text{for } m_k = 0 \end{cases} \quad (3.42)$$

Considering that the weights are only required upto proportionality we can divide Equation

(3.42) by  $\prod_{r,d} p_N(z_k^{rd})$ , [20]. So the measurement likelihood becomes

$$\tilde{w}_k^i \propto p(z_k|x_k^i, m_k^i) \propto \begin{cases} \prod_{r,d \in B(x_k)} L(z_k^{rd}|x_k^i) & \text{for } m_k^i = 1 \\ 1 & \text{for } m_k^i = 0 \end{cases} \quad (3.43)$$

where  $L(z_k^{rd}|x_k^i)$  is the 'likelihood ratio' defined as,

$$L(z_k^{rd}|x_k^i) = \frac{p_{S+N}(z_k^{rd}|x_k^i)}{p_N(z_k^{rd})} \quad (3.44)$$

Therefore the unnormalized weight of each particle is equal to the product of the likelihood ratios in the vicinity of the particle. Notice that unnormalized weights defined by the equation (3.43) has a very similar expression with the Neyman-Pearson detection rule defined in Section 2.2.

### 3.4.4 Normalize

In order to approximate the true posterior the weights are normalized as

$$w_k^i = \frac{\tilde{w}_k^i}{\sum_{i=1}^N \tilde{w}_k^i} \quad (3.45)$$

Thus after normalization the discrete approximation of the true posterior is obtained as

$$p(x_k|Z_k) \approx \sum_{i=1}^N w_k^i \delta(x_k - x_k^i) \quad (3.46)$$

### 3.4.5 Output

Since the accuracy of any estimate of a distribution can only decrease with resampling, [14], the output of the running PF is calculated prior to resampling. Let us define the output of the algorithm at time  $k$  as, the *Probability of Target Existence*,  $P_{E_k}$  and the *Minimum Mean-Square Error (MMSE) Estimate* of the state vector,  $\hat{x}_{k|k}^{MMSE}$ .

The probability of target existence is calculated by its definition

$$P_{E_k} \triangleq p(m_k = 1|Z_k) \quad (3.47)$$

which is equal to

$$P_{E_k} = \sum_{i=1}^N m_k^i w_k^i \quad (3.48)$$

The MMSE estimator is defined by the conditional expectation of  $x_k$  given the measurement sequence  $Z_k$

$$\hat{x}_{k|k}^{MMSE} \triangleq E[x_k|Z_k] = \int x_k p(x_k|Z_k) dx_k \quad (3.49)$$

Consequently, the MMSE estimate of the state can be calculated as follows

$$\hat{x}_{k|k} = \frac{\sum_{i=1}^N x_k^i m_k^i w_k^i}{\sum_{i=1}^N m_k^i w_k^i} \quad (3.50)$$

### 3.4.6 Resample

Sample degeneracy is defined as after a finite number of steps, depending on how close the importance density to the true posterior is, all particles but a few will have negligible normalized weights. This is an unavoidable problem for particle filters and resampling tries to overcome this problem. The aim of resampling is to increase the effective number of particles calculated below

$$\hat{N}_{eff} = \frac{1}{\sum_i (w_{k|k}^i)^2} \quad (3.51)$$

Detailed discussion about degeneracy has been made in Section 2.3.2. For those reasons resampling is included in the TBD algorithm as defined in Algorithm 2.

### 3.4.7 MCMC Move Step

In Chapter 2.3.4 it has been shown that the loss of diversity is a severe problem of particles filters especially when the process noise is small or a parameter estimation is being made. Notice that in this algorithm target mean power is estimated as a parameter using the model

$$I_k = I_{k-1} \quad (3.52)$$

Therefore MCMC move step is applied only for the target power such that

$$I_k^{*i} = I_k^i + bD_k\epsilon^i \quad (3.53)$$

where  $b$  is the Gaussian kernel bandwidth and  $D_k$  is the standart deviation obtained for the target power at time  $k$ . In [19] it is given that the optimum bandwidth for a Gaussian kernel is given by

$$b = \left[ \frac{4}{n+2} \right]^{\frac{1}{n+4}} N^{-\frac{1}{n+4}} \quad (3.54)$$

where  $N$  is the number of particles and  $n$  is the number of state dimensions. See Algorithm 7 for the implementation of the MCMC move step in the TBD algorithm. This is the final step of the particle filtering part of the TBD algorithm defined in this chapter. Pseudo-code of the algorithm is given in Algorithm 7.

## 3.5 Detection

In this section, it will be determined how to perform detection from the output of the particle filtering algorithm defined above. As explained in Chapter 2.2 two hypotheses are defined as

1. Noise Present Only  $\quad \quad \quad : H_0$
2. Signal plus Noise Present  $\quad \quad \quad : H_1$

where the detection is based on the likelihood ratio test defined as

$$L(z_k) = \frac{p(z_k|H_1)}{p(z_k|H_0)} \quad (3.55)$$

It is given in Equation (3.20) that, the probability density of the measurement given that a target is present, i.e.  $p(z_k|H_1)$  is defined by  $p_{S+N}(z_k|x_k)$  and the probability density of the measurement given that a target is absent, i.e.  $p(z_k|H_0)$  is defined by  $p_N(z_k)$ . Thus the likelihood ratio test can be written as

$$L(z_k) = \frac{p(z_k|H_1)}{p(z_k|H_0)} = \frac{p_{S+N}(z_k|x_k)}{p_N(z_k)} \underset{H_0}{\overset{H_1}{>}} \lambda \quad (3.56)$$

Now for the sake of completeness let us repeat the update equation of the PF algorithm defined in Equation (3.43) as

$$\tilde{w}_k^i \propto p(z_k|x_k^i, m_k^i) \propto \begin{cases} \prod_{r,d \in B(x_k)} L(z_k^{rd}|x_k^i) & \text{for } m_k^i = 1 \\ 1 & \text{for } m_k^i = 0 \end{cases} \quad (3.57)$$

where

$$L(z_k^{rd}|x_k^i) = \frac{p_{S+N}(z_k^{rd}|x_k^i)}{p_N(z_k^{rd})} \quad (3.58)$$

Notice that in the PF algorithm the likelihood ratio is inherently calculated in the weight update equation. Thus for those particles where  $m_k^i = 1$ , the sum of the importance weights can be used as the likelihood ratio.

$$L(z_k) \cong \sum_{i=1}^N m_k^i w_k^i \quad (3.59)$$

Remember that the probability of target existence is defined in Equation (3.47) as the weighted sum of the importance weights where the weights are the mode variable.

$$P_{E_k} = \sum_{i=1}^N m_k^i w_k^i \quad (3.60)$$

So that the likelihood ratio test can be realized based on the probability of target existence such that

$$P_{E_k} \underset{H_0}{\overset{H_1}{>}} \lambda_d \quad (3.61)$$

It is shown that the required likelihood ratio test is inherently implemented in particle filtering algorithm and by its definition  $P_{E_k}$  can be used in order to declare the target existence in the measurement matrix under test.

---

**Algorithm 7** TBD Algorithm

---

**function**  $[\hat{x}_{k|k}, P_{E_k}] = \text{TBD}(z_k, S_k)$

- Initialize particles using  $p_{x_0}(x_0^i)$  and  $p_{m_0}(m_0^i)$
- *FOR*  $i = 1 : N$ 
  - Evaluate Markov Chain from:  $p(m_k^i | m_{k-1}^i)$ 
    - \* *If* existing particle
      - Draw sample from:  $p(m_k^i | m_{k-1}^i)$
    - \* *If* newborn particle
      - Draw sample from:  $q_b(x_k^i | z_k)$  using Algorithm 6
  - Evaluate 'Importance Weights'

$$\tilde{w}_k^i \propto p(z_k | x_k^i, m_k^i) \propto \begin{cases} \prod_{r,d \in B(x_k)} L(z_k^{rd} | x_k^i) & \text{for } m_k^i = 1 \\ 1 & \text{for } m_k^i = 0 \end{cases}$$

$$L(z_k^{rd} | x_k^i) = \frac{p_{S+N}(z_k^{rd} | x_k^i)}{p_N(z_k^{rd})}$$

- *END*
  - Continue from Algorithm 8
-

---

**Algorithm 8** TBD Algorithm (ctd.)

---

**function**  $[\hat{x}_{k|k}, P_{E_k}] = \text{TBD}(z_k, S_k)$ 

- *FOR*  $i = 1 : N$

- Normalize

$$w_k^i = \frac{\tilde{w}_k^i}{\sum_{i=1}^N \tilde{w}_k^i}$$

- *END*

- Calculate Output

$$P_{E_k} = \sum_{i=1}^N m_k^i w_k^i$$
$$\hat{x}_{k|k} = \frac{\sum_{i=1}^N x_k^i m_k^i w_k^i}{\sum_{i=1}^N m_k^i w_k^i}$$

- Calculate  $D_k$  such that  $D_k D_k^T = S_k$  where  $S_k$  is the variance of target power

- Resample using Algorithm 2

- *FOR*  $i = 1 : N$

- Draw  $\epsilon^i$  from Gaussian kernel and calculate  $I_k^{*i} = I_k^i + b D_k \epsilon^i$

- Evaluate probability of move according to Equation (2.41)

- Draw a candidate  $u$  such that  $u \sim U[0, 1]$

- \* *IF*  $u \leq \alpha(x_k^i, x_k^{*i})$

- Move to the new location such that:  $x_k^i = x_k^{*i}$

- \* *ELSE*

- Reject move such that:  $x_k^i = x_k^i$

- \* *END*

- *END*

- Declare target existence based on

$$P_{E_k} \begin{matrix} > \\ < \end{matrix} \begin{matrix} H_1 \\ H_0 \end{matrix} \lambda_d$$

## CHAPTER 4

### RESULTS AND SIMULATIONS

#### 4.1 Simulation Environment

To evaluate the performance of the proposed TBD algorithm, a radar simulator is used as the raw data generator [15]. Radar simulator is a MATLAB based simulator which is capable of generating echo signals based on radar, target and environmental parameters. Radar parameters are tuned to obtain different SNR's for different target scenario's. Radar simulator is capable of simulating target reflections for different Swerling models. As it will be discussed in the later sections Swerling-0 and Swerling-1 type of targets are considered within this thesis. Radar simulator is also capable of simulating various environmental effects although we have assumed a free space for the simulations.

At the output, the radar simulator generates Range Doppler (R-D) matrices for each run. A slight modification has been made to the simulator in order to generate the R-D matrices for a given target trajectory. A sample output of the simulator is shown in Figure 3.2. Note that for all of the scenarios R-D matrices have 160 range cells versus 14 Doppler cells.

#### 4.2 Scenario and Parameters

To analyze the performance of the TBD algorithm developed in Chapter 3, the following scenarios are used

1. Incoming Target with Constant Amplitude
2. Incoming Target with Fluctuating Amplitude

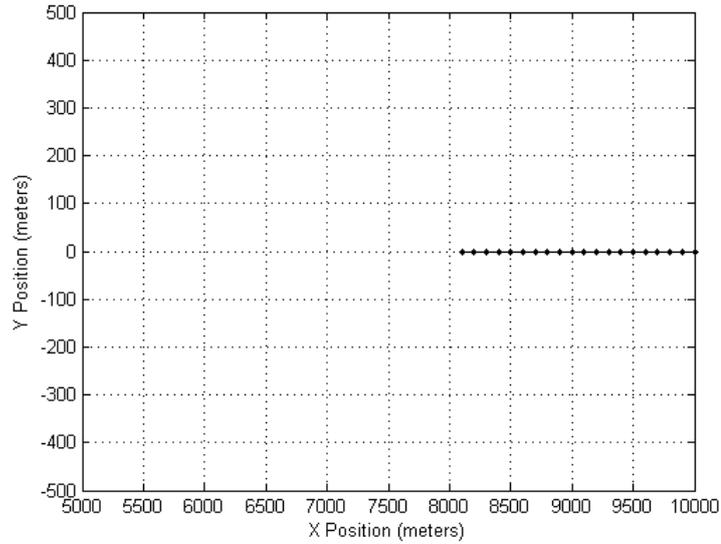


Figure 4.1: Incoming Target Trajectory

3. Maneuvering Target with Fluctuating Amplitude
4. Noise Only Detection

The simulations aim to analyze the performance of the algorithm for different aspects. Constant target amplitude i.e. Swerling-0 case is of course not realistic for most of the time however it provides a base performance for the other cases. Incoming target scenario with fluctuating target amplitude i.e. Swerling-1 aims to analyze the proposed algorithm's performance under straight line trajectories. This is a commonly used scenario to evaluate the track initiation and tracking algorithm performances in practice. Maneuvering target with fluctuating amplitude case is used to analyze the performance degradation due to the target motion uncertainties. Finally, noise only scenario specifies the false track declaration performance of the algorithm.

#### 4.2.1 Target Scenario

Let us define two different target scenarios to be used within the simulations.

In the first scenario an incoming target is assumed to appear at time  $t = 6 \text{ sec}$  with a radial velocity of  $100 \text{ m/sec}$ . It is assumed that the target stays in the surveillance region for 20

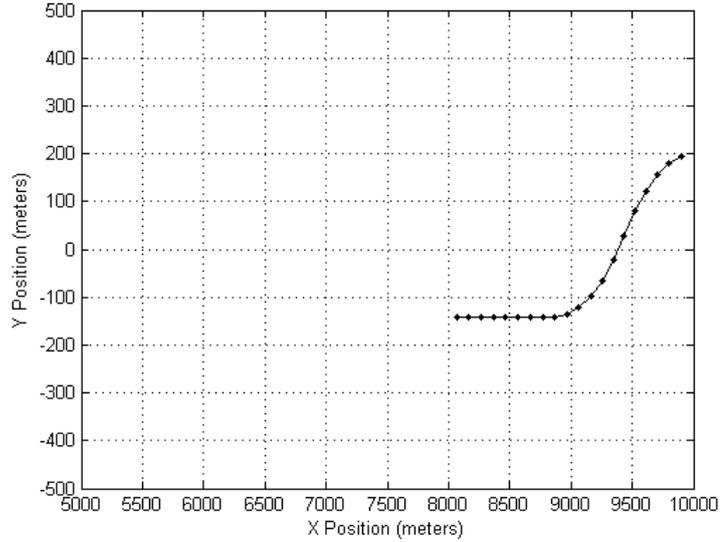


Figure 4.2: Maneuvering Target Scenario

seconds and then disappears. Figure 4.1 shows the trajectory of the target in  $X - Y$  plane. It is assumed that the target altitude stays constant throughout the scenario.

In the second scenario a Maneuvering target is assumed to appear again at time step of  $t = 6 \text{ sec}$  with a radial velocity of  $100 \text{ m/s}$ . However, the target maneuvers with  $1g$  for 12 seconds and then settles in a straight line trajectory. Figure 4.2 shows the trajectory of the target in  $X - Y$  plane. It is again assumed that the target altitude stays constant during the maneuver.

Let us assume that the radar is located at position  $(0, 0)$  and has an antenna rotation period of 1 second.

#### 4.2.2 Surveillance Region

Surveillance region indicates the expected position, velocity and amplitude of targets. Table 4.1 shows the surveillance region parameters of 7 dimensional state variable  $x_k$ . Note that the mode variable  $m_k$  is binary.

As it is noted before, initial state distribution is assumed to be uniform over the surveillance region for the state variable  $x_0^i$  and the mode variable  $m_0^i$ . Figure 4.3 illustrates the initially drawn particles in  $X - Y$  plane using the surveillance region parameters.

Table 4.1: Surveillance Region Parameters

Parameter	Min Value	Max Value	Unit
X Position	7000	10000	meters
X Velocity	-150	-50	meters/sec
Y Position	-250	250	meters
Y Velocity	-10	10	meters/sec
Z Position	-100	100	meters
Z Velocity	-10	10	meters/sec
Target SNR	6	26	dB

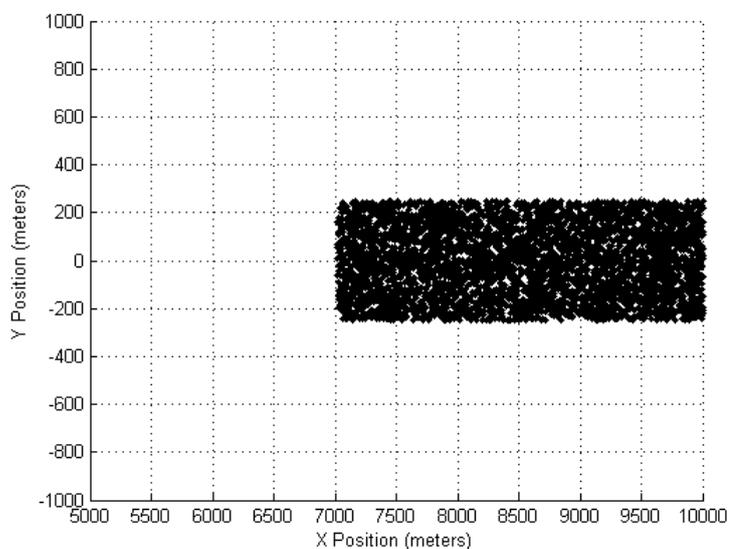


Figure 4.3: Initial Position of Particles

Note that for those particles who do not claim a target existence at time 0, i.e.  $m_0^i = 0$ , target state is undefined.

### 4.2.3 The Mode Variable

The mode transition probability matrix is defined as

$$\Pi = \begin{bmatrix} 1 - P_b & P_b \\ P_d & 1 - P_d \end{bmatrix} \quad (4.1)$$

where the 'birth' and 'death' probabilities are taken as follows

$$P_b = 0.05 \quad (4.2)$$

$$P_d = 0.01 \quad (4.3)$$

#### 4.2.4 Target State Model

Let the state defined by the Equation (3.16), is modeled with a discrete time linear stochastic system with the following equations

$$x_{k+1} = \begin{cases} Fx_k + Gw_k & \text{if } m_k = 1 \\ \text{undefined} & \text{if } m_k = 0 \end{cases} \quad (4.4)$$

In this expression the state vector  $x_k$  is defined as  $x_k = [x_k \ \dot{x}_k \ y_k \ \dot{y}_k \ z_k \ \dot{z}_k \ I_k]^T$ ,  $F$  is the state transition matrix,  $w_k$  is the process noise sequence and  $G$  is noise gain. State transition matrix is taken as

$$F_k = \begin{bmatrix} \tilde{F} & 0 & 0 & 0 \\ 0 & \tilde{F} & 0 & 0 \\ 0 & 0 & \tilde{F} & 0 \\ 0 & 0 & 0 & A(x) \end{bmatrix} \quad (4.5)$$

where

$$\tilde{F} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \quad (4.6)$$

and

$$A(x) = (r_{k-1}/r_k) \quad (4.7)$$

where  $r$  denotes the range of the target such that  $r_k = \sqrt{x_k^2 + y_k^2 + z_k^2}$ .

The noise gain is defined as

$$G = \begin{bmatrix} \tilde{G} & 0 & 0 & 0 \\ 0 & \tilde{G} & 0 & 0 \\ 0 & 0 & \tilde{G} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (4.8)$$

where

$$\tilde{G} = \begin{bmatrix} T^2/2 \\ T \end{bmatrix} \quad (4.9)$$

Finally, the discrete process noise sequence is assumed to be white Gaussian according to

$$w_k \sim N(w_k; 0, Q_k) \quad (4.10)$$

where

$$Q_k = \begin{bmatrix} \tilde{q}_k & 0 & 0 & 0 \\ 0 & \tilde{q}_k & 0 & 0 \\ 0 & 0 & \tilde{q}_k & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (4.11)$$

Note that,  $\tilde{q}_k$  in the above equation is used to model the unknown acceleration of the target motion for  $x - y - z$  axes. This model is known as the 'Constant Velocity, CV' or 'Nearly Constant Velocity' motion model [16]. The covariance matrix  $Q_k$  determines the expected amount of acceleration, i.e maneuver of the target. As noted before in this work state vector is augmented in order to include the target power,  $I_k$  and thus the CV motion model is also augmented. Target echo power is assumed to be range dependent.

Equations (4.4) to (4.11) defines a discrete time non-linear stochastic system to be used within the simulations.

## 4.2.5 Target Measurement Model

The measurement model of the target is assumed to be

$$p(z_k|x_k, m_k) = \begin{cases} \prod_{r,d} p_{S+N}(z_k^{rd}|x_k) & \text{for } m_k = 1 \\ \prod_{r,d} p_N(z_k^{rd}) & \text{for } m_k = 0 \end{cases} \quad (4.12)$$

where both  $p_{S+N}(z_k^{rd}|x_k)$  and  $p_N(z_k^{rd})$  are assumed to be exponential densities as defined in Chapter 3.2.3. Also note that no blurring effect is taken into account in the measurement model such that  $B(x) = \delta(x_k^i)$ .

## 4.2.6 Detection

Finally the target is declared once the probability of target existence is above the predefined threshold,  $P_{E_k} \geq 0.6$ .

## 4.3 Simulation Results

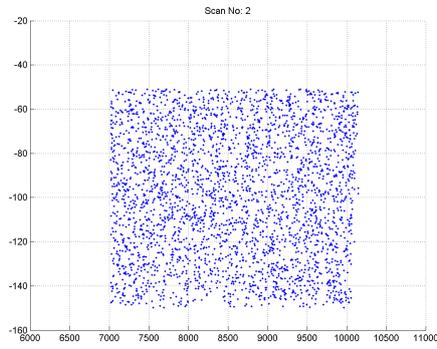
### 4.3.1 Incoming Target with Constant Amplitude

In the constant target amplitude analysis, an incoming target scenario is used as shown Figure 4.1. It is assumed that the target return power does not fluctuate from mean SNR. This scenario matches to the Swerling-0 model as defined in [23]. Note that one would expect an increase on the target SNR as it gets closer to the radar. This effect does not depend on Swerling models and is taken into account within the simulations.

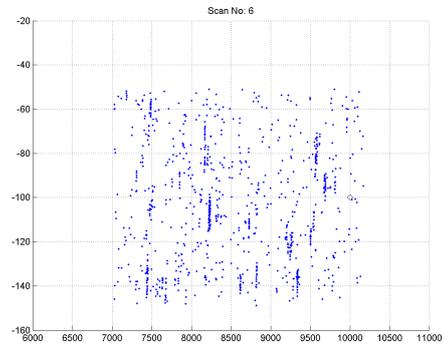
Simulation results are given below for different initial target SNR's and different number of particles. Note that all of the results are obtained by 100 Monte Carlo simulations.

### 10dB Initial Target SNR

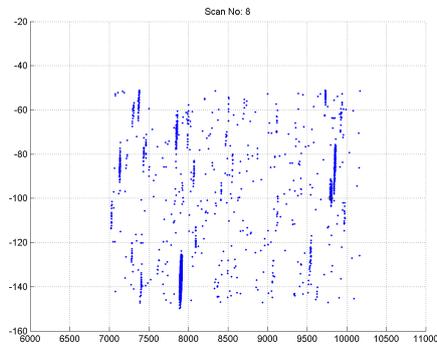
In the first experiment a relatively high SNR value is chosen to obtain a base performance of the TBD algorithm. Figure 4.4 shows the behaviour of the particles for  $t = 2, 6, 8, 10, 16$  and  $33$



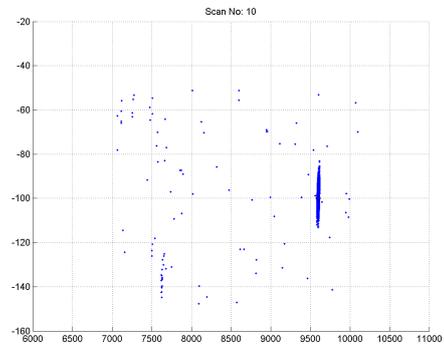
(a)  $t = 2sec$



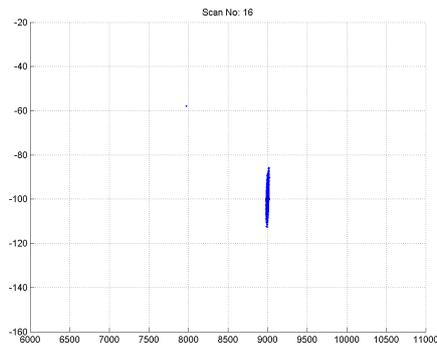
(b)  $t = 6sec$



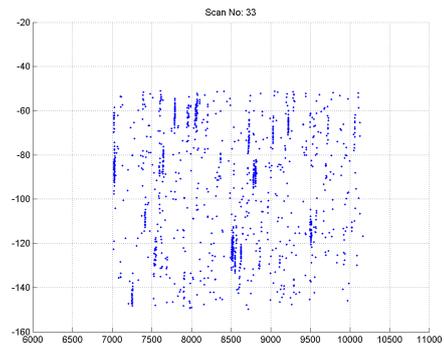
(c)  $t = 8sec$



(d)  $t = 10sec$



(e)  $t = 16sec$



(f)  $t = 33sec$

Figure 4.4: Distributions of Particles in R-D Plane

seconds. Blue dots indicate particle positions and black 'o' denotes the actual target position when it is in the surveillance region. It is shown in the figure that when the target does not exist particles remain uniform over the surveillance region. On the other hand, when the target appears at time  $t = 6\text{sec}$  particles tend to accumulate around it. Those particles who accumulate around the target defines the probability of target existence. Finally, it is also shown that the particles become uniformly distributed after the target disappears at  $t = 26\text{sec}$ .

Figure 4.5 shows the probability of target existence calculated by the TBD algorithm together with target SNR. Left axis shows the probability of target existence for different number of particles while right axis shows the target SNR in the same figure.

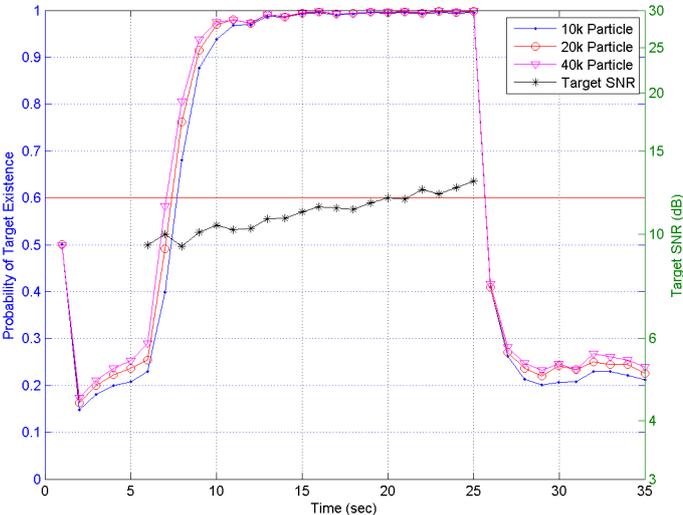


Figure 4.5: Probability of Target Existence and Target SNR (Swerling-0, 10dB Initial SNR)

Figure 4.5 shows that for a 10dB initial target SNR, proposed TBD algorithm performs target declaration in 3 seconds. It is also shown that increasing the number of particles slightly increase the confidence of TBD algorithm in the transient region about the target existence although it does not affect the overall detection performance.

The output of the PF algorithm also gives the MMSE estimate of range and Doppler as described in Section 3.4.5. Figure 4.6 shows the range versus Doppler estimates together with the target's actual positions.

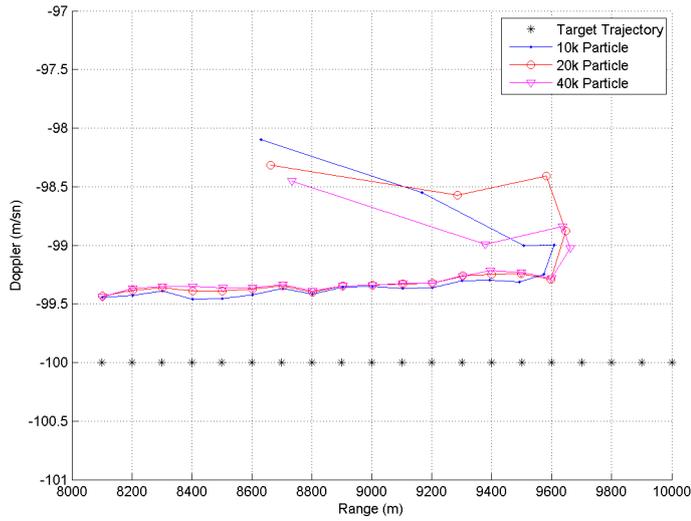


Figure 4.6: Range and Doppler Estimate (Swerling-0, 10dB Initial SNR)

Figure 4.6 shows that the particle filter is able estimate the range and the Doppler of the target more accurately after a few more steps then the detection is actually declared. Figure 4.7 shows the RMS range and Doppler tracking errors respectively.

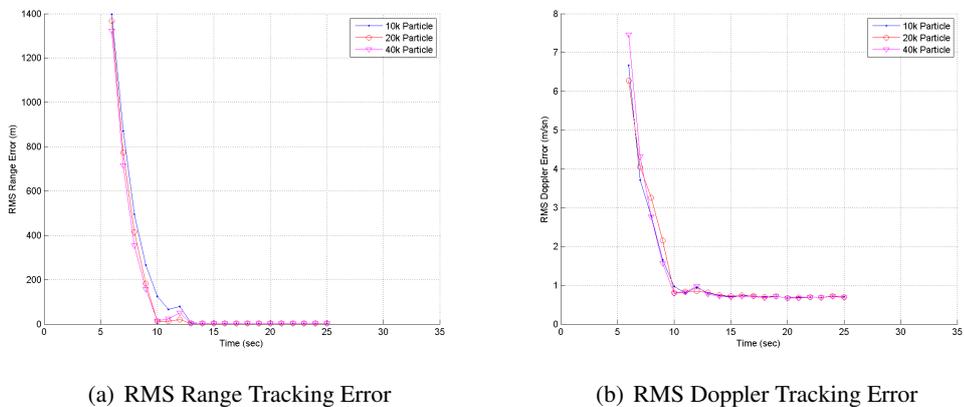


Figure 4.7: RMS Tracking Error (Swerling-0, 10dB Initial SNR)

Effective number of particles as explained in Section 2.3.2 gives an idea about the performance of the PF algorithm. It is expected that the larger effective number of particles yields better performance in terms of detection and estimation accuracies. Figure 4.8 shows the effective number of particles in percentage. It is shown in the figure that the effective number of

particles increases after the target appears in the surveillance region until they reach to a steady state value.

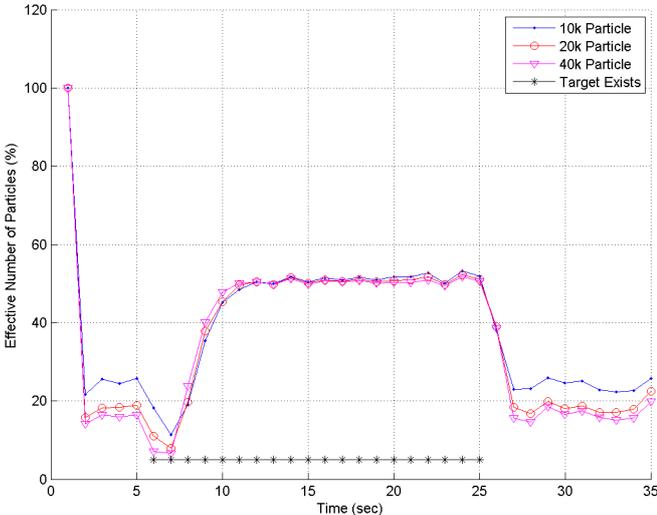


Figure 4.8: Effective Number of Particles (Swerling-0, 10dB Initial SNR)

Figure 4.9 shows the SNR estimate of the PF algorithm and the actual SNR of the target. It is shown that the algorithm is able to estimate the target SNR as soon as the target appears in the surveillance region.

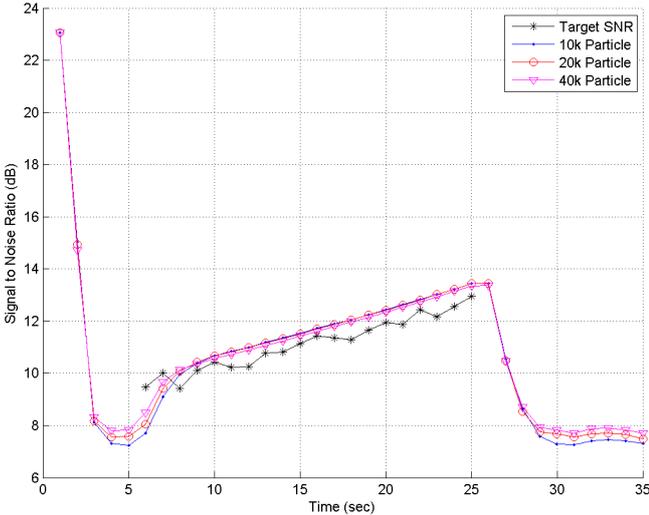


Figure 4.9: SNR Estimate (Swerling-0, 10dB Initial SNR)

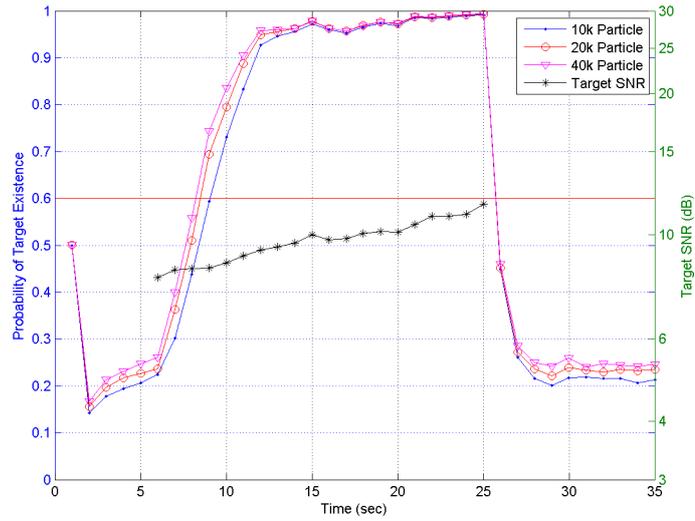


Figure 4.10: Probability of Target Existence and Target SNR (Swerling-0, 8dB Initial SNR)

### 8dB Initial Target SNR

The second experiment is performed for a constant amplitude target with the initial SNR of 8dB. Figure 4.10 shows the probability of target existence together with target SNR. It is shown in Figure 4.10 that the algorithm has declared the target existences between 4 to 5 seconds after the target is appeared in the surveillance region. For this case, as the number of particles are increased less detection times are required.

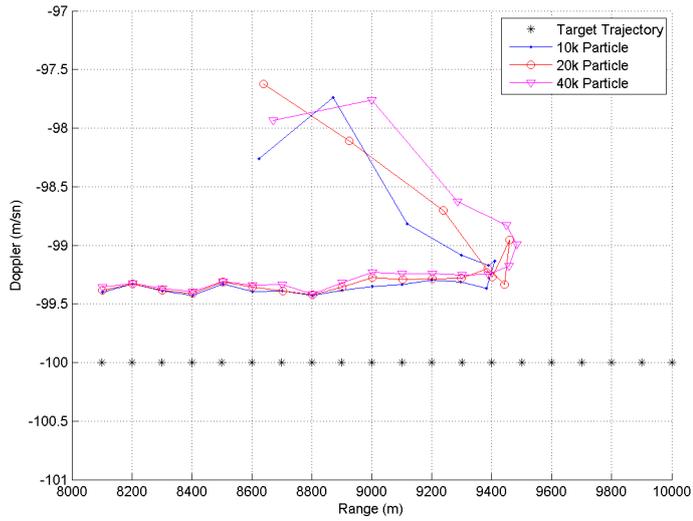
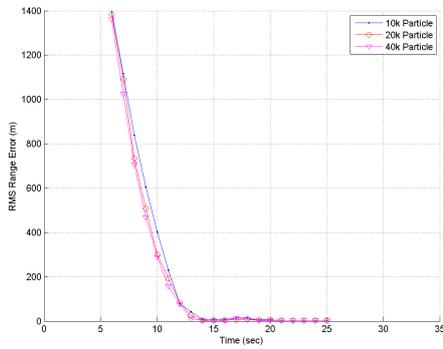
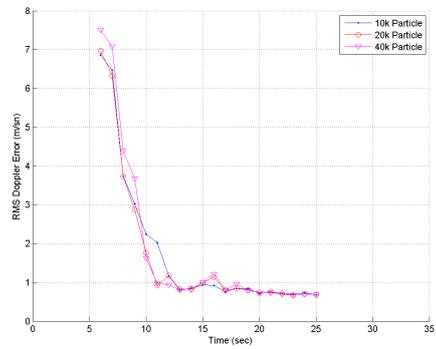


Figure 4.11: Range and Doppler Estimate (Swirling-0, 8dB Initial SNR)

The range and Doppler estimate of the algorithm is shown in Figure 4.11 and the corresponding RMS error plots are given in Figure 4.12



(a) RMS Range Tracking Error



(b) RMS Doppler Tracking Error

Figure 4.12: RMS Tracking Error (Swirling-0, 8dB Initial SNR)

The SNR estimate of the target is given in Figure 4.13.

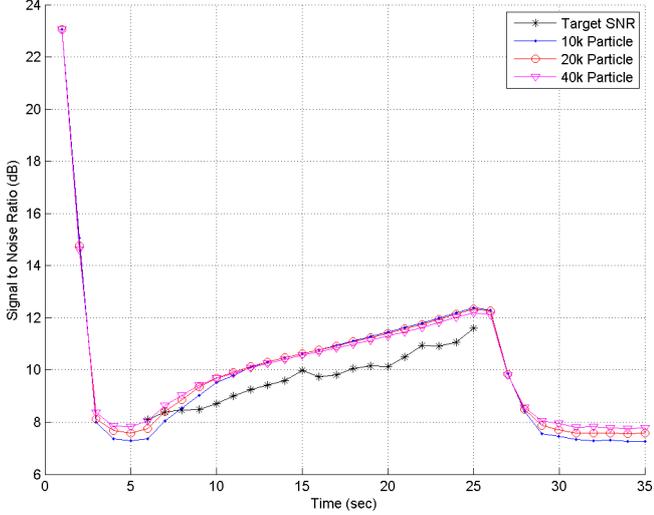


Figure 4.13: SNR Estimate (Swerling-0, 8dB Initial SNR)

Figures 4.12 and 4.13 show that the range, Doppler and SNR estimation accuracies are degraded comparing to the 10dB SNR case.

Finally number of effective particles are given in Figure 4.14

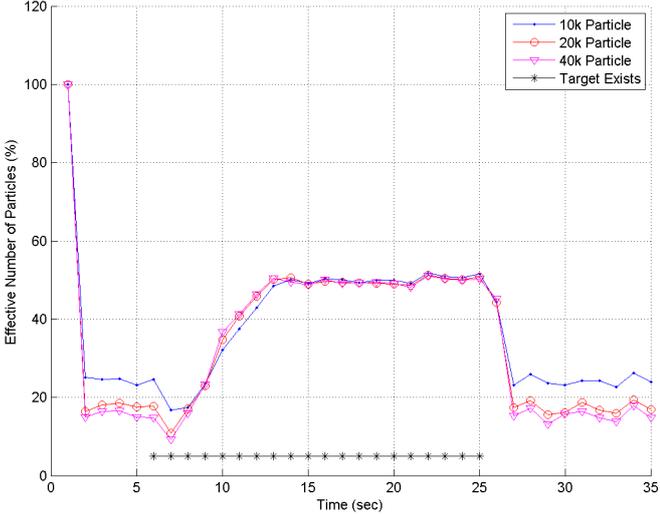


Figure 4.14: Effective Number of Particles (Swerling-0, 8dB Initial SNR)

## 6dB Initial Target SNR

The initial target SNR is then reduced to 6dB. Previous cases show that the performance degrades as the SNR decreases. This analysis also shows that the trend continues for 6dB case however the target declaration is made eventually. Figure 4.15 shows the probability of target existence together with the target SNR. It is shown in the figure that for 6dB initial target SNR, TBD algorithm performs target declaration between 9 to 11 time steps depending on the number of particles. In this case, for low number of particles i.e. 10k, the SNR of the target has increased to 7.5dB until the target is declared. However for higher number of particles i.e. 40k target is declared at the SNR values as low as 6.7dB.

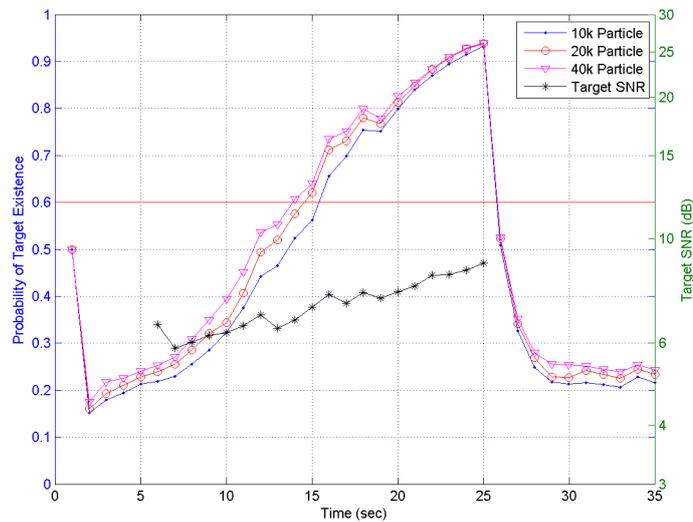


Figure 4.15: Probability of Target Existence and Target SNR (Swerling-0, 6dB Initial SNR)

The range Doppler estimate of the algorithm is shown in Figure 4.16 and the corresponding RMS error plots are given in Figure 4.17. Figure 4.17 shows that the tracking accuracy of the TBD algorithm is degraded with initial target SNR of 6dB as expected.

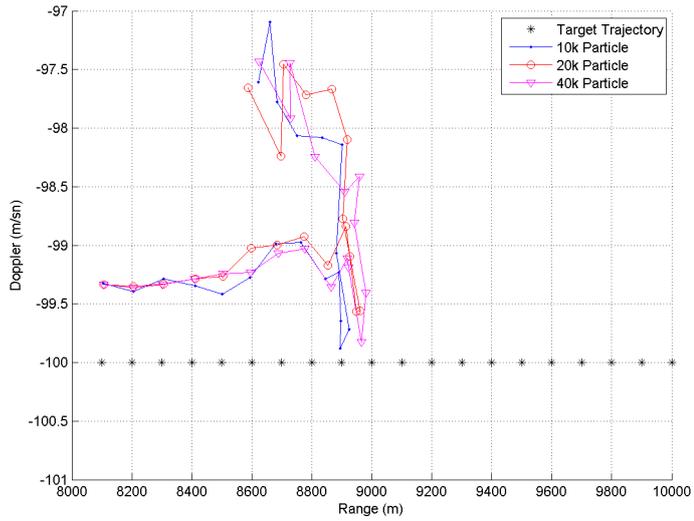
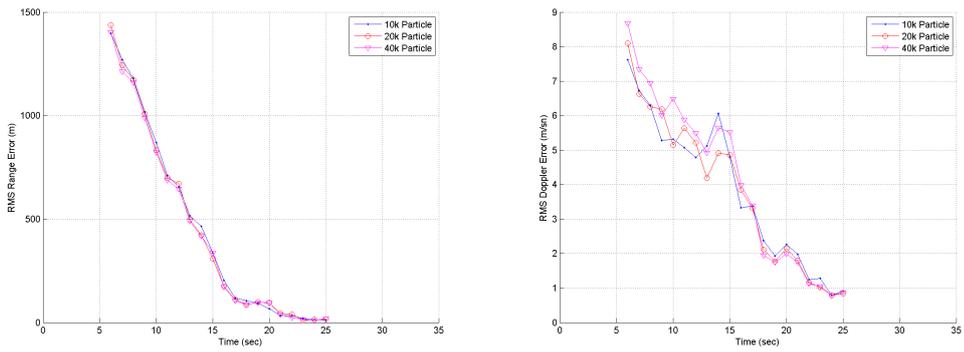


Figure 4.16: Range and Doppler Estimate (Swerling-0, 6dB Initial SNR)



(a) RMS Range Tracking Error

(b) RMS Doppler Tracking Error

Figure 4.17: RMS Tracking Error (Swerling-0, 6dB Initial SNR)

The SNR estimate performance of the algorithm is also degraded with the reduced SNR as shown in Figure 4.18.

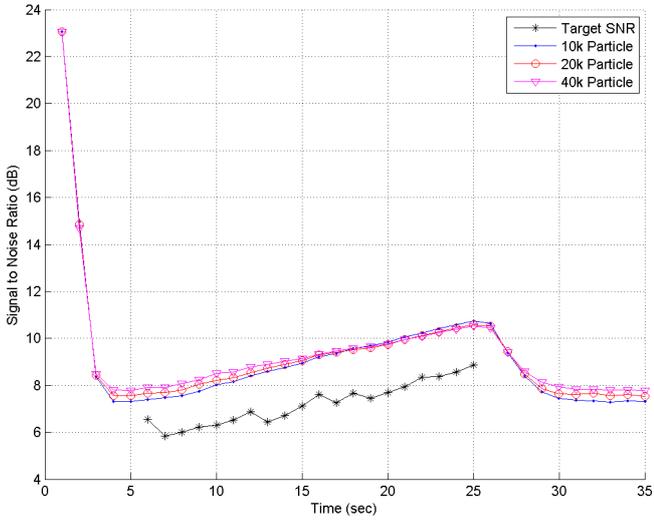


Figure 4.18: SNR Estimate (Swerling-0, 6dB Initial SNR)

Finally effective number of particles are given in Figure 4.19. It can be seen in the figure that the number of effective particles remains in the transient state longer than the higher SNR values. This is a corollary of the reduced SNR.

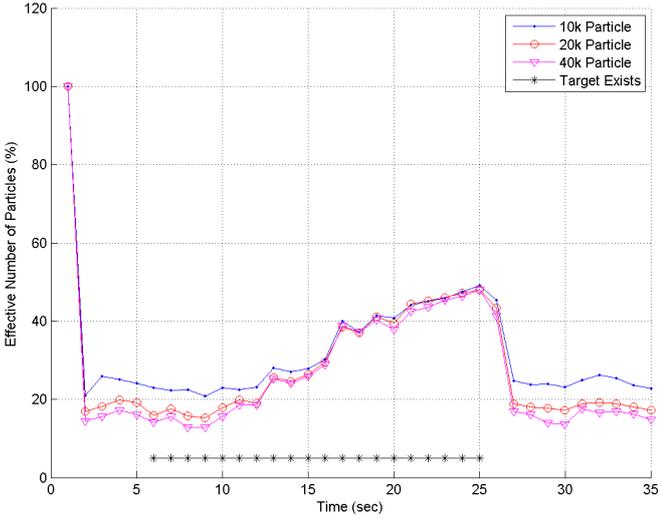


Figure 4.19: Effective Number of Particles (Swerling-0, 6dB Initial SNR)

Table 4.2: Declaration Times for Different SNR's (Swerling-0)

Number of Particles / Initial SNR	15dB	10dB	8dB	6dB
10k Particles	2 (15.2dB)	3 (9.5dB)	5 (8.7dB)	11 (7.6dB)
20k Particles	2 (15.2dB)	3 (9.4dB)	4 (8.5dB)	10 (7.1dB)
40k Particles	2 (15.2dB)	3 (9.4dB)	4 (8.5dB)	9 (6.7dB)

#### 4dB Initial Target SNR

As the final experiment related with this scenario the initial target SNR is reduced to 4dB. This particular case is out of the surveillance region in terms of signal power, see Table 4.1. However, analysis is done in order to test the TBD algorithm performance.

From Figure 4.20, it can be seen that the TBD algorithm fails to detect the target at 4dB SNR.

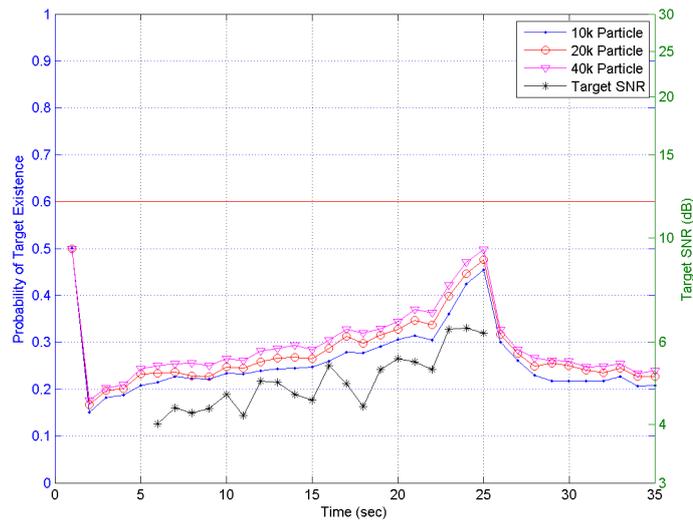


Figure 4.20: Probability of Target Existence and Target SNR (Swerling-0, 4dB Initial SNR)

#### Summary

In the first set of experiments, an incoming constant velocity target with a radial velocity of 100m/sec is simulated under constant amplitude (Swerling-0) assumption. Table 4.2 shows the target declaration times for different target initial SNR's and number of particles.

It is observed that the proposed TBD algorithm is successful in terms of target declaration for

SNR's as low as 6 dB although it takes more time to achieve the objective as the SNR decreases. It is also shown that the declaration times are reduced by increasing the number of particles especially for very low SNR values, see Table 4.2. In most of the radar detection problems 10dB to 6dB SNR values are considered as very low. See [18] for detection performance vs target SNR for a non-fluctuating target.

Another conclusion is that as the target SNR decreases both detection and estimation performances of the TBD algorithm reduces as expected. Notice that for all of the results the target SNR increases with the reduced range according to the radar range equation. Table 4.3 shows the target SNR's when it is declared by the TBD algorithm. It is shown that around 7dB SNR is sufficient to detect a Swerling-0 target.

Note that the proposed algorithm has a Swerling-1 measurement model. This analysis also shows that this model handles the Swerling-0 target case successfully.

### **4.3.2 Incoming Target with Fluctuating Amplitude**

In order to evaluate the performance of the TBD algorithm under fluctuating target amplitude assumption an incoming target scenario is used again as shown Figure 4.1. It is assumed that the target return power fluctuates according to Swerling-1 model. Simulation results are given below for different initial target mean SNR's and different number of particles. Note that during the scenario as the target becomes closer to the radar, the mean SNR increases according to the radar range equation defined in Section 2.1.2. Also note that all of the results are obtained by 100 Monte Carlo simulations. During the analysis the target appears at  $t = 6sec$  and disappears at  $t = 26sec$ .

#### **10dB Initial Target SNR**

Since the target return signal fluctuates, first the histogram of the power of the return signal is given in Figure 4.21 at time  $t = 6$  for 100 Monte Carlo trials. Figure 4.21 shows that the exponential distribution obtained from the measurements fit well into the observed data. The mean value is also shown with a black line which is equal to 9.92dB. Thus, Figure 4.21 justifies the measurement model defined earlier as

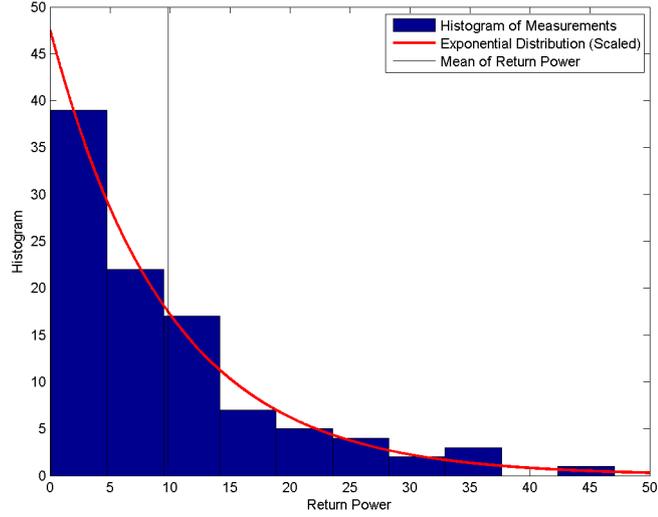


Figure 4.21: Demonstration of Target Plus Noise Signal

$$p_{S+N}(z_k|x_k) \sim \begin{cases} \frac{1}{P_r} e^{-x/P_r} & ; x \geq 0 \\ 0 & ; x \leq 0 \end{cases} \quad (4.13)$$

Figure 4.22 shows the probability of target existence and target SNR with respect to time. On the left axis probability of target existence is shown for different number of particles. On the right axis target mean SNR is shown for 100 Monte Carlo trials.

Note that the SNR is shown only between the times where the target presents in the surveillance region. Also note that the displayed SNR values are the mean SNR taken through 100 Monte Carlo trials.

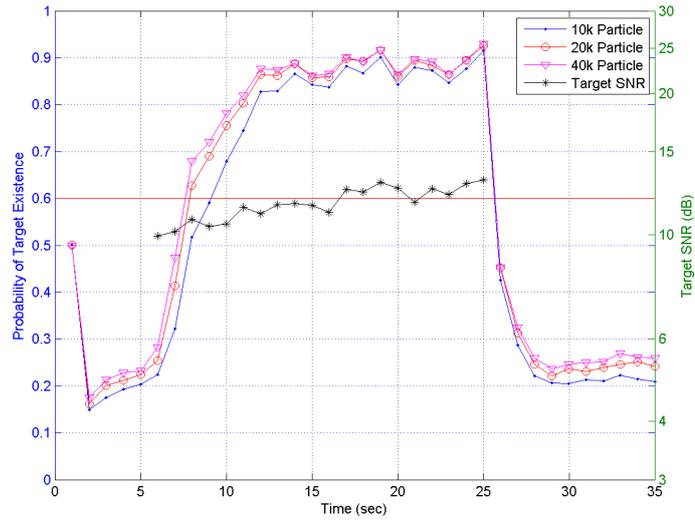


Figure 4.22: Probability of Target Existence and Target SNR (Swirling-1, 10dB Initial SNR)

From Figure 4.22, it is seen that the target existence is declared within 3–5 seconds depending on the number of particles.

The range Doppler estimate of PF algorithm is shown in Figure 4.23 and the corresponding RMS error plots are given in Figure 4.24.

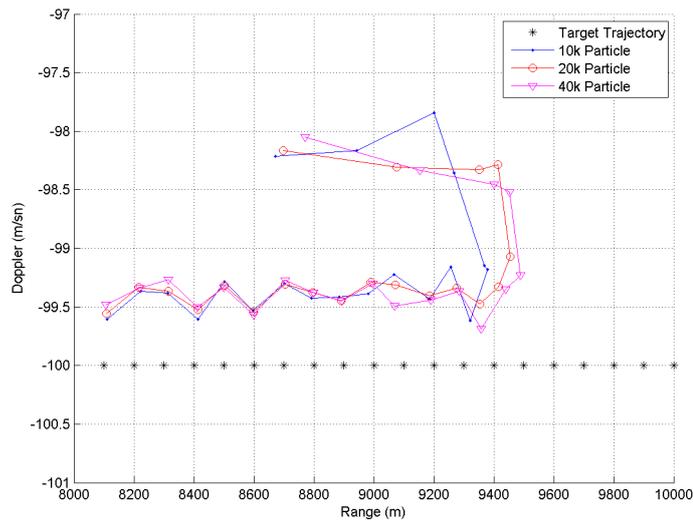
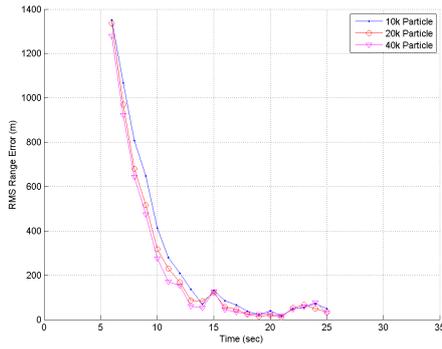
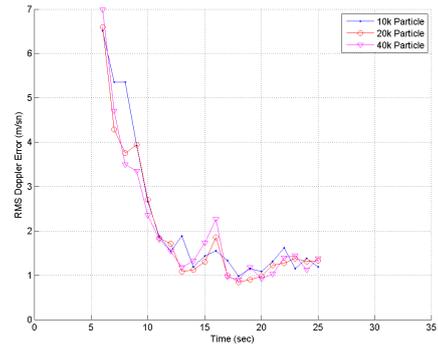


Figure 4.23: Range and Doppler Estimate (Swirling-1, 10dB Initial SNR)



(a) RMS Range Tracking Error



(b) RMS Doppler Tracking Error

Figure 4.24: RMS Tracking Error (Swerling-1, 10dB Initial SNR)

The SNR estimate of the target is given in Figure 4.25.

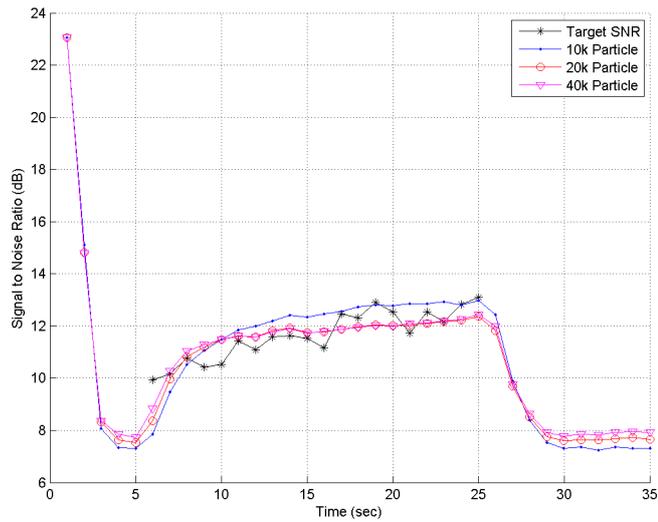


Figure 4.25: SNR Estimate (Swerling-1, 10dB Initial SNR)

Figures 4.23, 4.24 and 4.25 show that the TBD algorithm estimates the range, Doppler and SNR of the target successfully even with 10k particles although the accuracies slightly increase with increased number of particles.

Finally effective number of particles are given in percentage in Figure 4.26.

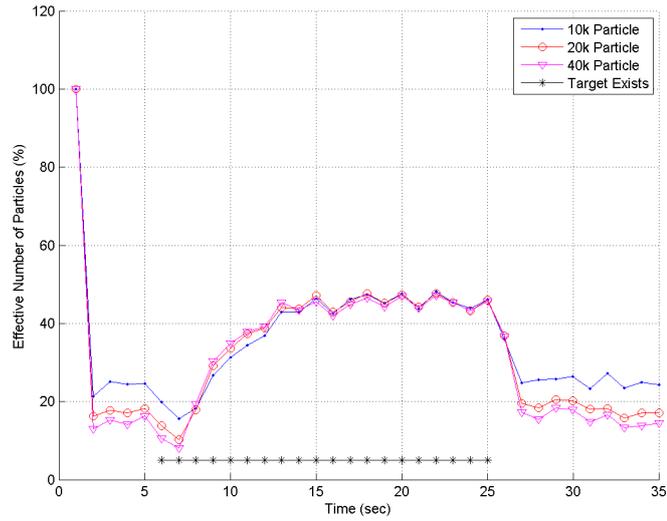


Figure 4.26: Effective Number of Particles (Swerling-1, 10dB Initial SNR)

### 8dB Initial Target SNR

In the second experiment the initial target SNR is reduced to 8dB. The probability of target existence is shown in Figure 4.27. For this case the target existence is declared between 6 – 7 seconds. Again, it is observed that the increased number of particles improves the performance of the algorithm in terms of declaration times.

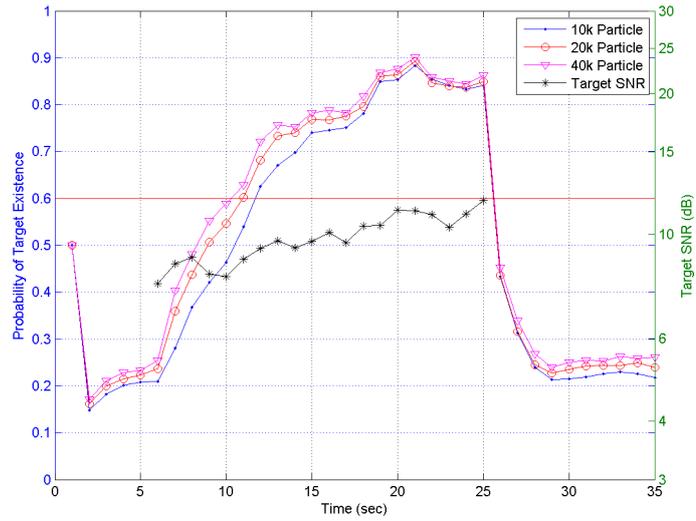


Figure 4.27: Probability of Target Existence and Target SNR (Swerling-1, 8dB Initial SNR)

The range Doppler estimate and the corresponding RMS error plots are given in Figures 4.28 and 4.29 respectively.

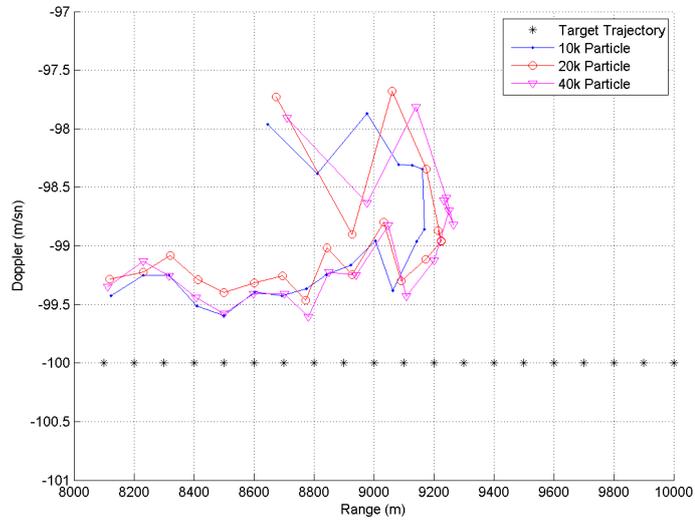
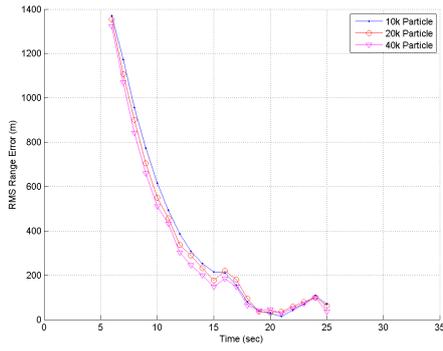
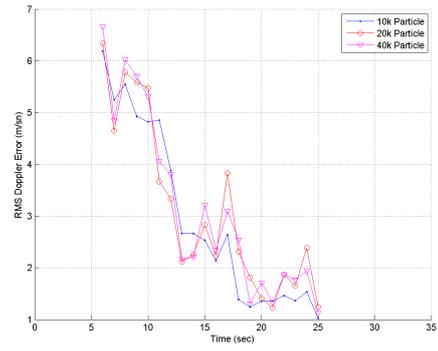


Figure 4.28: Range and Doppler Estimate (Swerling-1, 8dB Initial SNR)



(a) RMS Range Tracking Error



(b) RMS Doppler Tracking Error

Figure 4.29: RMS Tracking Error (Swerling-1, 8dB Initial SNR)

The SNR estimate of target is given in Figure 4.30.

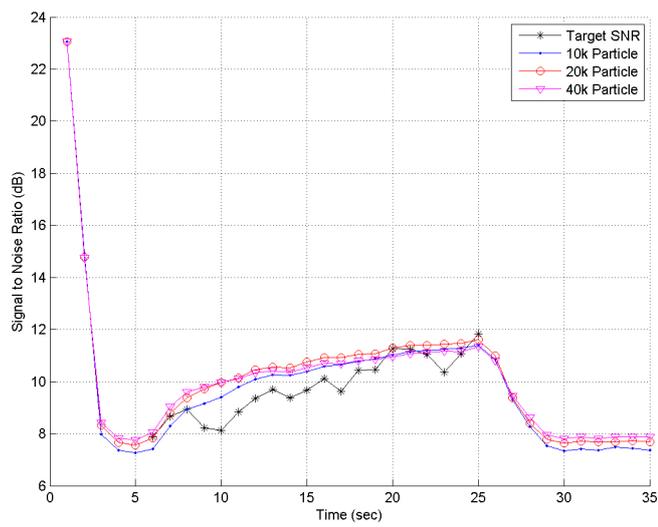


Figure 4.30: SNR Estimate (Swerling-1, 8dB Initial SNR)

Finally Figure 4.31 shows the effective number of particles.

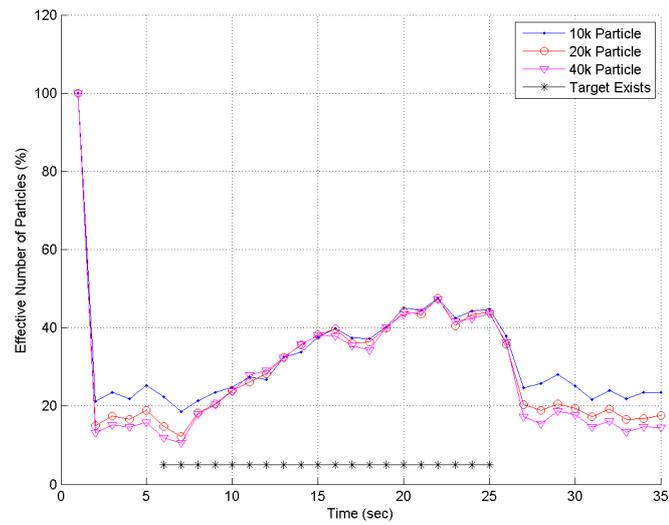


Figure 4.31: Effective Number of Particles (Swerling-1, 8dB Initial SNR)

### 6dB Initial Target SNR

The initial target SNR is then reduced to 6dB for the same trajectory. The probability of target existence obtained in this case is shown in Figure 4.32

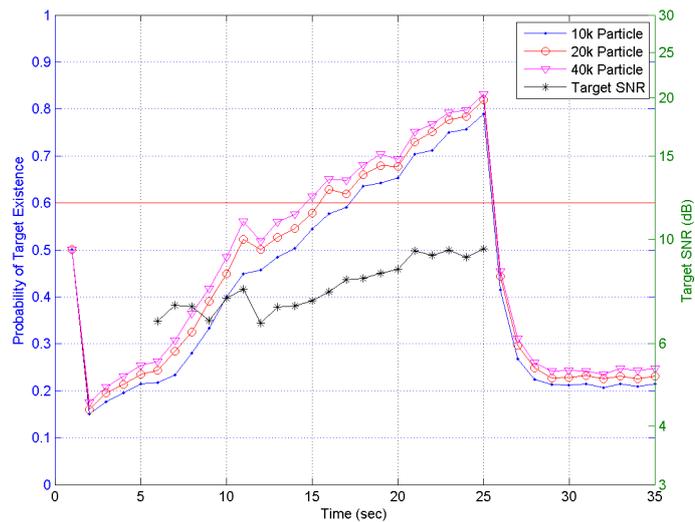


Figure 4.32: Probability of Target Existence and Target SNR (Swerling-1, 6dB Initial SNR)

In Figure 4.32, it is shown that the proposed TBD algorithm is able to detect Swerling-1 target even under 6dB initial target SNR. The target declaration times are obtained between 10 to 14 seconds depending on the number of particles although the SNR of the target increases to about 7.5dB to 8dB at that time.

The range and Doppler estimate of PF algorithm is shown in Figure 4.33 and the corresponding RMS error plots are given in Figure 4.34.

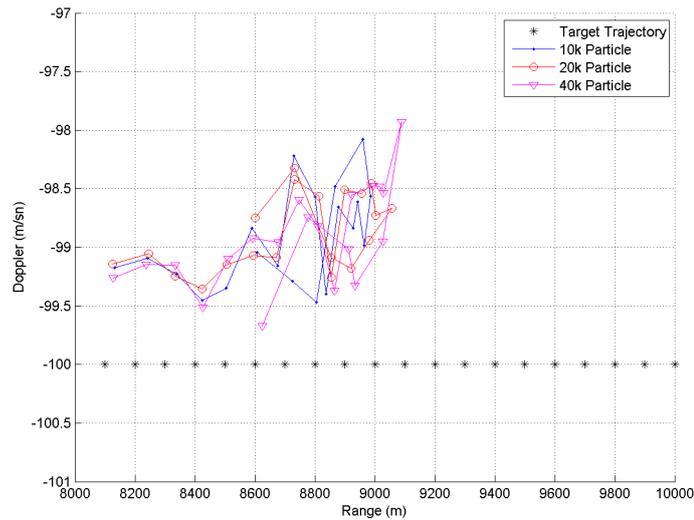
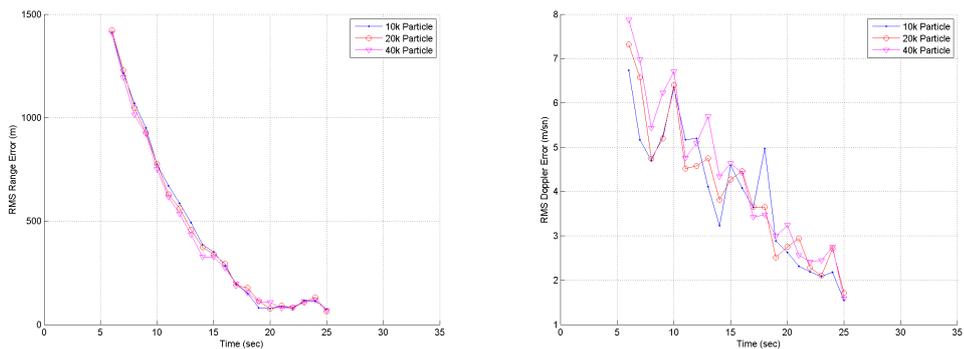


Figure 4.33: Range and Doppler Estimate (Swerling-1, 6dB Initial SNR)



(a) RMS Range Tracking Error

(b) RMS Doppler Tracking Error

Figure 4.34: RMS Tracking Error (Swerling-1, 6dB Initial SNR)

The SNR estimate of target is given in Figure 4.35.

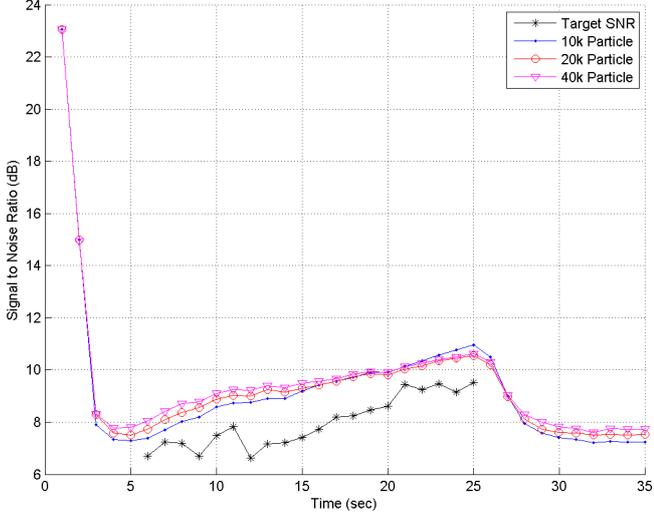


Figure 4.35: SNR Estimate (Swerling-1, 6dB Initial SNR)

Note that although the target declaration is performed eventually the estimation accuracy is not very satisfactory when the target SNR is 6dB.

The effective number of particles are given in Figure 4.36.

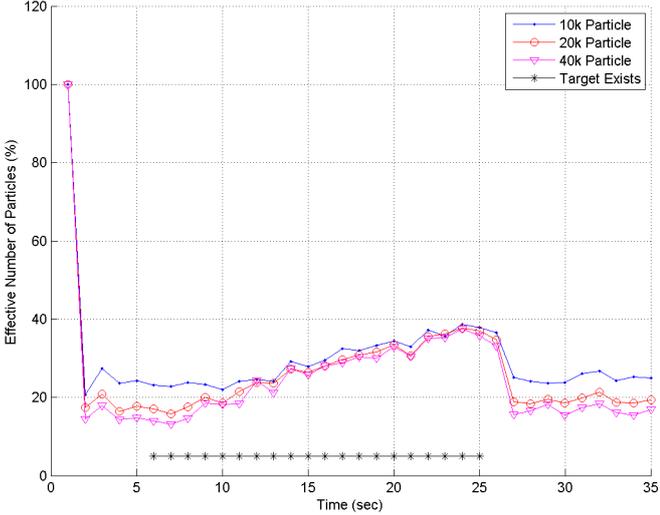


Figure 4.36: Effective Number of Particles (Swerling-1, 6dB Initial SNR)

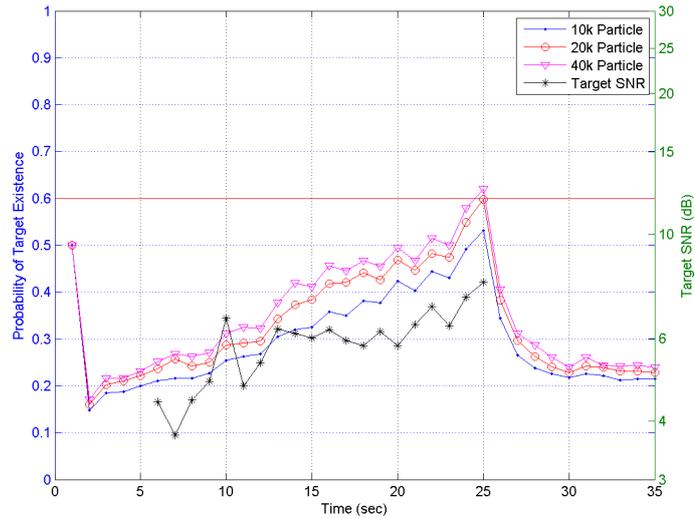


Figure 4.37: Probability of Target Existence and Target SNR (Swerling-1, 4dB Initial SNR)

#### 4dB Initial Target SNR

The target SNR is then reduced to 4dB when the target is appeared in the surveillance region. From Figure 4.37, it can be seen that the TBD algorithm fails to detect the target at 4dB SNR for 10k and 20k particle cases. Although it manages to detect the target with 40k particles at the end of 20 seconds, target SNR becomes 8dB at that time.

#### Summary

In the second part of the experiments the same scenario is used as an incoming constant velocity target with a radial velocity of 100m/sec. However, this time it is assumed that the target power fluctuates according to Swerling-1 mode. It is observed that for relatively high SNR values number of particles does not play a crucial role on the performance of the algorithm. However, for low SNR values like 6dB and 8dB, effects of the number of particles are seen directly. Table 4.3 shows the target declaration times for different number of particles and target SNR's. For most of the radar detection and track initiation problems 10dB to 6dB SNR values are considered as very low for a Swerling-1 target, [18]

Table 4.3: Declaration Times Observed for Different SNR's (Swerling-1)

Number of Particles / Initial SNR	15dB	10dB	8dB	6dB
10k Particles	2 (15.5dB)	5 (10.5dB)	7 (9.3dB)	13 (8.2dB)
20k Particles	2 (15.5dB)	3 (10.8dB)	6 (8.8dB)	11 (7.7dB)
40k Particles	2 (15.5dB)	3 (10.8dB)	6 (8.8dB)	10 (7.4dB)

It is observed that both the detection and the estimation performance of the algorithm degrades with low SNR values as expected. Although it takes more time to achieve the target detection, TBD algorithm manages to work until 6dB SNR of target appearance at the beginning of the trajectory. Since the target is approaching to the radar it's SNR is increasing so that the SNR value at the time of declaration is different from the initial value. Table 4.3 shows the target SNR's when it is declared by the TBD algorithm and from the table mean SNR around 8dB is necessary to detect a target with the proposed algorithm.

### 4.3.3 Maneuvering Target with Fluctuating Amplitude

To get more insight about the performance of the proposed TBD algorithm we conducted similar experiments on a maneuvering target. Target trajectory is given in Figure 4.2. Detailed information about the trajectory is given in Section 4.2.1. For this scenario target exists in the surveillance region in the time interval of [6 – 25] seconds. In between 6 – 11 seconds it makes a maneuver of +1g and in between 11 – 17 seconds it makes the second maneuver of -1g. From 17sec to the end of the trajectory target has a constant velocity motion.

Simulation results are given below for different initial target SNR's and different number of particles. Note that all of the results are obtained by 100 Monte Carlo simulations.

#### 10dB Initial Target SNR

Figure 4.39 shows the probability of target existence for different number of particles together with the target SNR. Note that, the target still appears at time  $t = 6sec$  and disappears at time  $t = 20sec$ . Also note that the displayed SNR values are the mean SNR obtained by 100 Monte Carlo trials.

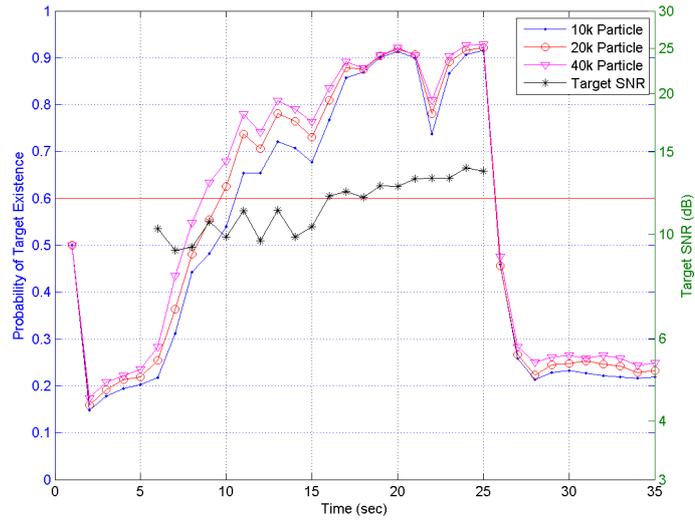


Figure 4.38: Probability of Target Existence and Target SNR (Swerling-1, 10dB Initial SNR)

Figure 4.39 shows that the proposed TBD algorithm detects a 10dB SNR of Swerling-1 target at 4 to 6 time steps during maneuver depending on the number of particles.

The range Doppler estimate and the corresponding RMS error plots are given in Figures 4.39 4.40 respectively.

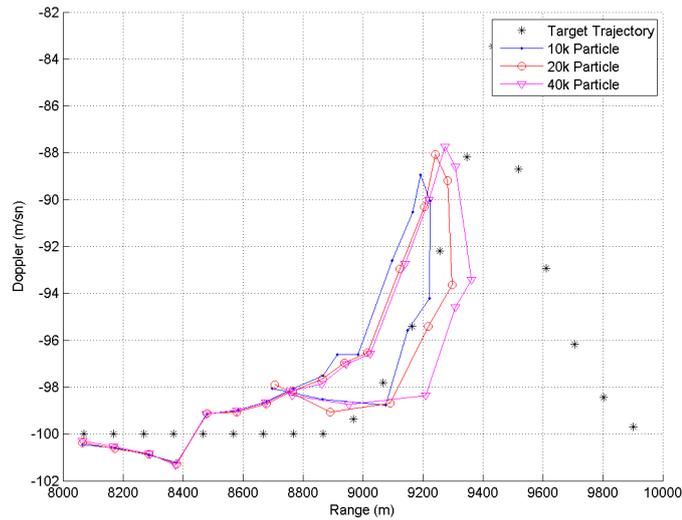
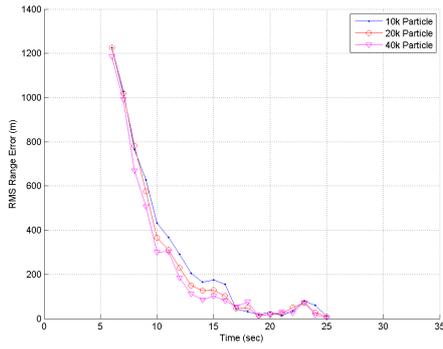
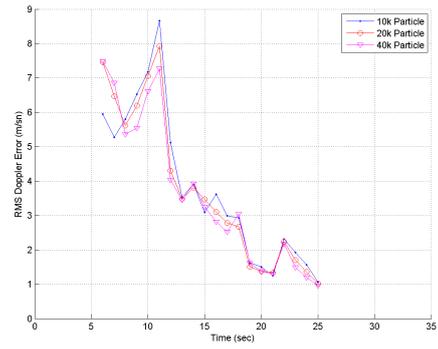


Figure 4.39: Range and Doppler Estimate (Swerling-1, 10dB Initial SNR)



(a) RMS Range Tracking Error



(b) RMS Doppler Tracking Error

Figure 4.40: RMS Tracking Error (Swerling-1, 10dB Initial SNR)

The SNR estimate of target is given in Figure 4.41.

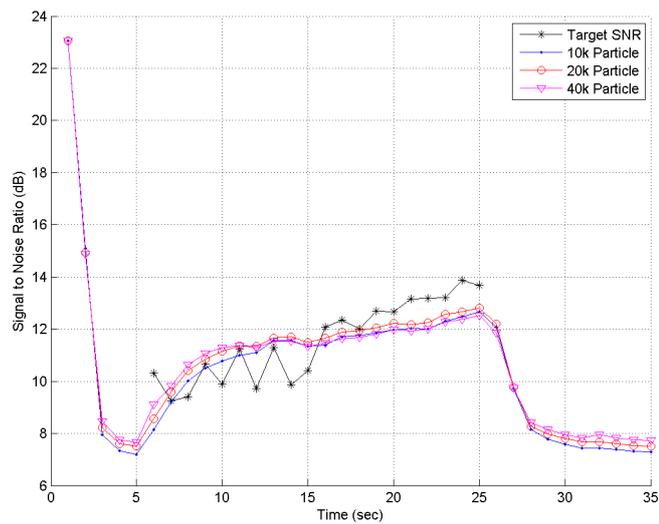


Figure 4.41: SNR Estimate (Swerling-1, 10dB Initial SNR)

Finally effective number of particles are given in Figure 4.42.

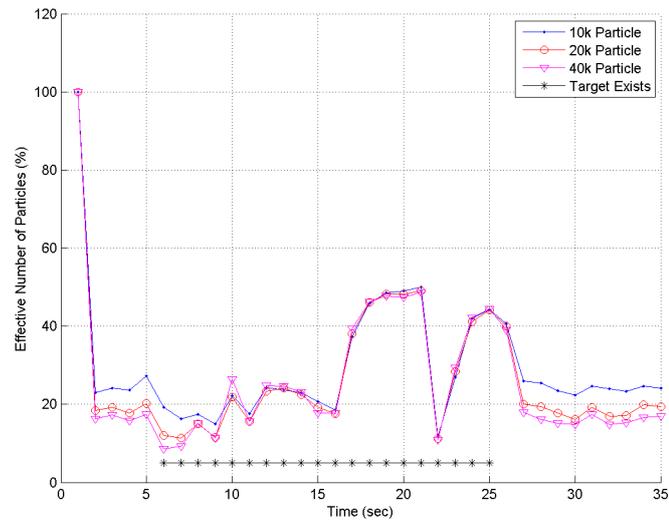


Figure 4.42: Effective Number of Particles (Swerling-1, 10dB Initial SNR)

### 8dB Initial Target SNR

The experiments are repeated for 8dB initial target SNR. The results are given in Figures 4.43 to 4.47.

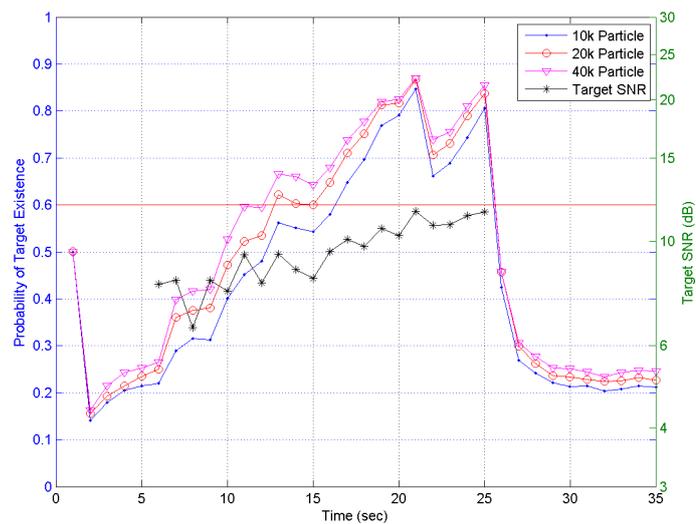


Figure 4.43: Probability of Target Existence and Target SNR (Swerling-1, 8dB Initial SNR)

Figure 4.43 shows that the proposed TBD algorithm detects an 8dB SNR of Swerling-1 target at 8 to 12 seconds during maneuver depending on the number of particles.

The range Doppler estimate of PF algorithm is shown in Figure 4.44 and the corresponding RMS error plots are given in Figure 4.45.

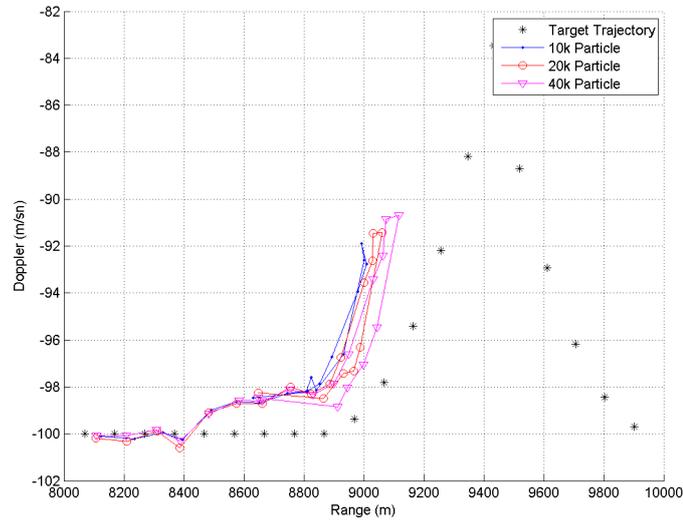


Figure 4.44: Range and Doppler Estimate (Swerling-1, 8dB Initial SNR)

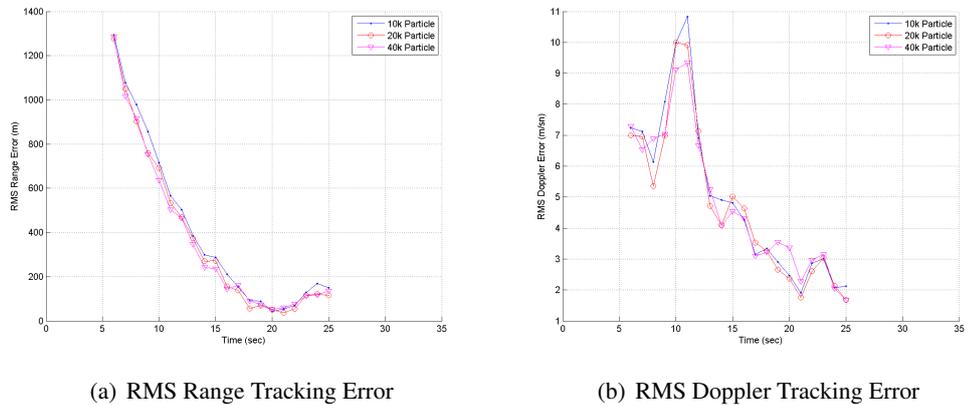


Figure 4.45: RMS Tracking Error (Swerling-1, 8dB Initial SNR)

The SNR estimate of target is given in Figure 4.46.

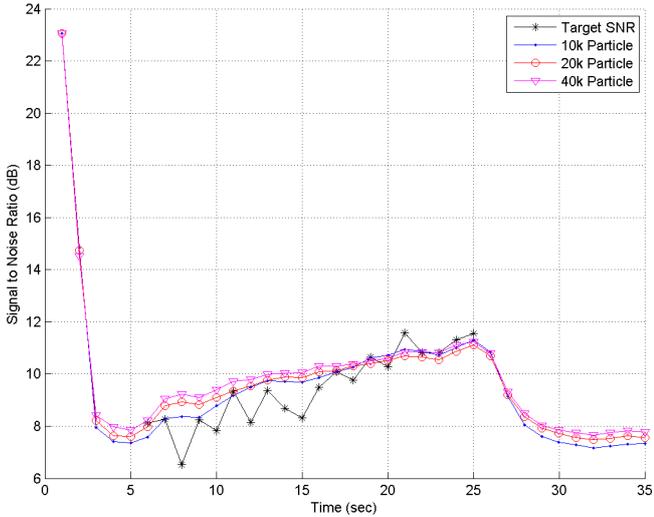


Figure 4.46: SNR Estimate (Swerling-1, 8dB Initial SNR)

Finally effective number of particles are given in Figure 4.47.

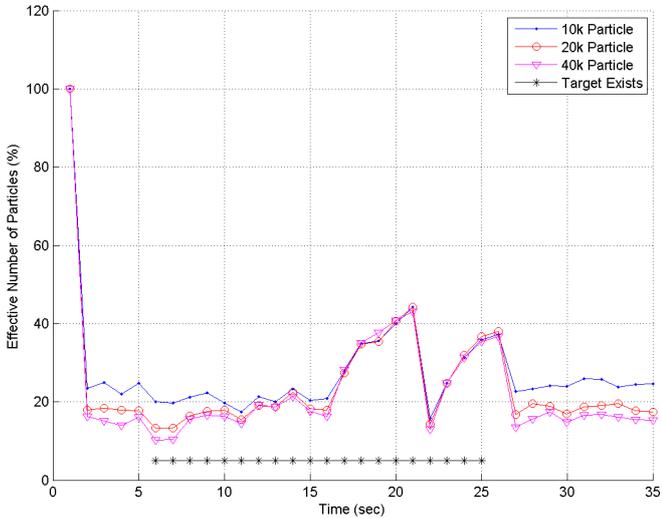


Figure 4.47: Effective Number of Particles (Swerling-1, 8dB Initial SNR)

### 6dB Initial Target SNR

The initial target SNR is then reduced to 6 dB. The probability of target existence for a 6dB mean SNR target is shown in Figure 4.48.

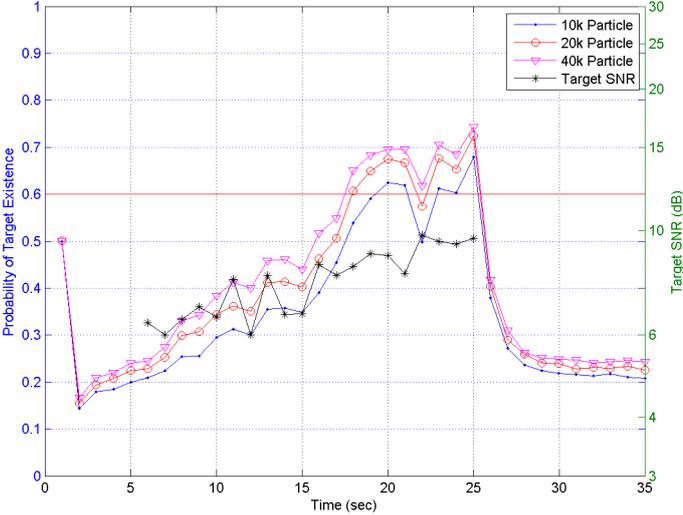


Figure 4.48: Probability of Target Existence and Target SNR (Swerling-1, 6dB Initial SNR)

From Figure 4.48, it can be seen that the TBD algorithm detects the target in between 13 to 15 seconds after its appearance. Although TBD algorithm manages to detect the target, at the time of detection its SNR is 8dB.

### 4dB Initial Target SNR

The probability of target existence for a 4dB mean SNR target is shown in Figure 4.49. It is shown in the figure that the TBD algorithm fails to detect the target appearing at 4dB SNR.

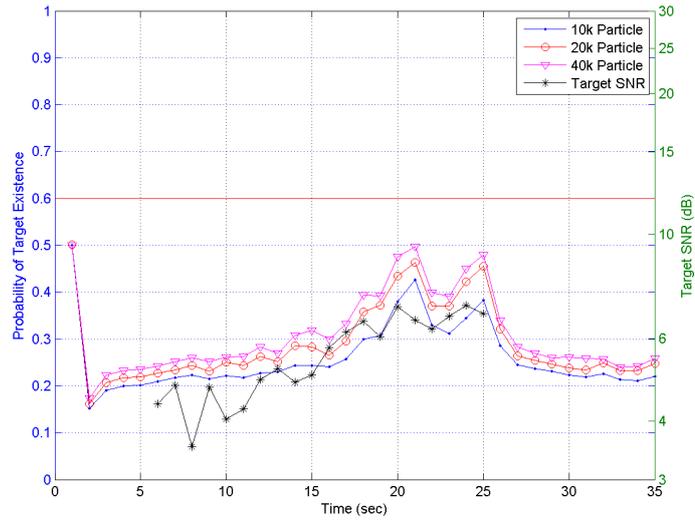


Figure 4.49: Probability of Target Existence and Target SNR (Swerling-1, 4dB Initial SNR)

## Summary

In this case a maneuvering target with an initial radial velocity of 100m/sec is used under fluctuating amplitude (Swerling-1) assumption. It is observed that the number of particles play a crucial role on the performance of the algorithm. Table 4.4 shows the target declaration times obtained in this analysis.

Table 4.4: Declaration Times Observed for Different SNR's (Swerling-1)

Number of Particles / Initial SNR	15dB	10dB	8dB	6dB
10k Particles	3 (15.2dB)	6 (11.2dB)	12 (10.1dB)	15 (8.8dB)
20k Particles	2 (15.2dB)	5 (10.0dB)	8 (9.4dB)	13 (8.4dB)
40k Particles	2 (15.2dB)	4 (10.6dB)	8 (9.4dB)	13 (8.4dB)

As it would be expected both detection and estimation performance of the algorithm degrades as the SNR decreases. Although it takes more time to achieve the target detection, TBD algorithm manages to work until 6dB initial target SNR's. Table 4.4 also shows the target SNR's when it is declared by the TBD algorithm.

#### 4.3.4 Noise Only Detection

The final experiment performed in this section is the simulation of a noise only scenario to analyze the false track initiation performance of the proposed algorithm. For this reason 500 steps of noise data is simulated and used within the algorithm. The target existence probability,  $P_{E_k}$  versus time is shown in Figure 4.50.

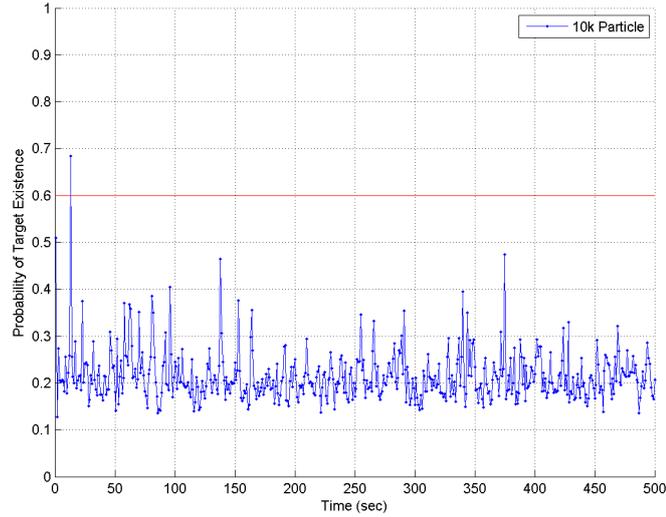


Figure 4.50: Target Existence Probability vs Time

Assuming that the target is declared once the  $P_{E_k}$  exceeds the given threshold  $\lambda_d = 0.6$ , it is shown that only 1 false track is initiated during the scenario. Note that the false track generated at frame 13, is immediately deleted by the algorithm. Let us define the 'Probability of False Track Initiation' as

$$P_{FT} = \frac{\text{Total Number of False Tracks}}{\text{Resolution Cells Tested}} \quad (4.14)$$

By using Equation (4.14) and the simulation parameters,  $P_{FT}$  can be calculated as

$$P_{FT} = \frac{1}{500 * 160 * 14} = 8.92 * 10^{-7} \quad (4.15)$$

By this analysis, it is shown that the proposed TBD algorithm has an excellent false track

initiation probability.

### 4.3.5 Summary of the Results

In this chapter the proposed PF based TBD algorithm is tested in four different scenarios. It is observed that the algorithm performs outstanding performance under target SNR's as low as 6dB. Table 4.5 shows a summary of the results obtained by the algorithm for 40k particles.

Table 4.5: Declaration Times Observed

Scenario / Initial SNR	15dB	10dB	8dB	6dB
Swerling-0, Incoming Target	2	3	4	9
Swerling-1, Incoming Target	2	3	6	10
Swerling-1, Maneuvering Target	2	4	8	13

It is also shown that the false track probability of the algorithm is  $8.92 \cdot 10^{-7}$ . As denoted before for most of the radar detection problems 10dB to 6dB SNR values are considered as very low. However, track before detect approach performs detection with those SNR's by means of integrating particles for a number of time steps. This integration is based on the target state model which is used by the particle filter. Table 4.5 shows that using TBD method defined in this thesis, 6dB initial target SNR is obtained as the performance limit in terms of target declaration for the defined false track probability. However in terms of the target SNR when it is actually declared by the algorithm, the performance limit is just above 8dB for Swerling-1 cases and below 7dB for Swerling-0 cases.

Another important result is that for each case increased number of particles causes better detection and estimation performances for very low SNR values as 6dB to 8dB. However for SNR values as 10dB or higher, the number of particles does not play a crucial role on the performance of the algorithm if it is on the order of 10k. This is an important result in order to use the algorithm in practical situations and can be used to determine the number of particles to be used for a specific scenario.

## CHAPTER 5

### CONCLUSION

#### 5.1 Thesis Summary

A Particle Filter (PF) based Track Before Detect (TBD) algorithm is derived within this thesis. It has been shown that using the TBD approach the detection and the tracking performance of low SNR targets have been improved.

In this work, track before detect algorithm is developed in order to account for target amplitude fluctuations according to Swerling models. Furthermore, in contrast to many other works in literature, target mean SNR is not assumed to be known and it is estimated within the algorithm. The state model is selected as an augmented state that includes the target kinematics as well as its amplitude. The kinematic model is adjusted to cover the maneuvering target cases also.

The developed algorithm is tested under four different scenarios

1. Incoming Target with Constant Amplitude
2. Incoming Target with Fluctuating Amplitude
3. Maneuvering Target with Fluctuating Amplitude
4. Noise Only Detection

It is observed that the proposed PF based TBD algorithm has superior performance under very low SNR targets upto 6dB even with number of particles as low as 10k. Also it has been shown that after a few number of steps, PF algorithm detects the target. Moreover, it has been

shown that the proposed algorithm has an excellent false track probability demonstrated by the noise only scenario.

## **5.2 Future Work**

There may be several topics to be investigated using the concepts introduced in this thesis. These topics can be as follows

1. Multiple target cases can be embedded to the system model
2. Structured background can be embedded to the system model in order to account for undesired objects and/or clutter
3. A new measurement model can be developed in order to use the complex data instead of power measurements
4. Rao Blackwellised particle filters can be investigated in order to reduce computational demands and particle filter state dimensions

## REFERENCES

- [1] M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 2002.
- [2] Y. Bar-Shalom, P.K. Willet, and X. Tian. *Tracking and Data Fusion. A Handbook of Algorithms*. Yaakov Bar-Shalom, 2011.
- [3] S.S. Blackman. Multiple hypothesis tracking for multiple target tracking. *IEEE Aerospace and Electronic Systems Magazine*, 19(1), 2004.
- [4] Y. Boers and H. Driessen. Particle filter based track before detect algorithms. *Proceedings of SPIE - Signal and Data Processing of Small Targets*, 2003.
- [5] Y. Boers and J.N. Driessen. Multitarget particle filter track before detect application. *IEE Proceedings - Radar, Sonar and Navigation*, 2004.
- [6] B.D. Carlson, E.D. Evans, and S.L. Wilson. Search radar detection and track with the hough transform part i: System concept. *IEEE Transactions on Aerospace and Electronic Systems*, 1194.
- [7] S. Chib and E. Greenberg. Understanding the metropolis-hastings algorithm. *The American Statistician*, 1995.
- [8] S.J. Davey and M.G. Rutten. A comparison of three algorithms for tracking dim targets. In *Information, Decision and Control*, 2007.
- [9] A. Doucet, S.Godsill, and C. Andrieu. *On sequential Monte Carlo sampling methods for Bayesian filtering*. Kluwer Academic, 2000.
- [10] N.J. Gordon, D.J.Salmond, and A.F.M. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEE Proceedings-F*, 140(2), 1993.
- [11] F. Gustafsson. Particle filter theory and practice with positioning applications. *IEEE Aerospace and Electronic Systems Magazine*, 25(7), 2010.
- [12] A.H. Jazwinski. Stochastic processes and filtering theory. *IEEE Transactions on Automatic Control*, 1972.
- [13] L.A. Johnston and V. Krishnamurthy. Performance analysis of a dynamic programming track before detect algorithm. *IEEE Transactions on Aerospace and Electronic Systems*, 2002.
- [14] S.J. Julier and J.K. Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3), 2004.
- [15] S. Koç. Aselsan - odtü çalışmaları radar simülatörü. Technical report, ASELSAN A.S., 2005.

- [16] X.R. Li and V.P. Jilkov. Survey of maneuvering target tracking. part i: Dynamic models. *IEEE Transactions on Aerospace and Electronic Systems*, 2003.
- [17] X.R. Li and V.P. Jilkov. Survey of maneuvering target tracking. part v: Multiple-model methods. *IEEE Transactions on Aerospace and Electronic Systems*, 41(4), 2005.
- [18] Mark A. Richards. *Fundamentals of Radar Signal Processing*. McGraw-Hill, 2005.
- [19] B. Ristic, S. Arulampalam, and N. Gordon. *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House, 2004.
- [20] D.J. Salmond and H. Birch. A particle filter for track-before-detect. In *Proceedings of the American Control Conference*, 2001.
- [21] M.I. Skolnik. *Introduction to Radar Systems*. McGraw Hill, 2001.
- [22] M.E. Smith and P.K. Varshney. Intelligent cfar processor based on data variability. *IEEE Transactions on Aerospace and Electronic Systems*, 36(3), 2000.
- [23] P. Swerling. Radar probability of detection for some additional fluctuating target cases. *IEEE Transactions on Aerospace and Electronic Systems*, 1997.
- [24] S.M. Tonissen and Y. Bar-Shalom. Maximum likelihood track-before-detect with fluctuating target amplitude. *IEEE Transactions on Aerospace and Electronic Systems*, 1998.
- [25] S.M. Tonissen and R.J. Evans. Performance of dynamic programming techniques for track-before-detect. *IEEE Transactions on Aerospace and Electronic Systems*, 1996.
- [26] M.J. Walsh, M.L. Graham, R.L. Streit, T.E. Luginbuhl, and L.A. Mathews. Tracking on intensity-modulated sensor data streams. Technical report, Naval Undersea Warfare Center Division, 2000.