A COMPARATIVE STUDY OF TRACKING ALGORITHMS IN UNDERWATER ENVIRONMENT USING SONAR SIMULATION

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

BY

EMRE EGE

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

SEPTEMBER 2007

Approval of the thesis:

A COMPARATIVE STUDY OF TRACKING ALGORITHMS IN UNDERWATER ENVIRONMENT USING SONAR SIMULATION

Submitted by **EMRE EGE** 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**

Asst. Prof. Dr. Afşar Saranlı _____
Supervisor, **Electrical and Electronics Engineering Dept., METU**

Examining Committee Members:

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

Prof. Dr. M. Kemal Leblebicioğlu _____
Electrical and Electronics Engineering Dept., METU

Asst. Prof. Dr. Afşar Saranlı _____
Electrical and Electronics Engineering Dept., METU

Asst. Prof. Dr. İlkay Ulusoy _____
Electrical and Electronics Engineering Dept., METU

Asst. Prof. Dr. Uluç Saranlı _____
Computer Engineering Dept., Bilkent University

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 : Emre EGE

Signature :

ABSTRACT

A COMPARATIVE STUDY OF TRACKING ALGORITHMS IN UNDERWATER ENVIRONMENT USING SONAR SIMULATION

EGE, Emre

M. Sc., Department of Electrical and Electronics Engineering

Supervisor: Asst. Prof. Dr. Afşar SARANLI

September 2007, 102 pages

Target tracking is one the most fundamental elements of a radar system. The aim of target tracking is the reliable estimation of a target's true state based on a time history of noisy sensor observations. In real life, the sensor data may include substantial noise. This noise can render the raw sensor data unsuitable to be used directly. Instead, we must filter the noise, preferably in an optimal manner. For land, air and surface marine vehicles, very successful filtering methods are developed. However, because of the significant differences in the underwater propagation environment and the associated differences in the corresponding sensors, the successful use of similar principles and techniques in an underwater scenario is still an active topic of research. A comparative study of the effects of the underwater environment on a number of tracking algorithms is the focus of the present thesis. The tracking algorithms inspected are: the Kalman Filter, the Extended Kalman Filter and the Particle Filter. We also investigate in particular the IMM extension to KF and EKF filters. These algorithms are tested under several underwater environment scenarios.

Keywords: Underwater Target Tracking, Kalman Filter, Interactive Multiple Model, IMM, Particle Filter, Sonar Modeling

ÖZ

SUALTI ORTAMINDA SONAR BENZETİMİ KULLANARAK HEDEF İZLEME ALGORİTMALARININ KARŞILAŞTIRILMASI

EGE, Emre
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği
Tez Yöneticisi: Yrd. Doç. Dr. Afşar SARANLI

Eylül 2007, 102 sayfa

Hedef izleme radar sistemlerinin en temel öğelerinden birisidir. Hedef izlemenin amacı hedefin gerçek durum bilgisini yapılan gürültülü algılayıcı ölçümlerini kullanarak en iyi şekilde kestirmektir. Gerçek hayatta algılayıcıdan gelen veriler önemli ölçüde gürültüler içerebilir. Bu gürültü işlenmemiş almaç bilgisinin doğrudan kullanımına engel olabilir. Bunun yerine gürültüyü bilgiden en iyi şekilde süzgeçleyerek güvenilir bir durum kestirimi yapılabilmesi gereklidir. Kara, hava ve su üstü deniz araçları için birçok başarılı süzgeçleme yöntemi geliştirilmiştir. Ancak su altı ortamında ses dalgalarının yayınımındaki güçlü değişikliklerden ve kullanılan algılayıcıların önemli farklarından dolayı, benzeri başarılı yöntemlerin su altı ortamı için incelenmesi hala güncel bir araştırma alanıdır. Bu tezin odağı su altı ortamının hedef izleme yöntemlerine etkisinin karşılaştırmalı olarak incelenmesidir. İncelenen hedef izleme yöntemleri Kalman Filtresi, Genişletilmiş Kalman Filtresi ve Parçacık Filtresidir. Ayrıca Kalman Filtresi ve Genişletilmiş Kalman Filtresinin Etkileşimli Çoklu Model genişletmesi de incelenmiştir. Bu yöntemler çeşitli su altı senaryoları için denenmiştir.

Anahtar Kelimeler: Sualtı Hedef İzleme, Kalman Filtre, Etkileşimli Çoklu Model, EÇM, Parçacık Filtresi, Sonar Modelleme

To My Family

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my supervisor Asst. Prof. Dr. Afşar Saranlı for his guidance and motivation throughout this study.

I would also like thank to Prof. Dr. Kemal Leblebicioğlu for his support during this study.

I would also like thank to my room friends Kadir ISIYEL and Hüseyin YİĞİTLER for their contributions.

I would also like thank to my for my big brother Yunus EGE for his help in the grammar corrections in this study.

Finally, I would like to thank my family for their great support during my life. I am sure I would not be able to finish this work without their great motivation and support.

TABLE OF CONTENTS

ABSTRACT ..... **iv**
ÖZ ..... **v**
ACKNOWLEDGEMENTS ..... **vii**
TABLE OF CONTENTS ..... **viii**
LIST OF TABLES ..... **xi**
LIST OF FIGURES ..... **xii**
CHAPTERS
CHAPTER 1 INTRODUCTION **1**
1.1. State Estimation and Target Tracking 1
1.2. Underwater Target Tracking 3
1.3. Contributions of the Thesis 4
1.4. Outline of the Thesis 4
CHAPTER 2 ESTIMATION METHODS FOR TARGET TRACKING **6**
2.1. Introduction 6
2.2. Kalman Filter 7
2.2.1. Notes on Kalman Filter 12
2.3. Extended Kalman Filter 15
2.4. Interacting Multiple Model for KF and EKF 19
2.5. Particle Filter 24
2.6. Interacting Multiple Model for PF 27
2.7. Kinematic Models 30
2.7.1. White Noise Acceleration Model 30
2.7.2. Wiener Process Acceleration Model 31
2.7.3. Coordinated Turn Model 32

2.8. Performance Comparison of Tracking Algorithms for Aerial Vehicles 34
2.8.1. Performance Comparison of KF and PF for Linear Model.....36
2.8.2. Performance Comparison of KF, EKF and PF for non-linear Motion...39
2.8.3. Performance Comparison of IMM-KF and IMM-PF for a Maneuvering Target 43
CHAPTER 3 ACOUSTIC MODELLING.....51
3.1. Introduction 51
3.2. Propagation of Sound in Underwater Environment 52
3.2.1. Transmission Loss.....53
3.2.1.1. Spreading Loss 53
3.2.1.2. Absorption Loss 54
3.2.2. Speed of Sound.....55
3.2.3. Propagation Models.....56
3.2.3.1. Ray Theory.....56
3.2.3.2. Normal Mode 58
3.2.3.3. Other Propagation Models 58
3.3. Noise.....60
3.3.1. Thermal Noise 61
3.3.2. Ambient Noise.....62
3.3.3. Other Noise Sources.....63
3.4. Reverberation 64
3.5. Sonar Simulator Development for the Present Study.....65
3.5.1. Propagation Model 65
3.5.2. Measurement Model.....67
3.5.2.1. Transmitted Pulse Model 68
3.5.2.2. Received Signal Generation Model.....69
3.5.2.3. Matched Filtering 70
3.5.2.4. Specifying the Sound Wave That is Reflected From Reflector 72
3.5.2.5. Range Calculation 75
3.5.2.6. Bearing Calculation.....76
3.5.2.7. Post Processing Stage.....77
CHAPTER 4 UNDERWATER TARGET TRACKING SIMULATION AND RESULTS.....79

4.1. Introduction 79
4.2. Scenario 1 80
4.3. Scenario 2 83
4.4. Scenario 3 86
4.5. Discussion 92
CHAPTER 5 CONCLUSION 94
5.1. Discussion of Results 94
5.2. Future Work 95
APPENDIX A 97
REFERENCES 99

LIST OF TABLES

Table 2.1 Descriptions of symbols used in system model 7
Table 2.2 Discrete-Time Kalman Filter Equations 11
Table 2.3 Discrete-Time Extended Kalman Filter 18
Table 3.1 Speed of Sound Formulation Parameter Ranges.....56
Table 3.2 Domains of Applicability of Underwater Propagation Models 59
Table 3.3 Sea state, wave and wind speed 62

LIST OF FIGURES

Figure 2.1 Kalman Filter Loop....12
Figure 2.2 Extended Kalman Filter Loop....18
Figure 2.3. Flowchart of IMM Algorithm....21
Figure 2.4 True Target Positions at Each Sampling Time....36
Figure 2.5 RMS Position Error....37
Figure 2.6 Normalized RMS Position Error....38
Figure 2.7 RMS Velocity Error....38
Figure 2.8 Performance vs. Time Comparison....39
Figure 2.9 True Trajectory of Target....40
Figure 2.10 RMS Position Error....41
Figure 2.11 Normalized RMS Position Error....41
Figure 2.12 RMS Velocity Error....42
Figure 2.13 Performance vs. Time Comparison....43
Figure 2.14True Trajectory of The Target....44
Figure 2.15 RMS Position Errors....45
Figure 2.16 Normalized RMS Position Errors....45
Figure 2.17 RMS Velocity Errors....46
Figure 2.18 IMM-KF Model Probabilities....47
Figure 2.19 IMM-PF-100 Model Probabilities....48
Figure 2.20 IMM-PF-500 Model Probabilities....49
Figure 2.21 Performance vs. Time Comparison....50
Figure 3.1 Refraction of Sound Between Different Layers....57
Figure 3.2 Propagation of Sound Wave in Underwater Environment. Output of the Rayson underwater communications software[38].....60
Figure 3.3 Thermal Noise characteristic....61
Figure 3.4 Ambient noise of the Sea....63

Figure 3.5 Propagation of Sound Wave Between Layered Underwater Environment 66
Figure 3.6 Transmitted Signal 68
Figure 3.7 Received Signal 70
Figure 3.8 Filtered Received Signal 71
Figure 3.9 Propagation of Sound in Underwater Environment 73
Figure 3.10 Sound Wave Reflected From Target 74
Figure 3.11 Ghost Image of The Target 77
Figure 4.1 True Trajectory of target 80
Figure 4.2 RMS Position Error 81
Figure 4.3 Normalized RMS Position Error 82
Figure 4.4 RMS Velocity Error 82
Figure 4.5 True Target Trajectory 84
Figure 4.6 RMS Position Errors 84
Figure 4.7 Normalized RMS Position Errors 85
Figure 4.8 RMS Velocity Errors 85
Figure 4.9 True Trajectory of Target 87
Figure 4.10 RMS Position Errors 88
Figure 4.11 Normalized RMS Position Errors 88
Figure 4.12 RMS Velocity Errors 89
Figure 4.13 IMM-KF Model Probabilities 90
Figure 4.14 IMM-PF-50 Model Probabilities 91
Figure 4.15 IMM-PF-500 Model Probabilities 92

CHAPTER 1

INTRODUCTION

1.1. State Estimation and Target Tracking

State estimation is the collection of techniques to find the best approximation to the true value of a state variable that can not be observed directly or the observation itself is contaminated by random noise. State estimation algorithms use the system dynamic model, a history of output observations and noise models for both the system dynamics and output observations.

There are numerous methods for state estimation. We can categorize the estimators as Bayesian and Non-Bayesian [1, 2]. Let us assume that an estimate for a variable (x) is required. In the Bayesian approach the parameter (x) has a prior probability density function (pdf) $p(x)$. But in the Non-Bayesian approach, no such known prior pdf is assumed for the parameter (x).

The general techniques within the Non-Bayesian approach are the maximum likelihood (ML) and least squares (LS) techniques [4]. In the ML technique a likelihood function is constructed. The construction of likelihood function depends

on the state and observations. After the likelihood function is constructed, the estimate for the state(s) can be found by maximizing the likelihood function. In LS technique the aim is to find the state estimate by minimizing a mean square error according to:

$$\hat{x}^{LS} = \arg\min_{x} \sum_{i=1}^{t} (y_i - h(x))^2 \tag{1.1}$$

where $\hat{x}^{LS}$ denotes least square estimation of variable x, y denotes measurement and $h(x)$ denotes measurement relation of x.

The general methods for Bayesian approach are Maximum A-Posteriori (MAP) estimation and Minimum Mean Square Error (MMSE) estimation [3]. MAP is based on maximizing the posteriori pdf of variable x. MMSE is based on minimization of expected value of given measurements.

Estimation methods are used in many areas of engineering such as tracking, navigation, guidance, signal processing as well as telecommunications. In this thesis we are using estimation methods for tracking applications, in particular in underwater acoustic setting. In target tracking applications estimation methods are used to optimally predict the next state of a possibly moving target from given noisy sensor observations. The most commonly used method in target tracking is the Kalman Filter (KF) [5, 6]. The Kalman Filter is the optimal estimator when the system model is linear and the noise is zero-mean, white Gaussian noise. However in many real world scenarios the systems are non-linear and the noise may be non-Gaussian. These effects violate the main assumptions of the KF and the KF may cause the deterioration of the Kalman Filter estimation performance. For this case where the system is non-linear, the KF was extended by linearizing the system (finding a linear approximation to the system). The resulting method is called the Extended Kalman Filter (EKF) [7]. In EKF, the model is linearized at every time step. EKF is a suboptimal algorithm. As in the Kalman filter, in EKF the noise is assumed to be zero-mean white Gaussian. For maneuvering targets where the dynamics of the target may be changing, Multiple Model (MM) methods are

proposed as presented in [8]. The MM approaches use more than one system dynamic model and make the overall estimation according to a weighting of the models. The most commonly used MM based method is Interactive Multiple Model (IMM) method [9]. IMM assigns weightings to all models and makes the overall estimation according to these weightings.

Another method for state estimation and target tracking is the Particle Filtering (PF) approach [10]. PF is based on Sequential Monte Carlo filtering [22, 23]. PF does not have the assumptions critical to KF and EKF such as linearity of the models or the Gaussianity of the noise. Recently, PF approach is heavily used in tracking applications recently. Clearly, its most important advantage is the aforementioned relaxed assumptions on the system model and noise. The most important drawback of PF is the significantly higher computational load.

1.2. Underwater Target Tracking

The tracking algorithms are largely used for land, air and sea surface vehicle tracking applications. Generally, the sensor used for these applications is RADAR. But in an underwater environment, the RADAR cannot be used due to the very high attenuation of the RF band electromagnetic waves. Instead of using the RF band electromagnetic waves in underwater, much lower frequency sound waves are used. Sound waves are much less attenuated by the medium and can travel long distances. The most common sensor used in underwater, using sound waves is SONAR. Sonar works like radar, namely it periodically transmits pulse packets and measures the time delay of the reflected pulse coming from a target. From these measurements, a noisy range and bearing information can be extracted. However, because of the underwater effects on the traveling sound waves, the sonar measurements are less reliable than radar. There are many dispersive effects of the underwater medium. Also multi-paths (multiple reflections of the transmitted sound pulse) are more common in this case.

In the present thesis, performances of tracking algorithms are compared for the underwater environment. Because of the difficulty of modeling all possible effects for this medium, we are selecting a small but interesting subset of phenomena and making some assumptions for defining a tractable problem. We are developing a sonar model for generating the observations that the tracking algorithms would face. In particular, our sonar model considers multiple layers of the underwater medium. The medium is partitioned into layers because it is typically the case in real-life and the speed of propagation of the sound waves differ in different layers. This causes refraction of the sound waves and changes the direction of propagation. The basic elements that change the sound propagation speed (and hence direction) are salinity, temperature and depth.

1.3. Contributions of the Thesis

The main contribution of the thesis is an attempt to characterize and compare the performances of the established state estimation and target tracking algorithms in a number of underwater scenarios. For achieving this purpose, the algorithms are implemented in Matlab and characterized for the general case first. Then tractable models are selected and implemented for the acoustic Sonar sensor case where a multi-layered medium that distorts the sound waves is considered. Algorithms are applied for a number of simulated medium-target-motion scenarios in order to explicitly show the strengths and weaknesses of the methods. A reusable Matlab model is developed to support the simulations.

1.4. Outline of the Thesis

We can categorize thesis presentation into three main sections: Tracking algorithms, their description, implementation and characterization, the underwater acoustic sensor (Sonar) modeling theory and the considered implementation, the integration of the tracking algorithms with the sonar model and comparative studies of performance for a number of different scenarios.

In Chapter 1, the current state of art in target tracking, underwater acoustic modeling and underwater target tracking is discussed. Since the field of state estimation and target tracking in general is vast and our focus is on the underwater case, we have selected the most popularly used and widely accepted approaches for our study.

In Chapter 2, the target tracking issue is further explored. First, a background of target tracking is given. In the associated subsections firstly Kalman Filtering, the derivation of KF and some key points of the KF are presented. Secondly EKF and its derivation are given. Then the derivation of IMM is presented followed by the particle filter - PF. Chapter 2 concludes with the simulation of these methods and presentation of their results for the non-underwater environment with the purpose of exposing the standard behavior of these algorithms considered.

In Chapter 3, we focus on acoustic modeling for sonar and present aspects of the problem. The propagation model of sound in underwater environment, noise sources in this medium and reverberation is explained consecutively. Then the sonar model is described. Lastly some simulation results are presented to illustrate the model based generation of sonar observations for the tracking algorithms.

In Chapter 4, target tracking in underwater environment is considered by combining the techniques of the previous two chapters. Firstly the main underwater effects, environmental effects and our simulated experimental setup are given. Our experimental setup is based on a number of distinct scenarios with specific environment-target-motion conditions. Finally in this section, the simulation results and their discussions are presented with respect to these scenarios.

The thesis is finally concluded Chapter 5 with an overall discussion and suggestions for possible extensions to the present work.

CHAPTER 2

ESTIMATION METHODS
FOR
TARGET TRACKING

2.1. Introduction

As stated in the introduction chapter, there are several techniques for estimation. But in this thesis we focus on KF, EKF, IMM and PF. In this chapter these methods are explained. The simulated results of these methods are given at the end of the section.

KF is used for state estimation on linear dynamic system which is driven by white noise. If the assumptions are provided, KF is the optimal estimator. EKF is used for non-linear dynamic system. It is similar to KF but at every time step the system is linearized (first order series expansion). IMM uses several models and the estimation is done according to the weights of the models. PF is an alternative to KF and its extensions. In PF, estimation is based on Bayesian approach. Details of the above methods are given in the following subsection.

2.2. Kalman Filter

As stated earlier, the KF assumes the system model is linear and the noise is white Gaussian. The optimality is in the sense of minimizing the mean square error of the estimated parameters. In this case KF is the optimal estimator. KF is best-linear estimator when the system is linear but the noise is non-Gaussian. In following, the KF is derived according to [1, 11]. Consider a linear dynamic system as given below:

$$x(k+1) = F(k)x(k) + G(k)u(k) + v(k) \qquad k = 1,... \qquad (2.1)$$

$$z(k) = H(k)x(k) + w(k) \qquad k = 1,... \qquad (2.2)$$

The symbols used in equations(2.1), (2.2) is described in Table 2.1

Table 2.1 Descriptions of symbols used in system model

Symbol	Size	Description
k	*1*	Time index
x(k)	*n*x*1*	System state vector at time *k*
F(k)	*n*x*n*	State transition matrix at time *k* (this is a relation with *x(k)* and *x(k+1)*)
u(k)	*u*x1	Input vector at time *k*
G(k)	*n*xu	Input gain matrix
v(k)	*n*x*1*	Process noise vector. It is assumed to be a zero-mean white Gaussian
z(k)	*m*x*1*	Measurement vector at time *k*
H(k)	*m*x*n*	Measurement matrix. This matrix gives the ideal measurement at time *k* related to state vector at time *k*
w(k)	*m*x*1*	Measurement noise vector. It is assumed to be a zero-mean white Gaussian.

The equation (2.1) is called "Plant Equation" and equation (2.2) is called "Measurement Equation". The covariance matrix for *v(k)* is:

$$E[v(k)v(k)'] = Q(k) \tag{2.3}$$

And the covariance matrix for *w(k)*:

$$E[w(k)w(k)'] = R(k) \tag{2.4}$$

The *v(k)* and w*(k)* noise sequence assumed to be mutually independent. This assumption yields:

$$E[v(k)w(i)'] = 0, \qquad \text{for all } k \text{ and } i \tag{2.5}$$

All the above, *x(k), v(k)* and *w(k)*, are mutually uncorrelated. Thus, we can write the following equations:

$$E[x(0)v(k)'] = 0 \qquad \forall k \tag{2.6}$$

$$E[x(0)w(k)'] = 0 \qquad \forall k \tag{2.7}$$

Since we assume the *v(k)* and *w(k)* are white Gaussian, we can write the following easily:

$$E[v(k) \mid Z^k] = E[v(k)] = 0 \tag{2.8}$$

$$E[w(k) \mid Z^{k-1}] = E[w(k)] = 0 \tag{2.9}$$

The initial state, *x(0),* can be known but in the unknown situation it can be modeled as a random variable.

The notations in the KF are given below.

$$\hat{x}(k+1 \mid k) = E[x(k+1) \mid Z^k] \tag{2.10}$$

$$\hat{x}(k \mid k) = E[x(k) \mid Z^k] \tag{2.11}$$

Z^k is the observation available at time k. The equation (2.10) is called as "Prior Estimate (or priori)" and equation (2.11) is called as "Updated Estimate(or posteriori)". Prior estimate is used to predict state *x(k+1)* using the measurements up to *k.* Updated estimate is the estimated state *x(k)* using measurements up to *k.*

We can find the estimation error using equation (2.12)

$$\tilde{x}(j \mid k) = x(j) - \hat{x}(j \mid k) \tag{2.12}$$

The error covariance matrix associated with the estimation error is:

$$\begin{aligned} P(j \mid k) &= E[[x(j) - \hat{x}(j \mid k)][x(j) - \hat{x}(j \mid k)]' \mid Z^k] \\ &= E[\tilde{x}(j \mid k)\tilde{x}(j \mid k)' \mid Z^k] \end{aligned} \tag{2.13}$$

The calculation of priori estimate can be calculated as:

$$E[x(k+1) \mid Z^k] = E[F(k)x(k) + G(k)u(k) + v(k) \mid Z^k] \tag{2.14}$$

By using equation (2.8) we can rewrite (2.14) as:

$$\hat{x}(k+1 \mid k) = F(k)\hat{x}(k \mid k) + G(k)u(k) \tag{2.15}$$

Equation (2.15) is used to calculate the state prediction (prior estimate). If we subtract (2.15) from (2.1), the result gives the state prediction error.

$$\tilde{x}(k+1 \mid k) = x(k+1) - \hat{x}(k+1 \mid k) = F(k)\tilde{x}(k \mid k) + v(k) \tag{2.16}$$

We can find the state prediction error covariance matrix as follows:

$$\begin{aligned} &E[\tilde{x}(k+1 \mid k)\tilde{x}(k+1 \mid k)' \mid Z^k] \\ &= F(k)E[\tilde{x}(k \mid k)x(k \mid k') \mid Z^k]F(k)' + E[v(k)v(k)'] \end{aligned} \tag{2.17}$$

We can rewrite the (2.17) as following and the following equation gives state prediction covariance:

$$P(k+1 \mid k) = F(k)P(k \mid k)F(k)' + Q(k) \tag{2.18}$$

We have derived the prediction of state. In the following the measurement prediction formulization is given as follows. Measurement prediction formula can be found by taking the expected value of (2.2) conditioned on the measurements Z^k.

$$E[z(k+1) \mid Z^k] = E[H(k+1)x(k+1) + w(k+1) \mid Z^k] \tag{2.19}$$

Using the equation (2.9) we can rewrite the equation as:

$$\hat{z}(k+1 \mid k) = H(k+1)\hat{x}(k+1 \mid k) \tag{2.20}$$

Equation (2.20) is used to calculate the measurement prediction. If we subtract (2.20) from measurement equation (2.2), the result gives the measurement prediction error.

$$\tilde{z}(k+1 \mid k) = z(k+1) - \hat{z}(k+1 \mid k) = H(k+1)\tilde{x}(k+1 \mid k) + w(k+1) \tag{2.21}$$

By using (2.21) we can write the measurement prediction covariance as:

$$S(k+1) = H(k+1)P(k+1|k)H(k+1)' + R(k+1) \tag{2.22}$$

Now we have predicted state and measurement vectors and their covariances. We can now find the covariance between state and the measurement. This covariance is used to find the KF gain. Derivations of gain and some equations are not given in this section. The intermediate steps can be found in [1]. The covariance between predicted state *x(k+1)* and the predicted measurement *z(k+1)* can be written as follows:

$$\begin{aligned} &E[\tilde{x}(k+1|k)\tilde{z}(k+1|k)' | Z^k] \\ &= E[\tilde{x}(k+1|k)[H(k+1)\tilde{x}(k+1|k) + w(k+1)]' | Z^k] \\ &= P(k+1|k)H(k+1)' \end{aligned} \tag{2.23}$$

The KF gain can be given as follows:

$$W(k+1) = P(k+1|k)H(k+1)'S(k+1)^{-1} \tag{2.24}$$

The KF gain is used in updating the estimation. The updated state estimate can be given as:

$$\hat{x}(k+1|k+1) = \hat{x}(k+1|k) + W(k+1)v(k+1) \tag{2.25}$$

where

$$v(k+1) = z(k+1) - \hat{z}(k+1|k) = \tilde{z}(k+1|k) \tag{2.26}$$

An important point at equation (2.25) is *v(k+1)*. We have defined *v(k+1)* as process noise vector but in this equation it is used as innovation or measure residual. Note that innovation is measurement error. And thus *S* is innovation covariance.

The last remaining calculation is state covariance. In equation (2.18) we have predicted the state covariance for time *k+1*. Now, we must update this state covariance using the KF gain. The updated state covariance is as follows:

$$\begin{aligned} &P(k+1|k+1) \\ &= P(k+1|k) - P(k+1|k)H(k+1)'S(k+1)^{-1}H(k+1)P(k+1|k) \\ &= [I - W(k+1)H(k+1)]P(k+1|k) \end{aligned} \tag{2.27}$$

and equation (2.27) can be rewritten as:

$$P(k+1|k+1) = P(k+1|k) - W(k+1)S(k+1)W(k+1) \tag{2.28}$$

Discrete-time KF is summarized in Table 2.2. Each equation for each step is given in this table.

Table 2.2 Discrete-Time Kalman Filter Equations

Model	$x(k+1) = F(k)x(k) + G(k)u(k) + v(k)$, v(k) ~ N(0,Q(k)) z(k)=H(k)x(k)+w(k), w(k) ~ N(0,R(k))
State and Covariance Prediction	$\hat{x}(k+1\mid k) = F(k)\hat{x}(k\mid k) + G(k)u(k)$ $P(k+1\mid k) = F(k)P(k\mid k)F(k)' + Q(k)$
Gain	$W(k+1) = P(k+1\mid k)H(k+1)'[H(k+1)P(k+1\mid k)H(k+1)' + R(k+1)]^{-1}$
State and Covariance Update	$\hat{x}(k+1\mid k+1) = \hat{x}(k+1\mid k) + W(k+1)[H(k+1)\tilde{x}(k+1\mid k) + w(k+1)]$ $P(k+1\mid k+1) = [I - W(k+1)H(k+1)]P(k+1\mid k)$

The KF has 2 main steps: Prediction and Update. In prediction step, state and covariance predictions are calculated. When a measurement comes, the update step starts. In the update step first KF gain is calculated and then the state and covariances are updated according to gain and measurement. Note that in KF gain equation (2.24) there is no need to an observation. Thus we can calculate the KF gain offline. In the Figure 2.1 a Kalman Filter loop is given. Once the KF is initialized, this loop turns until filtering operation is finished.

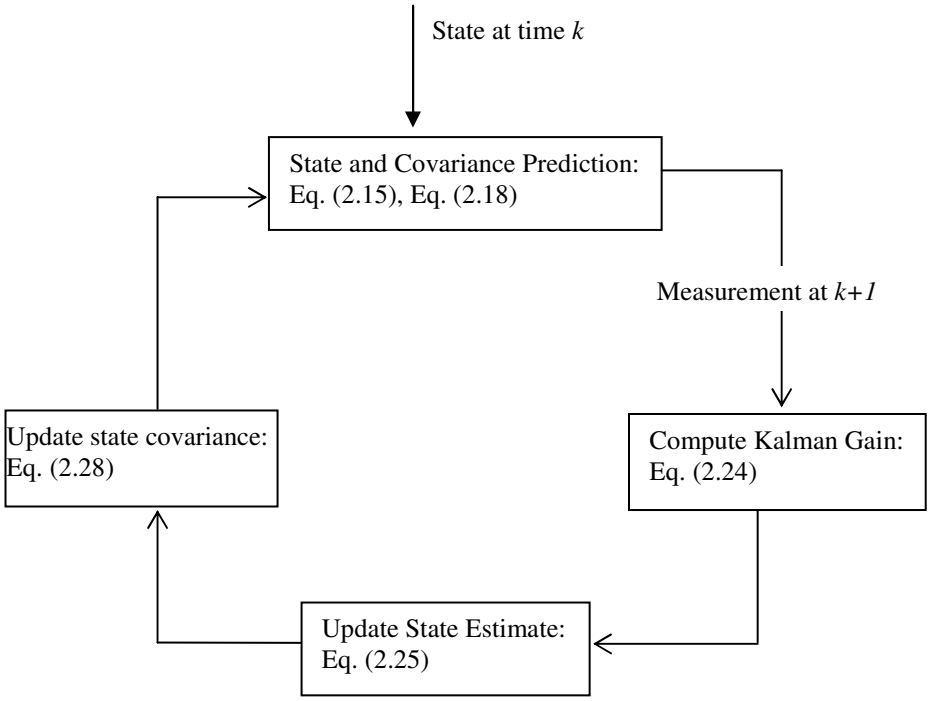

Figure 2.1 Kalman Filter Loop

2.2.1. Notes on Kalman Filter

In the previous subsection, general structure of the KF is given. In this subsection, some details of KF will be given.

As discussed above it is assumed that in the KF initialization model the initial state is a random variable, and its pdf is known and assumed to be Gaussian. Thus we can choose the initial state as:

$$\hat{x}(0|0) \sim N(x(0), P(0,0)) \tag{2.29}$$

The equation (2.29) says that the initial prediction of state x is a random variable with mean equal to $x(0)$ (true state) and its variance is equal to initial covariance. But in the practice we can not know the true state. So we must do initialization after

a number of observations are made. One method is two-point differencing which needs 2 observations. This method guarantees the consistency of initialization of the filter [1]. Assume the system have 2 state components, x and $\dot{x}$, position and velocity respectively. If the measurements only include position information then we can write that:

$$\hat{x}(0) = z(0) \tag{2.30}$$

$$\hat{\dot{x}}(0) = \frac{z(0) - z(-1)}{T} \tag{2.31}$$

where T is sampling period and $z(k)$,k=-1,0 is observations.

We can obtain the corresponding initial covariance matrix as follows:

$$P(0|0) = \begin{bmatrix} R & R/T \\ R/T & 2R/T \end{bmatrix} \tag{2.32}$$

where R is measurement covariance.

Another important point in the KF optimality is forms of covariance matrices Q and R. The main properties of these covariances are: Symmetry and positive definiteness. While predicting and the updating of these covariances, the symmetry and the positive definiteness can be lost due to round-off errors in the computation.

In computer arithmetic, the data, fixed or float, are stored with a fixed number of bits. This can cause errors for very precise situations. These errors are called round-off errors. In many precise applications round-off error is a big problem. Although the numeric value of this error very small, the results of this error can be very big. Round-off error can be defined as follows [2]:

$$1 + \varepsilon_{roundoff} = 1 \tag{2.33}$$

An example which shows that round-off errors that make matrix singular (non-invertible) is given below. This example is taken from [14]. Consider a measurement matrix as follows:

$$H = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1+\delta \end{bmatrix} \tag{2.34}$$

and the covariance matrices

$$P = I_3 \qquad\qquad R = \delta^2 I_2 \tag{2.35}$$

where I_n denotes nxn identity matrix and $\delta^2 < \varepsilon_{roundoff}$ but $\delta > \varepsilon_{roundoff}$. In equation (2.22) the result of $H(k+1)P(k+1|k)H(k+1)'$ is

$$\begin{bmatrix} 3 & 3+\delta \\ 3+\delta & 3+2\delta \end{bmatrix} \tag{2.36}$$

The result (2.36) is singular. Next step is adding *R(k+1)* to (2.36). This operation also doesn't change singularity so the matrix is still singular. When calculating the gain of KF (2.24), the inverse of measurement prediction covariance must be calculated ($S(k+1)^{-1}$). Hence measurement prediction covariance is singular it is not invertible. Thus an ill-condition occurs and computation fails.

To prevent round-off errors and loss of basic covariance matrix properties, the equation (2.27) can be rewritten as following:

$$\begin{aligned} P(k+1|k+1) &= [I - W(k+1)H(k+1)]P(k+1|k) \\ &.[I - W(k+1)H(k+1)]' + W(k+1)R(k+1)W(k+1)' \end{aligned} \tag{2.37}$$

The equation (2.37) is called Joseph form covariance update. Equation (2.37) is more computationally expensive than (2.27) and (2.28) but it prevents the loss of symmetry and the loss of positive definiteness. The multiplications of the matrices prevent the loss of symmetry. Since the only subtraction is in the *I-WH* this form of covariance update prevents the loss of positive definiteness [1].

There are also other methods like square-root filtering, information filter, etc. that makes the calculation precision better and prevents the loss of basic properties of covariance. In the practice if these methods haven't used, the covariances can loose their symmetry or positive definiteness and this yields the estimator to be non-optimal. And it is difficult to find the cause of non-optimality. Once these properties have lost the estimator works wrong and possibly the error increases.

2.3. Extended Kalman Filter

As discussed in previous section KF addresses the problem of estimating the state from noisy measurements in a linear dynamic system and the noises are Gaussian and independent from each other. In this case the KF is the optimal solution. But in the non-linear system case the KF doesn't works. The solution lies in linearization. We can linearize a non-linear system using linearization methods like Taylor series expansion and then we can apply KF to the linearized system. There are two basic linearization techniques used for non-linear systems. One of them is linearization about a nominal trajectory and the other is linearization about estimated trajectory [1, 2, 11, 12]. In literature the first method is called Linearized Kalman Filter, and the second method is called EKF. In this section the EKF will introduced. The detailed information about Linearized Kalman Filter can be found at[1, 2, 3, 13].

Generally dynamic system models and sensor model are not linear but not very far from linearity. We can think a nonlinear system as a linear system with some perturbations. By this way we can apply the KF for non-linear models. We can say that a system is non-linear if any one or both of the plant equation or measurement equation is non-linear.

A non-linear dynamic system model can be given as follows:

$$x(k+1) = f[k, x(k)] + v(k) \tag{2.38}$$

where f is a non-linear state transition function that relates the state at time k to the state at time $k+1$. Here v is process noise and assumed to be zero mean white Gaussian. The noise v is independent. So we can write that:

$$E[v(k)] = 0 \tag{2.39}$$

$$e[v(k)v(j)'] = Q(k)\delta_{kj} \tag{2.40}$$

where δ_{kj} is Dirac delta function.

We can write the measurement equation as follows:

$$z(k) = h[k, x(k)] + w(k) \tag{2.41}$$

where h is a non-linear measurement function that relates the measurement at time k to the state at time k. Here w is measurement noise and assumed to be zero mean white Gaussian. The noise w is independent. So we can write that:

$$E[w(k)] = 0 \tag{2.42}$$

$$E[w(k)w(j)'] = R(k)\delta_{kj} \tag{2.43}$$

As it can be seen above, the main assumptions are same with the standard KF. Now we can derive the EKF. For simplicity the EKF algorithm is derived according to flow of KF algorithm. First prediction step for EKF is discussed and then update step is discussed.

To find the prediction in the non-linear system first we must linearize the system by Taylor series expansion and then using the linear system we can find the state prediction. In [1], the non-linear system is expanded by Taylor series about the estimated state up to second order term. The higher order terms can be neglected. Generally first order expansion gives enough accurate linearization. In this research the expansion is done up to first ordered terms for simplicity. The Taylor series expansion of (2.38) up to first order terms is as follows:

$$\begin{aligned} x(k+1) &= f[k, \hat{x}(k \mid k)] + f_x(k)[x(k) - \hat{x}(k \mid k)] \\ &+ HOT + v(k) \end{aligned} \tag{2.44}$$

where HOT is abbreviation for higher order terms and $f_x(k)$ is Jacobian of f and can be given as follows:

$$f_x(k) = [\nabla_x f(k, x)']'\big|_{x=\hat{x}(k|k)} = \frac{\partial f}{\partial x} \tag{2.45}$$

If we have expanded up to second ordered terms we would have calculate the Hessian. As information the Hessian of the i th component of f (f_{xx}^i) can be given by:

$$f_{xx}^i = [\nabla_x \nabla_x^j f^i(k, x)]\Big|_{x=\hat{x}(k|k)} = \frac{\partial^2 f^i}{\partial x^2} \tag{2.46}$$

According to the equations given above we can write the state prediction equation as follows:

$$\hat{x}(k+1|k) = f[k, \hat{x}(k|k), u(k)] \tag{2.47}$$

We can find the state prediction error is by subtracting (2.47) from (2.44) and then state prediction covariance can be as follows:

$$P(k+1|k) = f_x(k)P(k|k)f_x(k)' + Q(k) \tag{2.48}$$

Similarly we can derive the measurement prediction as follows:

$$\hat{z}(k+1|k) = h[k+1, \hat{x}(k+1|k)] \tag{2.49}$$

From equation (2.49) we can obtain measurement prediction covariance as:

$$S(k+1) = h_x(k+1)P(k+1|k)h_x(k+1)' + R(k+1) \tag{2.50}$$

where ;

$$h_x(k+1) = [\nabla_x h(k+1, x)']'\big|_{x=\hat{x}(k+1|k)} = \frac{\partial h}{\partial x} \tag{2.51}$$

When a measurement comes we can get in to update step. In update step we must calculate the EKF gain. In the following equation the gain is given:

$$W(k+1) = P(k+1|k)h_x(k+1)'S(k+1)^{-1} \tag{2.52}$$

After EKF gain is obtained, update the state estimate as:

$$\hat{x}(k+1|k+1) = \hat{x}(k+1|k) + W(k+1)v(k+1) \tag{2.53}$$

where *v(k+1)* denotes measurement residual and calculated by:

$$v(k+1) = z(k+1) - \hat{z}(k+1|k) \tag{2.54}$$

Finally update the state covariance as following:

$$P(k+1|k+1) = P(k+1|k) - W(k+1)S(k+1)W(k+1)' \tag{2.55}$$

The complete set of EKF prediction and update steps equations are given in Table 2.3

Table 2.3 Discrete-Time Extended Kalman Filter

Model	$x(k+1) = f[k, x(k)] + v(k), \quad v(k) \sim N(0, Q(k))$ $z(k) = h[k, x(k)] + w(k), \quad w(k) \sim N(0, R(k))$
State and Covariance Prediction	$\hat{x}(k+1\mid k) = f[k, \hat{x}(k\mid k), u(k)]$ $P(k+1\mid k) = f_x(k) P(k\mid k) f_x(k)' + Q(k)$
Gain	$W(k+1) = P(k+1\mid k) h_x(k+1)' S(k+1)^{-1}$
State and Covariance Update	$\hat{x}(k+1\mid k+1) = \hat{x}(k+1\mid k) + W(k+1) v(k+1)$ $P(k+1\mid k+1) = P(k+1\mid k) - W(k+1) S(k+1) W(k+1)'$

In the Figure 2.2 an EKF loop is given. Once the EKF is initialized, this loop turns until filtering operation is finished.

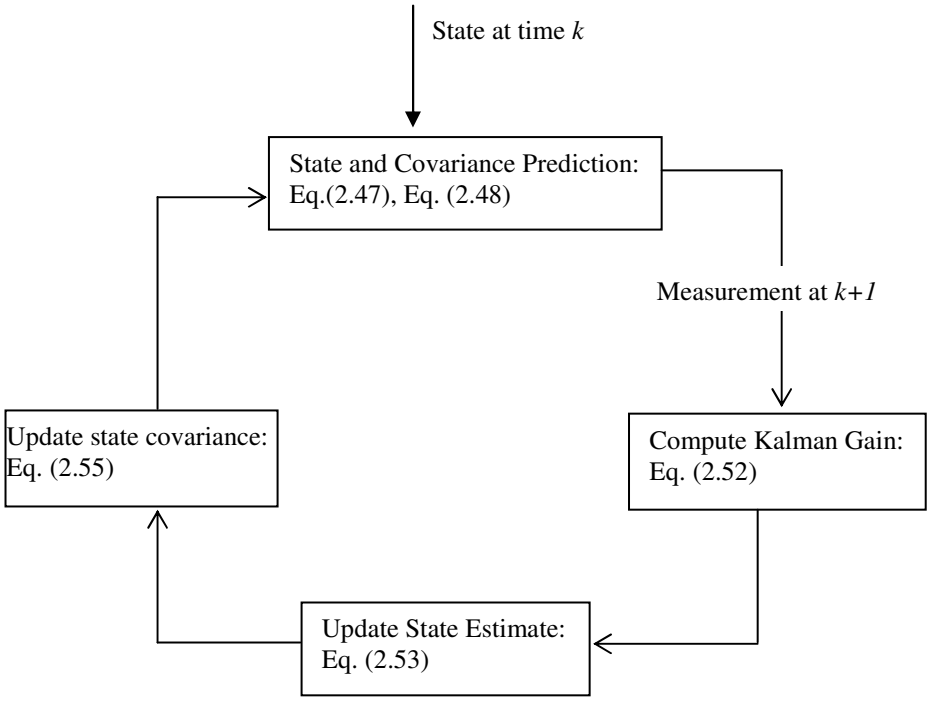

Figure 2.2 Extended Kalman Filter Loop

2.4. Interacting Multiple Model for KF and EKF

Only one model is used in KF or EKF. This causes high errors in estimation if the assumed model mismatches than the target's. Consider a target that moves along a line and then it makes a turn in clockwise direction and then continue to move along a line. In this situation we need at least two models: For moving along a line we need 'White Noise Acceleration Model (WNA)' or 'Wiener Process Acceleration Model(WPA)', and for turn we need 'Coordinate Turn Model(CT)'. These models are discussed later in this chapter. For KF or EKF, if WNA or WPA model is chosen there will be no problem for moving along a line but in the turn the estimation will be wrong since our model mismatches. If CT model is chosen then there will be no problem for turn maneuver but in the linear movement the estimations will be wrong again due to model mismatch. To resolve this problem Multiple Model approach is developed. In multiple model approach it is assumed that the system obeys one of a finite number of models. The appropriate model is chosen according to modal probabilities. Model probabilities are assigned to each model according to which model gives more related estimation with measurement

There are several MM approaches like Generalized Pseudo-Bayesian estimator of first order(GPB1), Pseudo-Bayesian estimator of second order(GPB2) [15, 16] and Interacting Multiple Model (IMM) [17, 18].

The IMM estimator is a suboptimal hybrid filter and it is one of best cost-effective hybrid state estimators. Although its complexity is less, the performance of the IMM is nearly same with the other complex algorithms. Another advantage of the IMM is that it has a modular structure. It can be used with KF, EKF and also with particle filters. So it is possible to handle non linear system by this structure of IMM. The IMM is based on several motion models and a probabilistic (Markov chain) switching between these models. Consider an IMM based on two models: one constant velocity and the other is coordinated turn model. When the target moves with the constant velocity, the mode probability of the first model increases

while the other decreasing. But when the target makes a turn maneuver, the mode probability of first model decreases while the mode probability of coordinated turn model is increases. The IMM for KF and EKF is notated as IMM-KF.

Derivation of IMM-KF is given below. The derivation of the IMM-KF is given as in [1,20]. The main feature of the IMM-KF is that state estimates and covariances from each model are combined according to a Markov model based on the target maneuver. The state estimation is computed using each model and their probability. It is possible to say that IMM-KF is a system in which *r* number of filter running in parallel and the final estimation is computed by combining the result of that *r* filter according to probability assigned to each filter.

Before deriving the IMM-KF steps, a picture of the IMM-KF estimator calculation steps is given in the Figure 2.3[19]. The Figure 2.3 shows the flowchart of IMM-KF algorithm. For each time step this loop restarts and it continues until the end of tracking.

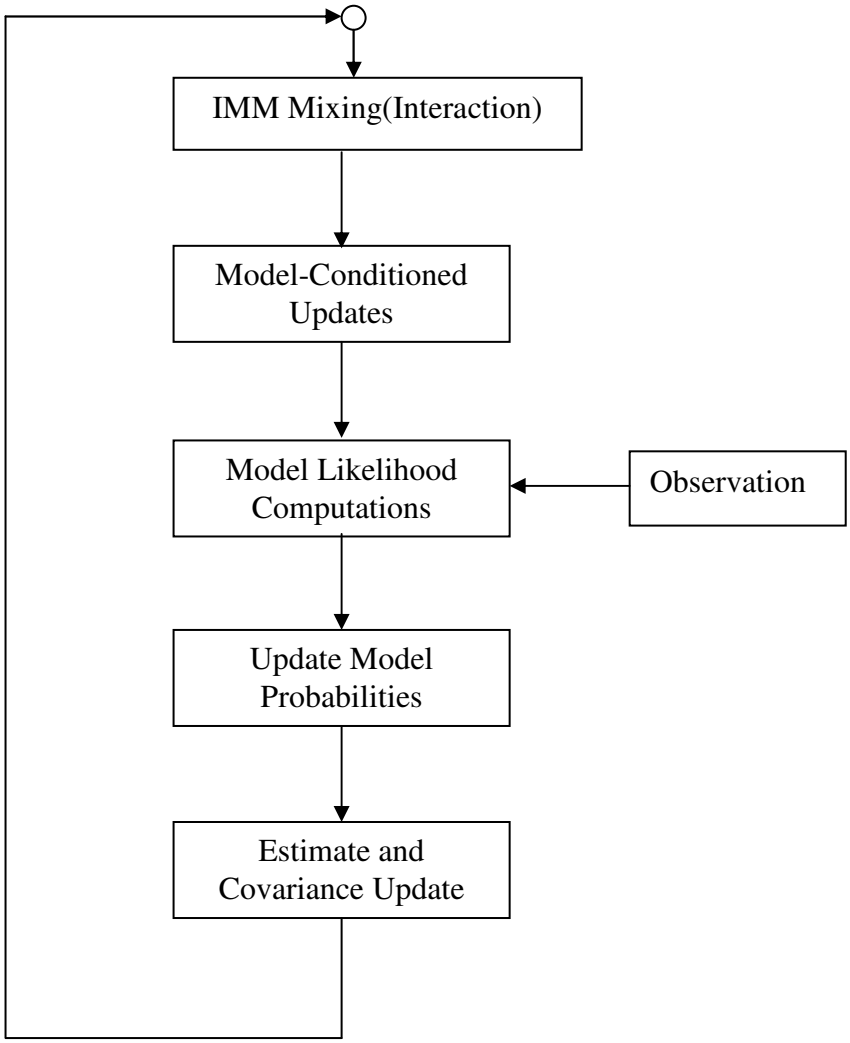

Figure 2.3. Flowchart of IMM Algorithm

First step is mixing of state estimates. In this step model-conditioned estimates are calculated using $\hat{x}^i(k-1|k-1)$ i=1,...r, where r is the number of models and $\hat{x}^n(k-1|k-1)$ is the model-conditioned estimate of model n at time *k-1*. The associated covariance for model n is denoted as $P^n(k-1|k-1)$ and the model probabilities calculated at time k-1 is denoted as $\mu^n(k-1|k-1)$.

For the filter matched to model $M^j(k)$, the mixed estimation for model can be calculated as following.

$$\hat{x}^{0j}(k-1|k-1) = \sum_{i=1}^{r} \mu^{i|j}(k-1|k-1)\hat{x}^i(k-1|k-1) \qquad j=1,...,r \tag{2.56}$$

The mixed covariance can be calculated as:

$$\begin{aligned} P^{0j}(k-1|k-1) &= \sum_{i=1}^{r} \mu^{i|j}(k-1|k-1)\{P^i(k-1|k-1) \\ &+[\hat{x}^i(k-1|k-1) - \hat{x}^{0j}(k-1|k-1)]. \\ &[\hat{x}^i(k-1|k-1) - \hat{x}^{0j}(k-1|k-1)]'\}, \qquad j=1,...,r \end{aligned} \tag{2.57}$$

The term $\mu^{i|j}(k-1|k-1)$ in the above equations is called conditional model probability. Conditional model probability represents the conditional probability that the target translates from state i to state j. The conditional probability $\mu^{i|j}(k-1|k-1)$ is given by

$$\begin{aligned} \mu^{i|j}(k-1|k-1) &= P\{M^i(k-1)\,|\,M^j(k), Z^{k-1}\} \\ &= \frac{1}{\bar{c}^j} P\left\{M^j(k)\,|\,M^i(k-1), Z^{k-1}\right\} P\left\{M^i(k-1)\,|\,Z^{k-1}\right\} \end{aligned} \tag{2.58}$$

which can be rewritten as?

$$\mu^{i|j}(k-1|k-1) = \frac{1}{\bar{c}^j} p_{ij}\mu^i(k-1|k-1) \qquad i,j=1,...,r \tag{2.59}$$

where the normalizing constants are

$$\bar{c}^j = \sum_{i=1}^{r} p_{ij}\mu^i(k-1|k-1) \qquad j=1,...,r \tag{2.60}$$

Up to here mixing of the state estimates is done. In the second step model-conditioned updates are calculated. As in the KF, first the state prediction and covariance prediction is calculated. Then KF gain is calculated. Finally the state and the covariance are updated using the KF gain. These calculations are done for all of the filters matched to $M^j(k)$. For the filter matched to $M^j(k)$ the following equations can be written.

$$\hat{x}^j(k|k-1) = F^j(k-1)\hat{x}^{0j}(k|k-1) \tag{2.61}$$

$$P^j(k|k-1) = F^j(k-1)P^{0j}(k|k-1)[F^j(k-1)]' + Q(k-1) \tag{2.62}$$

$$v^j(k) = z(k) - H^j(k)\hat{x}^j(k|k-1) \tag{2.63}$$

$$S^j(k) = H^j(k)P^j(k \mid k-1)[H^j(k)]' + R^j(k) \tag{2.64}$$

$$W^j(k) = P^j(k \mid k-1)[H^j(k)]'[S^j(k)]^{-1} \tag{2.65}$$

$$\hat{x}(k \mid k) = \hat{x}(k \mid k-1) + W^j(k)v^j(k) \tag{2.66}$$

$$P^j(k \mid k) = [I - W^j(k)H^j(k)]P^j(k \mid k-1) \tag{2.67}$$

The above symbols denote same thing as in the KF. The superscript *j* "^{j}" denotes the filter matched to $M^j(k)$. The second step, model-conditioned updates, is finished now.

The third step is computing the model likelihoods. The likelihood functions corresponding to the *r* filters are:

$$\Lambda^j(k) = \frac{1}{\sqrt{\det[2\pi S^j(k)]}} \exp\left\{-\frac{1}{2}\left[v^j(k)\right]'\left[S^j(k)\right]^{-1}v^j(k)\right\} \tag{2.68}$$

The intermediate steps can be found in [1, 20].

The next step in the IMM-KF algorithm is model probability update. The probability $\mu^j(k \mid k)$ for $M^j(k)$ is

$$\mu^j(k \mid k) = P\{M^j(k) \mid Z^k\} = \frac{1}{c}\bar{c}^j\Lambda^j(k) \qquad j = 1,...,r \tag{2.69}$$

where the normalization factor *c* is

$$c = \sum_{i=1}^{r} \bar{c}^j\Lambda^i(k) \tag{2.70}$$

The last step for IMM-KF algorithm is combining the state and covariance estimations. The updated state and updated covariance is given as follows:

$$\hat{x}(k \mid k) = \sum_{j=1}^{r} \mu^j(k \mid k)\hat{x}^j(k \mid k) \tag{2.71}$$

$$P(k \mid k) = \sum_{j=1}^{r} \mu^j(k \mid k)\left\{P^j(k \mid k) + [\hat{x}^j(k \mid k) - \hat{x}(k \mid k)][\hat{x}^j(k \mid k) - \hat{x}(k \mid k)]'\right\} \tag{2.72}$$

2.5. Particle Filter

The main assumption of KF is linearity and Gaussianity. Extensions to the KF remove the dependence to linearity in a sub-optimal manner but it still assumes the Gaussianity. Particle filter is a method that is based on Bayesian framework. PF is also called Sequential Monte Carlo (SMC). It is a technique for using Bayesian filter by Monte Carlo simulation. Further information about SMC can be found in [23]. The basis of the method is to construct a sample-based representation of the entire pdf [21]. Each sample is weighted and these samples and weights are used to obtain estimates. The weight of a sample shows its significance. The estimation of the state is weighted sum of samples. These samples are called particle, so this approach as called particle filter. Each particle is also different copies of the state vector. PF is more computationally complex than KF. But being not dependent to any assumption like Gaussianity or linearity this makes PF useful in real life. In this section theoretical details are not given but the PF algorithm and its implementation is given. The theoretical background and detailed derivation of the PF can be found in [21, 22, 23, 24].

PF has 2 main steps: prediction and update similar with KF. In the prediction state, a state prediction is calculated for all particles according to the system model. When a measurement comes all particles have been weighted according to some weighting criteria. Then the state estimation can be found using one of the several methods. These methods will be discussed later in this section. PF diverges if small weighted particles are not eliminated. After each weight assignment a re-sampling step performed. In the re-sampling step, the particles with small weights are eliminated and the particles with higher weights are re-sampled. In the below PF steps are given.

1. **Initialization:** Generate $x_0^i \sim p_{x0}, \qquad i = 1,...,N$ where N is the number of particles. In the initialization step, particles are randomly generated. For example, the initial pdf can be Gaussian or any other distribution.

Remember that each particle denotes a state vector. The number of particles is an important point. As the number of particles goes to infinity, the posterior pdf approaches to true density.

2. **Measurement Update:** When a measurement is arrived, updated weights of particles are calculated using

$$w_t^i = w_{t-1}^i p(y_t \mid x_t^i) = w_{t-1}^i p_{e_t}(y_t - h(x_t^i)), \qquad i = 1, 2, ...N \tag{2.73}$$

The weights can be normalized as following.

$$w_t^i = \frac{w_t^i}{\sum_{j=1}^{N} w_t^j} \tag{2.74}$$

After weights are normalized, the state estimation can be calculated in several ways. First the state estimation can be calculated by taking the weighted sum of particles. The estimation can be formulated as:

$$\hat{x}_t = \sum_{i=1}^{n} w_t^i x_t^i \tag{2.75}$$

The second method is selecting the best particle. In this method the state estimation is the particle which has the maximum weight.

The third algorithm is weighted mean of particles in a window that is around the best particle (i.e. maximum weighted particle).

3. **Re-sampling**

As the number of iterations is growing, only a few particles will have an important weight and the rest of the particles will have negligible weights. This will cause degeneracy in the filter. To prevent this effect re-sampling is used. In re-sampling step, the particles those have negligible weights are eliminated and new particles are generated from the particles those have higher weights. This step is very important to prevent the filter from divergence. Without this step, as the simulation continues only a few

particles will have important weights and the others will be unimportant and at the end only one particle's weight would be 1 and the others would be 0.

There are several re-sampling algorithms in the literature and each has advantages and disadvantages as discussed in [23]. Some of the methods used for re-sampling are multinomial sampling [25], residual sampling [26, 27] and minimum variance sampling [28]. The pseudo-code algorithm for minimum variance sampling is given below.

Resampling algorithm

$[\{x_k^{j*}, w_k^j, i^j\}_{j=1}^N = RESAMPLE[\{x_k^i, w_k^i\}_{i=1}^N]$

- Initialize the CSW(Cumulative Sum of weights): $c_1 = w_k^1$
- FOR i=2:N
  - Construct CSW: $c_i = c_{i-1} + w_k^i$
- END FOR
- Start at the bottom of the CSW:i=1
- Draw a starting point: $u_1 \sim U[0, N-1]$
- FOR j=1:N
  - Move along the CSW: $u_j = u_1 + N^{-1}(j-1)$
  - WHILE $u_j > c_i$
    - $i = i+1$
  - END WHILE
  - Assign sample: $x_k^{j*} = x_k^i$
  - Assign weight: $w_k^j = N^{-1}$
- END FOR

4. Prediction

The prediction step is similarly with KF prediction step. In this step, a state prediction is calculated for all particles. Prediction of particles is given in the following equation.

$$x_{t+1}^{i} = F_t x_t^{i} + G_t u_t \qquad i = 1,...,N \quad (2.76)$$

5. Update time index *t=t+1* and go to Step 2.

Several tracking algorithms are discussed up to here. We assumed that we know the system model for all algorithms. In next section, several motion models are given.

2.6. Interacting Multiple Model for PF

Only one model is used in PF. Thus particle filter can not handle maneuvering targets. A similar approach described in section 2.4 is applied for PF. This approach is called Interacting Multiple Model for PF (IMM-PF). In this approach more than one kinematic model are used with different probabilities. The IMM-PF is a combination of standard IMM estimator and a standard particle filter. There are four stages in IMM-PF: mixing and interacting stage, particle filtering stage, mode probability update stage and state update stage.[41]

In the following algorithm the four stages of the IMM-PF are given. In the following algorithm *r* denotes the number of modes.

1. Mixing and Interaction Stage

In this stage the mixing probabilities with mode switching probability matrix p_{ij} and the initial mode probability.

$$\mu_{ij}(k-1 \mid k-1) = \frac{1}{c_j} p_{ij} \mu_j(k-1), \qquad i, j = 1,....,r \qquad (2.77)$$

where c_j are normalizing constants:

$$c_j = \sum_{i \in M} p_{ij} \mu_i (k-1), \qquad j = 1, ..., r \tag{2.78}$$

The initial condition for the model *j* is

$$\hat{x}_{0j}(k-1 \mid k-1) = \sum_{i=1}^{r} \mu_{i \mid j}(k-1 \mid k-1) \hat{x}_i (k-1 \mid k-1) \qquad j = 1, ..., r \tag{2.79}$$

Prior probability for the model *j* is

$$\begin{aligned} &\hat{p}_{0j}(x_{0j}(k-1 \mid k-1) / z(k-1)) = \sum_{i=1}^{r} \mu_{i \mid j}(k-1 \mid k-1) \{ p_i (k-1 \mid k-1) \\ &+ [\hat{x}_i (k-1 \mid k-1) - \hat{x}_{0j}(k-1 \mid k-1)]. \\ &[\hat{x}_i (k-1 \mid k-1) - \hat{x}_{0j}(k-1 \mid k-1)]' \} \qquad j = 1, ...r \end{aligned} \tag{2.80}$$

2. Particle Filter Stage

For every model, *N* particles s_j^l (*l*=1,…,N) are sampled from $\hat{p}_{0j}(x_{0j}(k-1 \mid k-1) / z(k-1))$.

One step prediction of the particle set $\hat{s}_j^l(k)$ $(l = 1, ..., N)$ is

$$\hat{s}_j^l(k) = F_{k,k-1}[k, j] s_j^l (k-1) + w(k, j) \tag{2.81}$$

where $F_{k,k-1}[k, j]$ is the state transition matrix for model *j* and *w(k)* is process noise. One-step prediction of the measurements is:

$$\hat{z}_j^l (k \mid k-1) = H_k (k, j) \hat{s}_j^l (k) \tag{2.82}$$

where $H_k(k, j)$ is measurement matrix for model *j*.

Calculate the probability weights with the innovation and measurement noise:

$$\overline{q}_j^l = d_{v(k,j)} \left(z(k) - \hat{z}_j^l (k \mid k-1) \right) \tag{2.83}$$

where $d_{v(k,j)}$ is density of measurement noise. Normalized weights are

calculated as:

$$q_j^l = \frac{\overline{q}_j^l}{\sum_{l=1}^{N} \overline{q}_j^l} \tag{2.84}$$

The state estimation of the model *m(k)* over the sample set is:

$$\hat{x}_j(k) = \sum_{l=1}^{N} q_l^j \hat{s}_j^l(k) \tag{2.85}$$

$$\hat{P}_j(k) = \sum_{l=1}^{N} q_j^l (\hat{s}_j^l(k) - \hat{x}_j(k))(\hat{s}_j^l(k) - \hat{x}_j(k))^T \tag{2.86}$$

3. Mode Probability Update Stage

The mode probability of the model j is updated with the innovation and its covariance from the model *j* is updated with the innovation and its covariance from the mode-matched filter corresponding to the model *j*. In this stage, the particle set is used to calculate the innovation and its covariance.

Residual covariance over the sample set is:

$$\hat{S}_j(k) = \sum_{l=1}^{N} q_j^l (\hat{z}_j^l - H_k(k, j)\hat{x}_j(k))(\hat{z}_j^l - H_k(k, j)\hat{x}_j(k))^T \tag{2.87}$$

Residual is:

$$r_j^l(k) = z(k, j) - \hat{z}_j^l \tag{2.88}$$

Likelihood function is:

$$\Lambda_j(k) = N(r_j^l; 0, \hat{S}_j(k)) \tag{2.89}$$

Mode probability update is:

$$\mu_j(k) = \frac{1}{c} \Lambda_j(k) c_j \tag{2.90}$$

where

$$c = \sum_{j=M} \Lambda_j(k) c_j \tag{2.91}$$

4. Estimate Update Stage

The state estimation stage is not used in the algorithm but it is used for output purposes. State estimation can be calculated as:

$$\hat{x}(k \mid k) = \sum_{j=1}^{r} \hat{x}_j(k \mid k) \mu_j(k) \tag{2.92}$$

2.7. Kinematic Models

In target tracking applications, the common step is prediction state which can be calculated according to a model. The kinematic model describes the motion of a target. Since we don't know what the target will do we use several kinematic models to predict the next state of the target.

In the following subsection 3 kinematic models are given. White Noise Acceleration Model (WNA), Wiener Process Acceleration Model (WPA) and Coordinated Turn Model (CTM). The first two models are linear and the last model is non-linear. For the CTM linearization of the non-linear model is also given.

2.7.1. White Noise Acceleration Model

Consider a dynamic linear model equation:

$$x(k+1) = Fx(k) + \Gamma v(k) \tag{2.93}$$

where F is state transition matrix and Γ is an n_x dimensional noise gain vector. The state variable *x* is defined as

$$x = [\xi \ \dot{\xi}]' \tag{2.94}$$

where ξ denotes the position and $\dot{\xi}$ denotes the velocity.

For WNA model the transition matrix F is given by

$$F = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \tag{2.95}$$

and noise gain vector is given by

$$\Gamma = \begin{bmatrix} \frac{1}{2}T^2 \\ T \end{bmatrix} \tag{2.96}$$

where T is the sampling period. The covariance of process noise multiplied by noise gain is calculated as follows

$$Q = E[\Gamma v(k) v(k) \Gamma'] = \Gamma \sigma_v^2 \Gamma' = \begin{bmatrix} \frac{1}{4}T^4 & \frac{1}{2}T^3 \\ \frac{1}{4}T^3 & T^2 \end{bmatrix} \sigma_v^2 \tag{2.97}$$

In the equation above, the value of σ_v can be chosen as $0.5a_M 0.5a_M \le \sigma_v \le a_M$. More information can be found in [1].

2.7.2. Wiener Process Acceleration Model

For WPA model the transition matrix and noise gain vector is defined as

$$F = \begin{bmatrix} 1 & T & \frac{1}{2}T^2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} \tag{2.98}$$

$$\Gamma = \begin{bmatrix} \frac{1}{2}T^2 \\ T \\ 1 \end{bmatrix} \tag{2.99}$$

The process noise multiplied with the noise gain is given by

$$Q = \Gamma \sigma_v^2 \Gamma' = \begin{bmatrix} \frac{1}{4}T^4 & \frac{1}{2}T^3 & \frac{1}{2}T^2 \\ \frac{1}{2}T^3 & T^2 & T \\ \frac{1}{2}T^2 & T & 1 \end{bmatrix} \sigma_v^2 \tag{2.100}$$

In the equation above the value of σ_v can be chosen as $0.5\Delta a_M \le \sigma_v \le \Delta a_M$ where Δa_M is acceleration change in a sample period *T*. More information can be found in [1].

2.7.3.Coordinated Turn Model

The models described previously are linear model for linear movement of a target. But generally the target makes maneuvers like turning. To predict like this target a turn model is needed. In this section a turn maneuver model called Coordinated Turn Model is discussed.

In the CTM a new variable is added to the state vector for angular turn rate. The augmented state vector is given by

$$x = \begin{bmatrix} x & \dot{x} & y & \dot{y} & \Omega \end{bmatrix}' \tag{2.101}$$

where (x, y) denotes the position in Cartesian coordinates, $(\dot{x}, \dot{y})$ denotes the corresponding velocities and Ω denotes turn rate.

Then the CTM is defined as follows.

$$\begin{aligned} x(k+1) = & \begin{bmatrix} 1 & \frac{\sin\Omega(k)T}{\Omega(k)} & 0 & -\frac{1-\cos\Omega(k)T}{\Omega(k)} & 0 \\ 0 & \cos\Omega(k)T & 0 & -\sin\Omega(k)T & 0 \\ 0 & \frac{1-\cos\Omega(k)T}{\Omega(k)T} & 1 & \frac{\sin\Omega(k)T}{\Omega(k)} & 0 \\ 0 & \sin\Omega(k)T & 0 & \cos\Omega(k) & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} x(k) \\ & + \begin{bmatrix} \frac{1}{2}T^2 & 0 & 0 \\ T & 0 & 0 \\ 0 & \frac{1}{2}T^2 & 0 \\ 0 & T & 0 \\ 0 & 0 & T \end{bmatrix} v(k) \end{aligned} \tag{2.102}$$

For more information about deriving CTM refer to [1].

As it can be seen from (2.102) the CTM model is non-linear. To be able to use it in tracking algorithms we must first linearize this model. The linearization of a non-linear model is given in section 2.3. The calculated Jacobian of f is given below according to [1].

$$f_x(k) = [\nabla_x f(k,x)']'\big|_{x=\hat{x}(k|k)} = \begin{bmatrix} 1 & \frac{\sin\hat{\Omega}(k)T}{\hat{\Omega}(k)} & 0 & -\frac{1-\cos\hat{\Omega}(k)T}{\hat{\Omega}(k)} & f_{\Omega,1}(k) \\ 0 & \cos\hat{\Omega}(k)T & 0 & -\sin\hat{\Omega}(k)T & f_{\Omega,2}(k) \\ 0 & \frac{1-\cos\hat{\Omega}(k)T}{\hat{\Omega}(k)T} & 1 & \frac{\sin\hat{\Omega}(k)T}{\hat{\Omega}(k)} & f_{\Omega,3}(k) \\ 0 & \sin\hat{\Omega}(k)T & 0 & \cos\hat{\Omega}(k) & f_{\Omega,4}(k) \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.103}$$

In the above $f_{\Omega,1}(k), f_{\Omega,2}(k), f_{\Omega,3}(k)$ and $f_{\Omega,4}(k)$ denotes partial derivatives w.r.t. Ω. In the below the calculation of $f_{\Omega,1}(k), f_{\Omega,2}(k), f_{\Omega,3}(k)$ and $f_{\Omega,4}(k)$ is given.

$$\begin{bmatrix} f_{\Omega,1}(k) \\ f_{\Omega,2}(k) \\ f_{\Omega,3}(k) \\ f_{\Omega,4}(k) \end{bmatrix} = \begin{bmatrix} \frac{(\cos\hat{\Omega}(k)T)T\hat{\dot{x}}(k)}{\hat{\Omega}(k)} - \frac{(\sin\hat{\Omega}(k)T)\hat{\dot{x}}(k)}{\hat{\Omega}(k)^2} - \frac{(\sin\hat{\Omega}(k)T)T\hat{\dot{y}}(k)}{\hat{\Omega}(k)} - \frac{(-1+\cos\hat{\Omega}(k)T)\hat{\dot{y}}(k)}{\hat{\Omega}(k)^2} \\ -\left(\sin\hat{\Omega}(k)T\right)T\hat{\dot{x}}(k) - \left(\cos\hat{\Omega}(k)T\right)T\hat{\dot{y}}(k) \\ \frac{(\sin\hat{\Omega}(k)T)T\hat{\dot{x}}(k)}{\hat{\Omega}(k)} - \frac{(1-\cos\hat{\Omega}(k)T)\hat{\dot{x}}(k)}{\hat{\Omega}(k)^2} + \frac{(\cos\hat{\Omega}(k)T)T\hat{\dot{y}}(k)}{\hat{\Omega}(k)} - \frac{(\sin\hat{\Omega}(k)T)\hat{\dot{y}}(k)}{\hat{\Omega}(k)^2} \\ \left(\cos\hat{\Omega}(k)T\right)T\hat{\dot{x}}(k) - \left(\sin\hat{\Omega}(k)T\right)T\hat{\dot{y}}(k) \end{bmatrix} \tag{2.104}$$

Now it is able to use the linearized model in EKF.

2.8. Performance Comparison of Tracking Algorithms for Aerial Vehicles

The performances of tracking algorithms described above are given by simulation results in this subsection. The simulation is based on tracking an airplane where the measurement are simulated using a simplified radar model. The algorithms are tested in several scenarios. Since the algorithms are probabilistic the results of simulations can be realistic after running the simulation for several times in the same environmental scenario. This method is called Monte Carlo simulation.

The simulations are based on the tool, Tafsim(Tracking and Fusion Simulator), which is developed by Murat Şamil ASLAN. More information about Tafsim can be found in APPENDIX A.

Three scenarios are tested for KF, EKF, IMM-KF, PF and IMM-PF. The results are plotted also. The plotted results include the true trajectory of the target, root mean square (rms) position error, root mean square velocity error and normalized position error. The calculation of terms used are given by

1)RMS Position Error:

$$RMS(\tilde{x}) = \sqrt{\tilde{x}^2 + \tilde{y}^2} \tag{2.105}$$

where position vector x = [x y]. The RMS error from N Monte Carlo runs with observed errors $\tilde{x}_i$ and $\tilde{y}_i$ in run i is:

$$RMS(\tilde{x}) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\tilde{x}^2 + \tilde{y}^2)} \tag{2.106}$$

2)RMS Velocity Error: Similarly, for the velocity $\dot{x} = [\dot{x}\ \dot{y}]$, the RMS velocity error is

$$RMS(\tilde{\tilde{x}}) = \sqrt{\tilde{\tilde{x}}^2 + \tilde{\tilde{y}}^2} \tag{2.107}$$

The RMS error from N Monte Carlo runs with observed errors $\tilde{\tilde{x}}_i$ and $\tilde{\tilde{y}}_i$ in run i is:

$$RMS(\tilde{\tilde{x}}) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\tilde{\tilde{x}}^2 + \tilde{\tilde{y}}^2)} \tag{2.108}$$

The comparison of tracking algorithms is given in following subsection for different scenarios.

3) Normalized Estimation Error Squared

The state estimation error is defined as follows.

$$\tilde{x}(k \mid k) = x(k) - \hat{x}(k \mid k) \tag{2.109}$$

Then the normalized estimation error square (NEES) is defined as:

$$\varepsilon(k) = \tilde{x}(k \mid k)' P(k \mid k)^{-1} \tilde{x}(k \mid k) \tag{2.110}$$

For N monte carlo simulations the average NEES is given as follows.

$$\bar{\varepsilon} = \frac{1}{N}\sum_{i=1}^{N}\varepsilon^i(k) \tag{2.111}$$

where $\varepsilon^i, i = 1,...,N$ denotes independent samples from monte carlo runs.

4) Normalized Innovation Squared

The normalized innovation squared (NIS) is defined as follows.

$$\varepsilon_v(k) = v(k)' S(k)^{-1} v(k) \tag{2.112}$$

where $v(k)$ denotes innovation and $S(k)^{-1}$ denotes innovation covariance.

2.8.1. Performance Comparison of KF and PF for Linear Model

This scenario aims to compare the performance of Kalman Filter with a basic Particle Filter (PF-50 with 50 particles and PF-500 with 500 particles) in aerial environment and constant velocity target motion. In this experiment the performance comparison of KF and PF is given for a linear motion. In this scenario the target moves with 20 m/s in x-axis and 10 m/s in y-axis. The target's initial position is (0,0). The model used for the filters is WNAM. The process noise variance is chosen as 1 and measurement noise variance is chosen as 10. In PF algorithm the number of particles is chosen as 50 and 500. The sampling period for the radar is 0.5 s. The simulation time is 150 seconds. The results are obtained for 100 monte carlo runs. The true trajectory of the target is given in Figure 2.4.

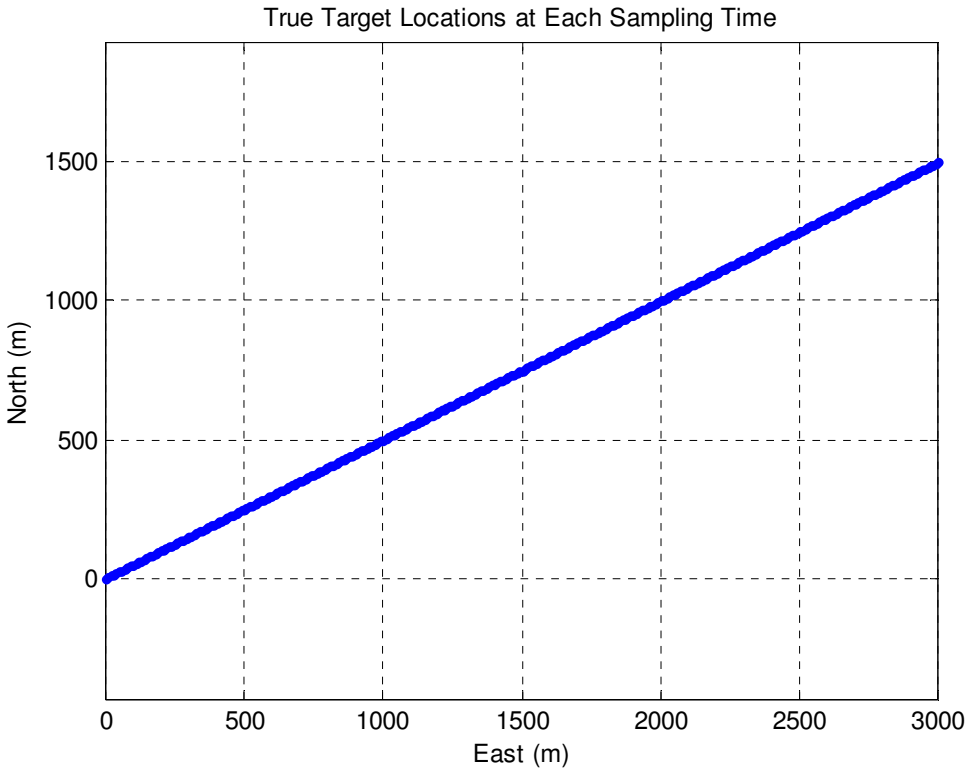

Figure 2.4 True Target Positions at Each Sampling Time

In the following three figures, we illustrate the tracking performance of the considered filters. In Figure 2.5 the RMS position errors, in Figure 2.6 the Normalized RMS position errors and in Figure 2.7 the RMS velocity errors are given for KF, PF-50 and PF-500.

As it can be seen from the Figure 2.5 performance of KF is better than PF-50 and PF-500. It is also seen from the figure that in the beginning of the simulation the errors are greater but as the simulation goes on the errors getting smaller. This is because the initial estimates are not accurate. Especially in the particle filter, the initialization is very important. The KF converges to steady state quicker but PF converges slower. Also as it can be seen from the figure that errors decreases by increasing the number of particles in the particles filter.

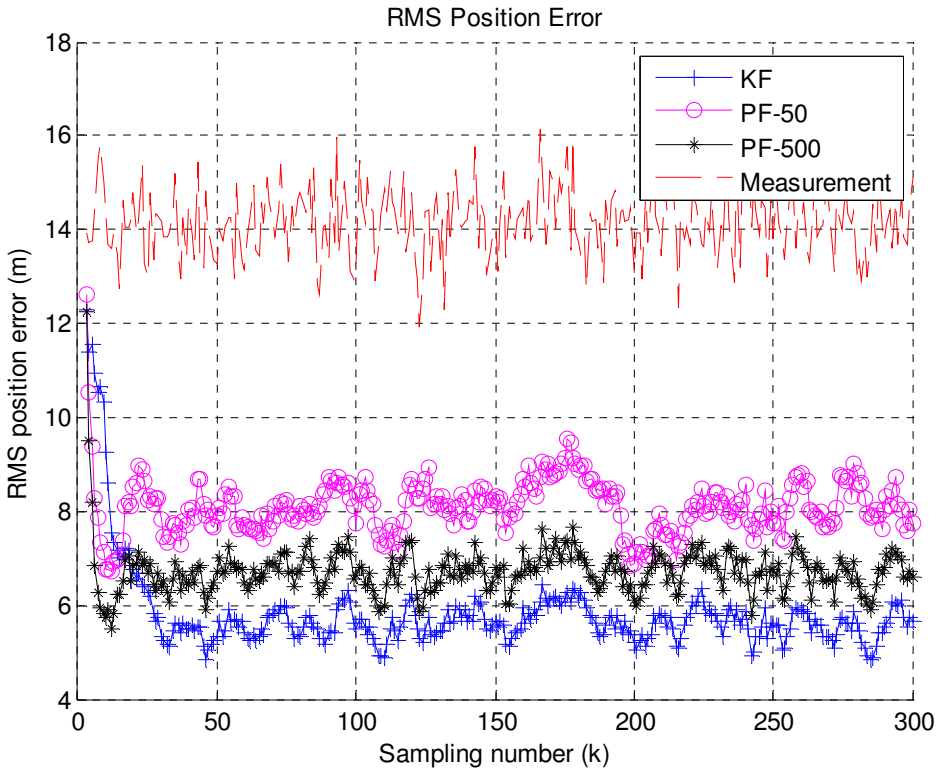

Figure 2.5 RMS Position Error

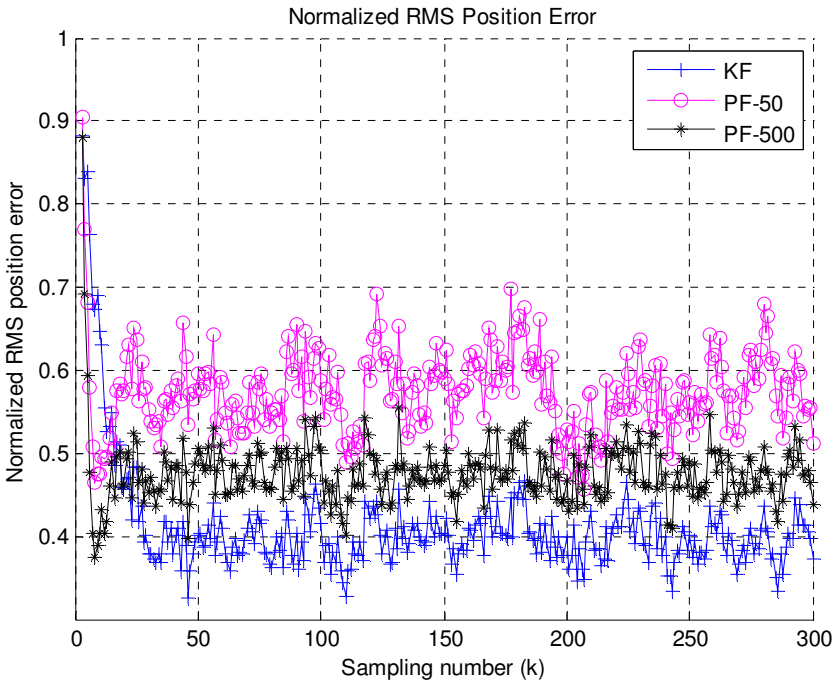

Figure 2.6 Normalized RMS Position Error

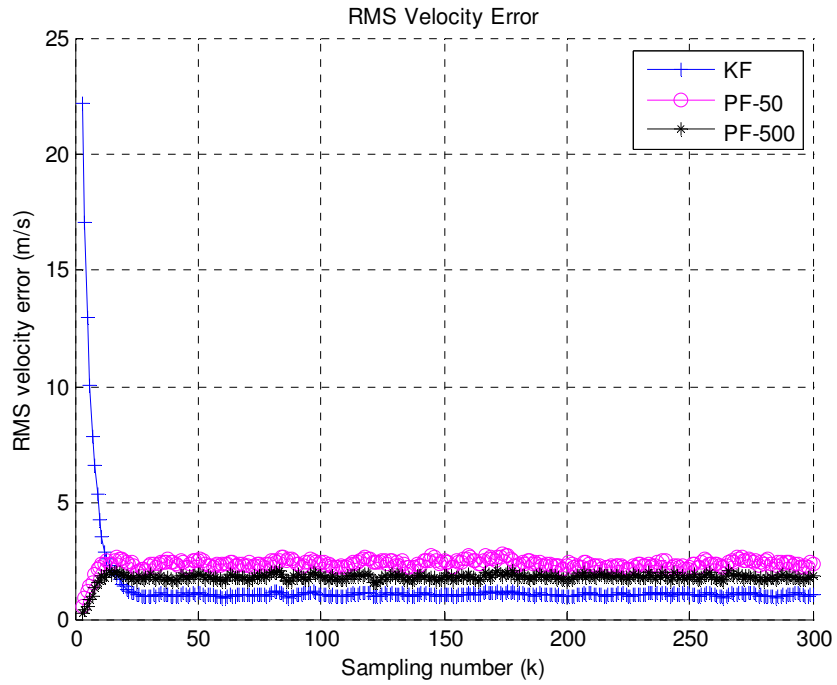

Figure 2.7 RMS Velocity Error

Performance vs. computation time comparison is given in Figure 2.8. It can be seen from the figure that the KF has the minimum average estimation error and computation time. The performance of the particle filter increases by increasing the number of particles but this also increases the computation time.

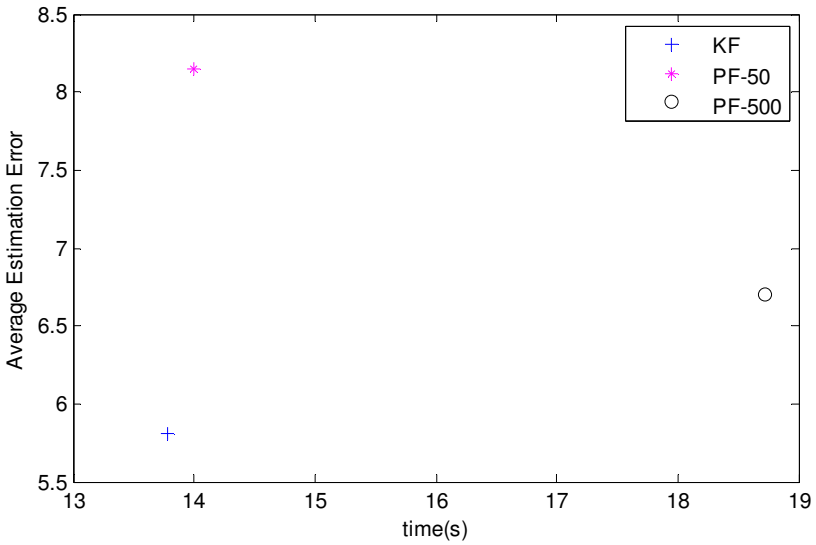

Figure 2.8 Performance vs. Time Comparison

2.8.2.Performance Comparison of KF, EKF and PF for non-linear Motion

This scenario aims to consider nonlinear (coordinated turn) motion and compare the performance of KF, EKF and PF-500 in aerial environment. In this scenario the performance comparison of KF, EKF and PF-500 are obtained. The target's initial position is (0,0). The target makes a nearly coordinated turn with turn rate around 1.5 degrees for 90 seconds. The omega greater than 0 implies a left turn and less than zero implies a right turn. Initial velocity along the x axis (east) is 5m/s. Since the motion model is non-linear the process noise variance of KF is chosen larger to handle the turn maneuver. In this scenario the process noise variance is chosen as 5 for KF, 1 for EKF and PF, and the measurement noise variance is chosen as 10. The number of particles is 500. The model that is used in both PF and EKF is CTM. The

model that is used in KF is WNAM. In EKF the linearized model (as given in equation (2.104)) of CTM is used. The number of Monte Carlo runs is 100. The sampling period for the radar is 0.5 s. In Figure 2.9 the true trajectory of the target is given. The subsequent figures present the tracking performance results for the cases of the three filters

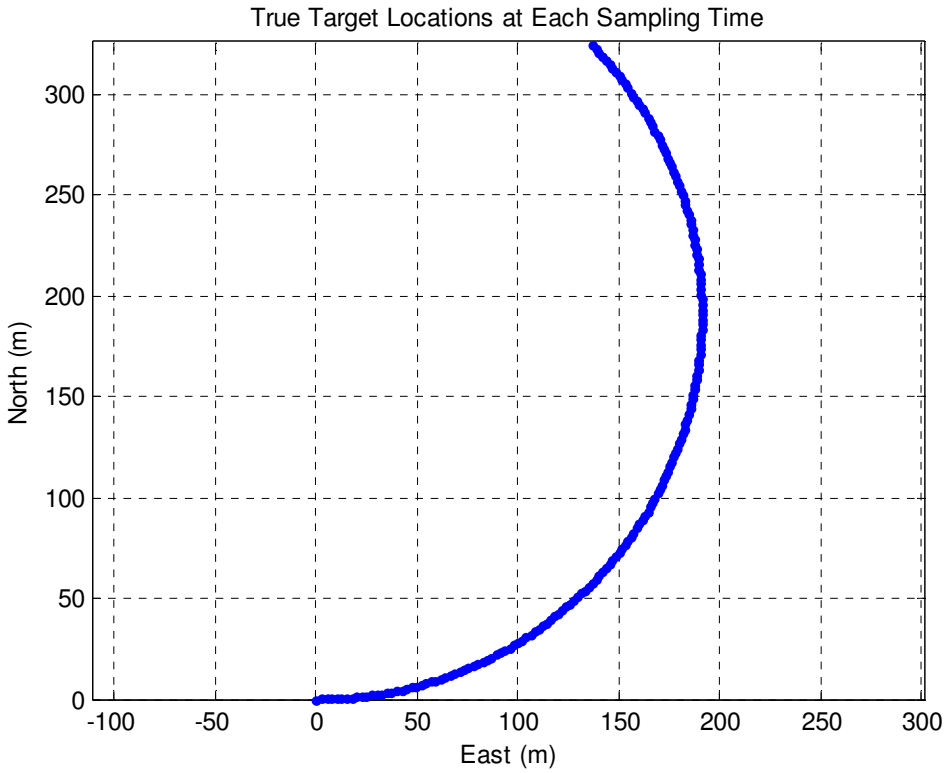

Figure 2.9 True Trajectory of Target

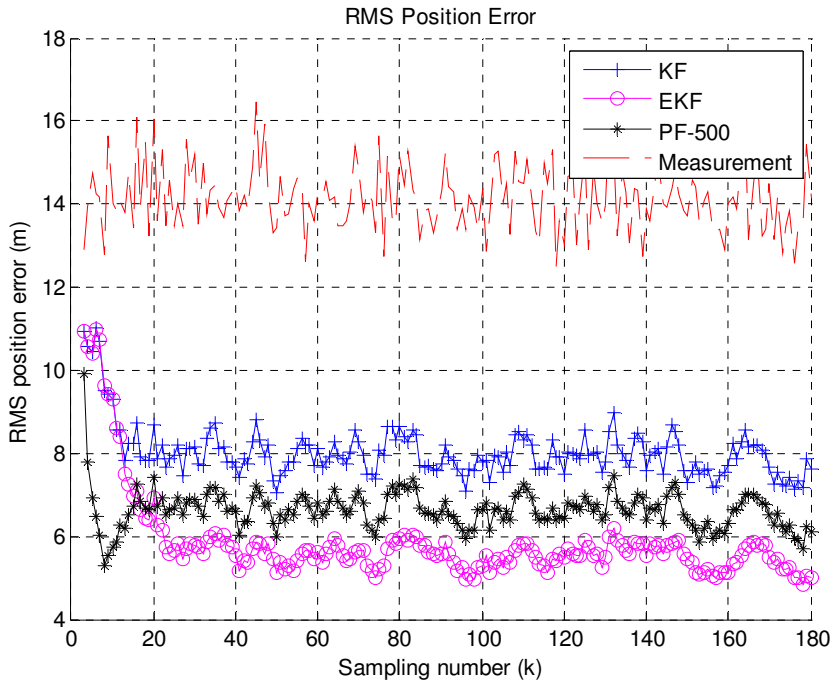

Figure 2.10 RMS Position Error

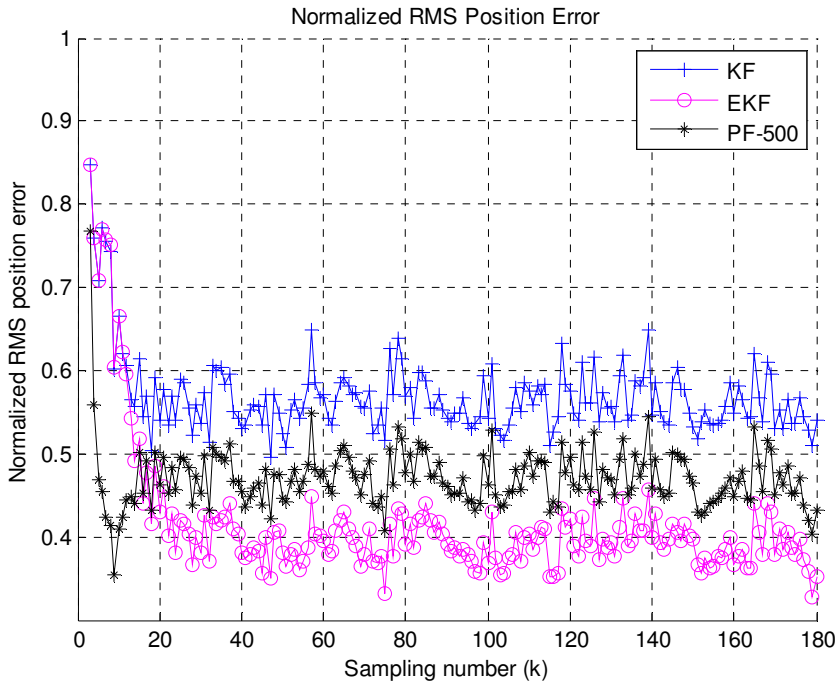

Figure 2.11 Normalized RMS Position Error

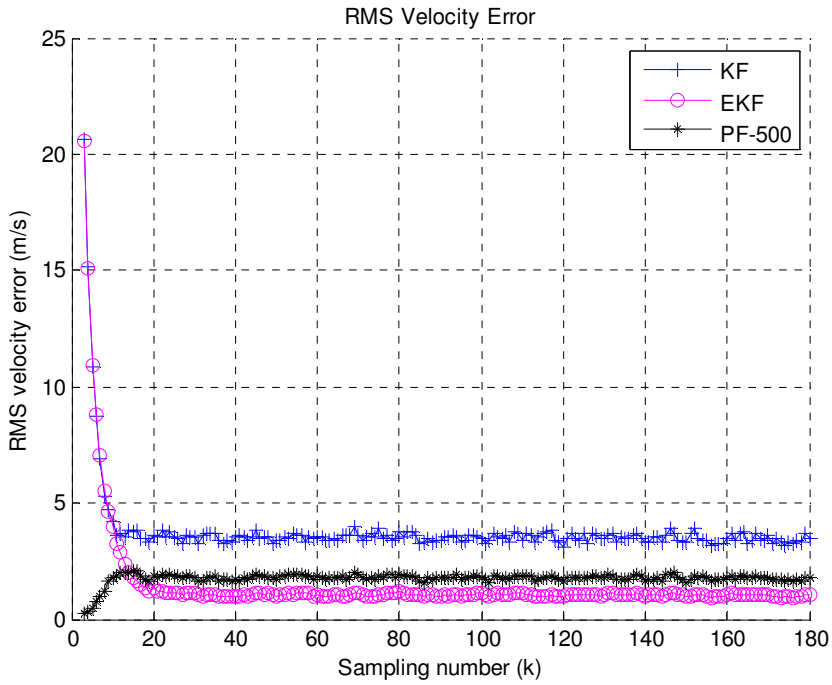

Figure 2.12 RMS Velocity Error

As it can be seen from the figures above the performance of the EKF is better than PF and KF. In this scenario the noise for both process and measurement is zero-mean white Gaussian noise. The performance of PF can be increased by increasing the number of particles.

In Figure 2.13 the average estimation error vs. computation time graph is given. The KF has the minimum computation time but its performance is the worst. In this case EKF is the best choice. Its computation time is much more than KF but it performance is the best. The performance of the PF is similar to EKF but its computation time is much more than EKF.

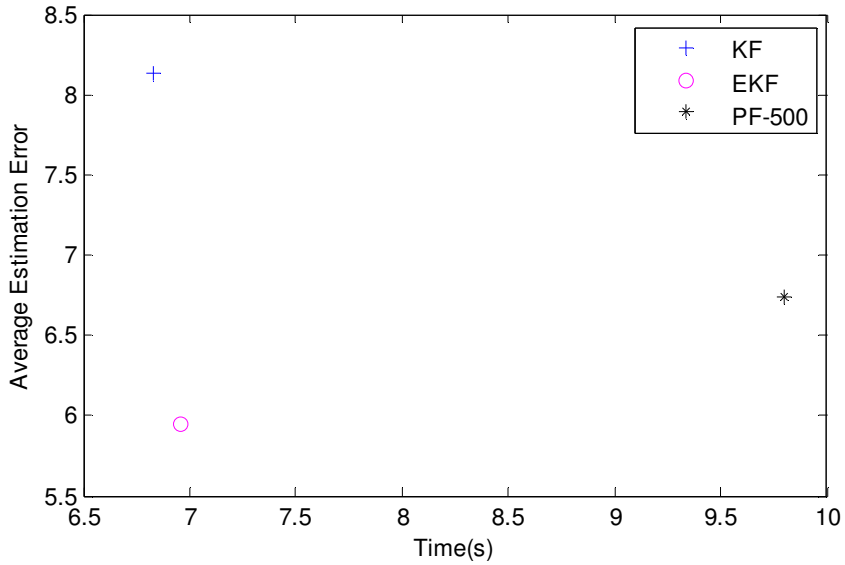

Figure 2.13 Performance vs. Time Comparison

2.8.3.Performance Comparison of IMM-KF and IMM-PF for a Maneuvering Target

In this scenario we compare the performances of IMM-KF (which in fact includes KF and EKF), IMM-PF-100 and IMM-PF-500. The target is initially located at (0,0).The target initially moves linearly in the y-axis. The target then switches to the near coordinated turn motion which is followed again by a linear portion. The parameters of the target motion are as follows: the target moves linearly along the y-axis (north) for 60 seconds with a speed of -75m/s, then makes a near coordinated turn at a turn rate of 3 degrees for 40 seconds and then moves linearly again for 50 seconds. Since multiple modes of motion are present, multiple models are present in the IMM filters. For the KF/EKF based IMM filter, KF with WNAM is used to model the linear part of the motion while EKF with CTM is used to model the turn and the IMM structure handles the mixing of the two models. For the PF based IMM filter, two models (again WNAM and CTM) with identical particle numbers are utilized. We have experimented with 100 (IMM-PF-100) particles and 500 (IMM-PF-500) particles. The process noise variance for both the WNAM and CTM models are taken as 4. The measurement noise variance is 50. The process noise

and measurement noise is zero-mean white Gaussian. In Figure 2.14 the true trajectory of the target is given. The subsequent figures present the tracking performance results for the cases of the three filters for 100 monte-carlo runs.

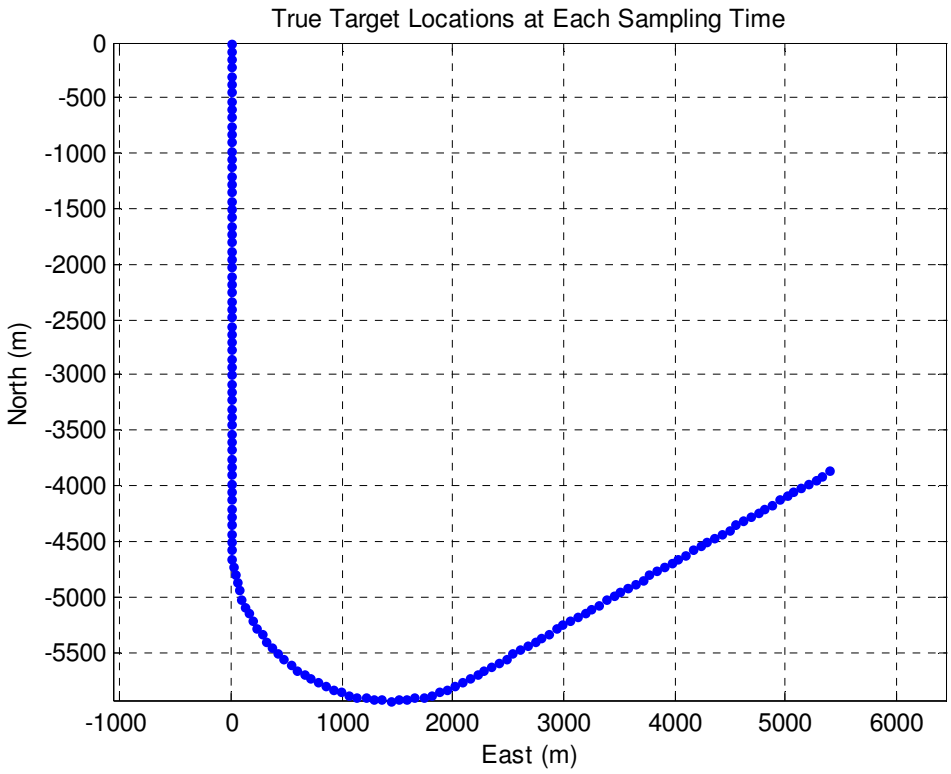

Figure 2.14True Trajectory of The Target

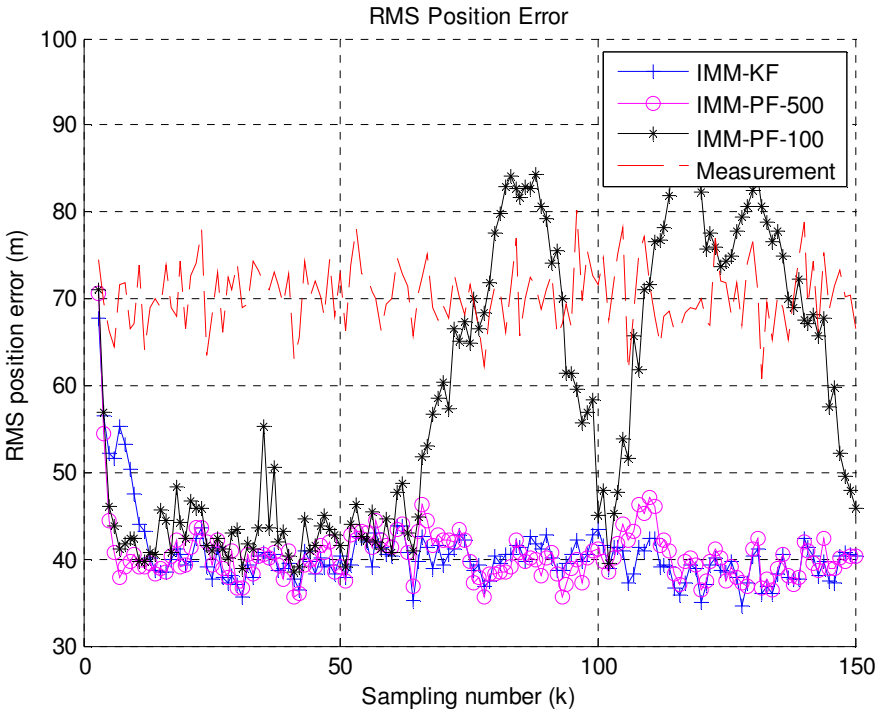

Figure 2.15 RMS Position Errors

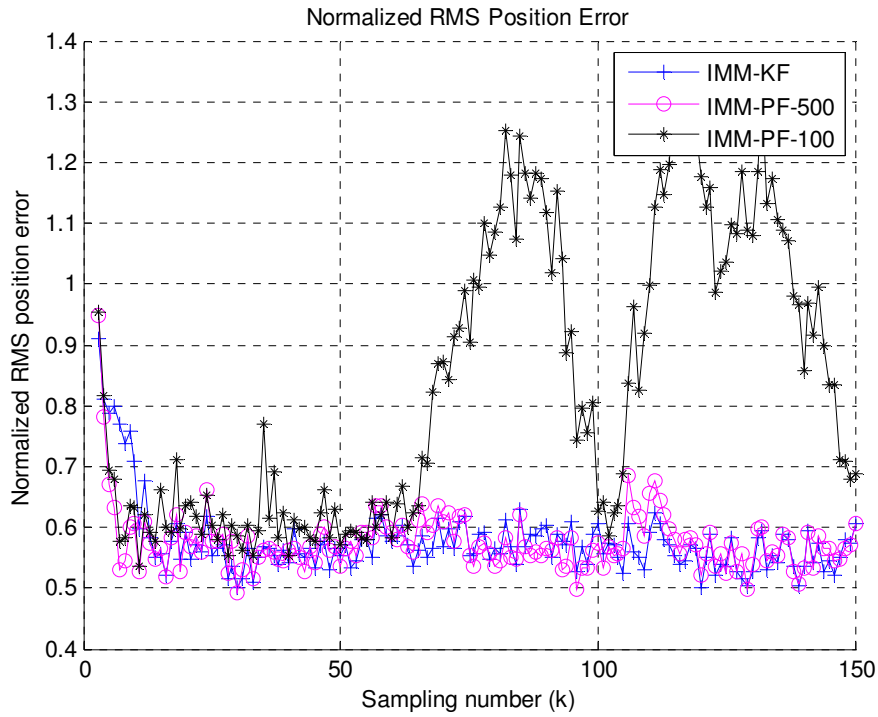

Figure 2.16 Normalized RMS Position Errors

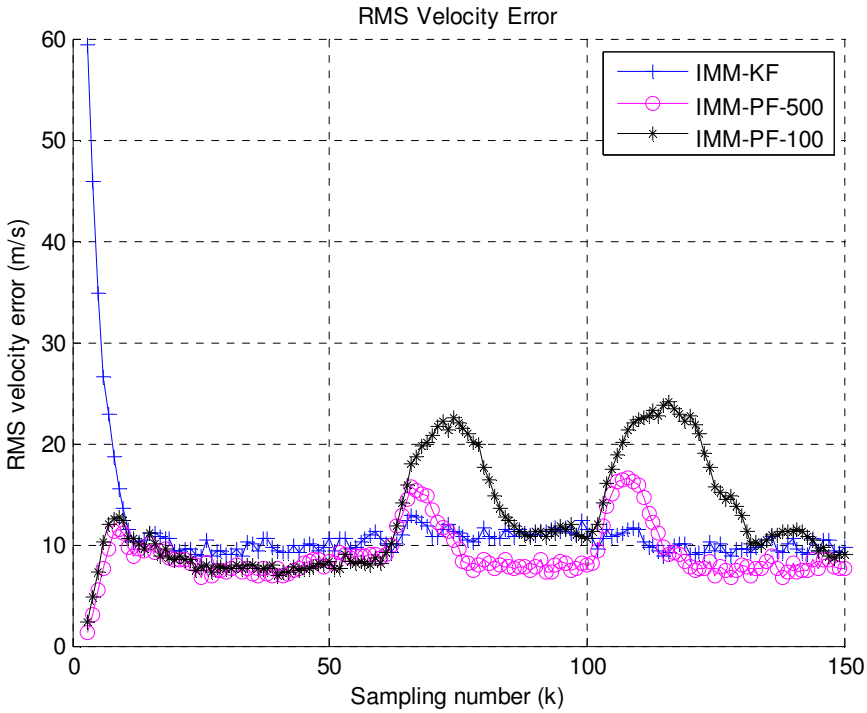

Figure 2.17 RMS Velocity Errors

In both the position and velocity errors, the characteristic adaptation of the IMM filter can easily be observed. Here, the estimation errors temporarily increase while the target changes its dynamics and the IMM filter adjusts the model probabilities after which the error is again brought down. Both the steady state performance differences as well as performance during these transitions can be inspected to compare two filters. As it can be seen from the figures above the performance of IMM-KF and IMM-PF-500 is very close to each other. It is obvious that the performance of IMM-PF will be better by increasing the number of particles.

In Figure 2.18 the mode probabilities of IMM-KF are given. In this figure WNAM model is denoted as ‘Model 1’ and CTM model is denoted as ‘Model 2’. The model probabilities denote which model is more reliable at current time step.

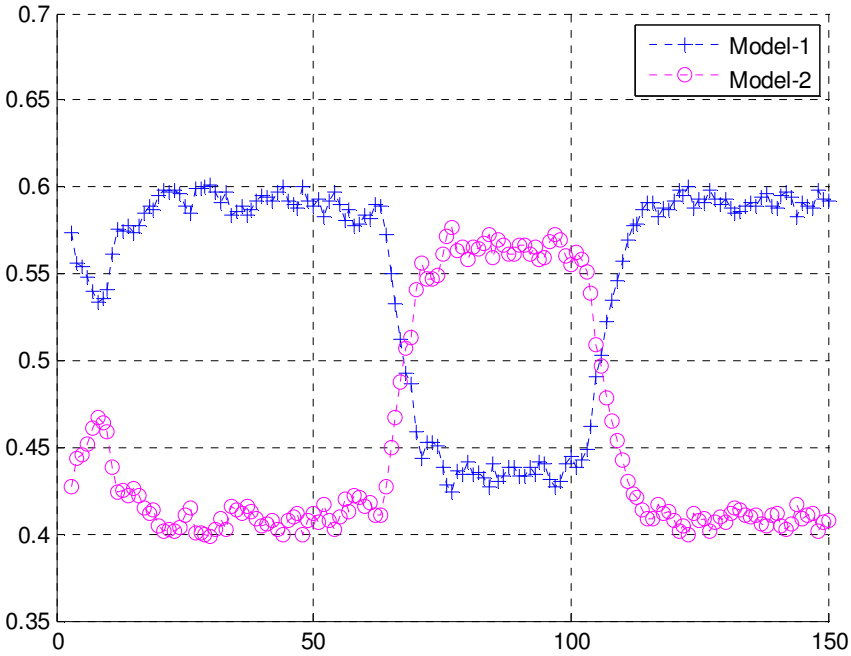

Figure 2.18 IMM-KF Model Probabilities

In Figure 2.19 the mode probabilities of IMM-PF-100 are given. In this figure WNAM model is denoted as 'Model 1' and CTM model is denoted as 'Model 2'.

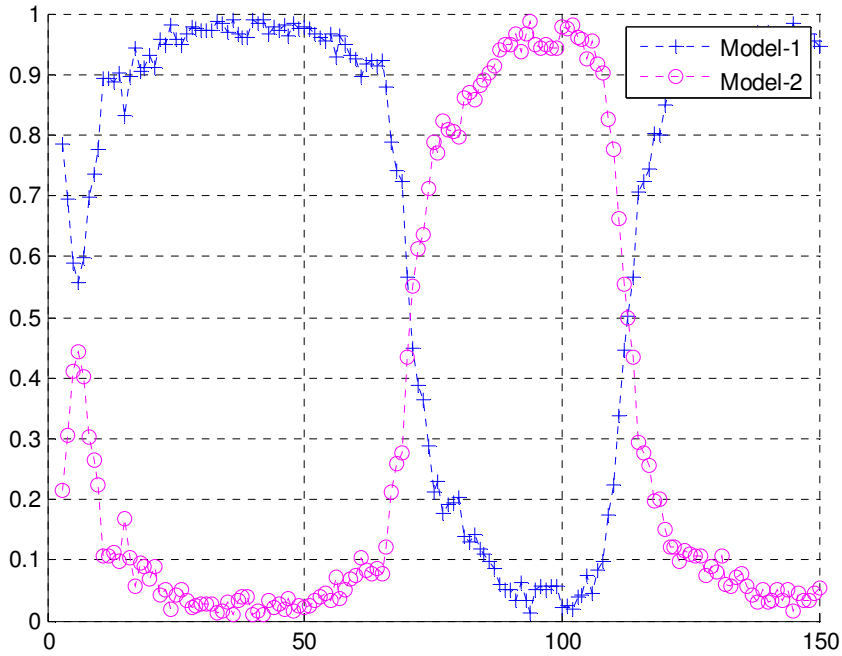

Figure 2.19 IMM-PF-100 Model Probabilities

In Figure 2.20 the mode probabilities of IMM-PF-500 are given. In this figure WNAM model is denoted as 'Model 1' and CTM model is denoted as 'Model 2'.

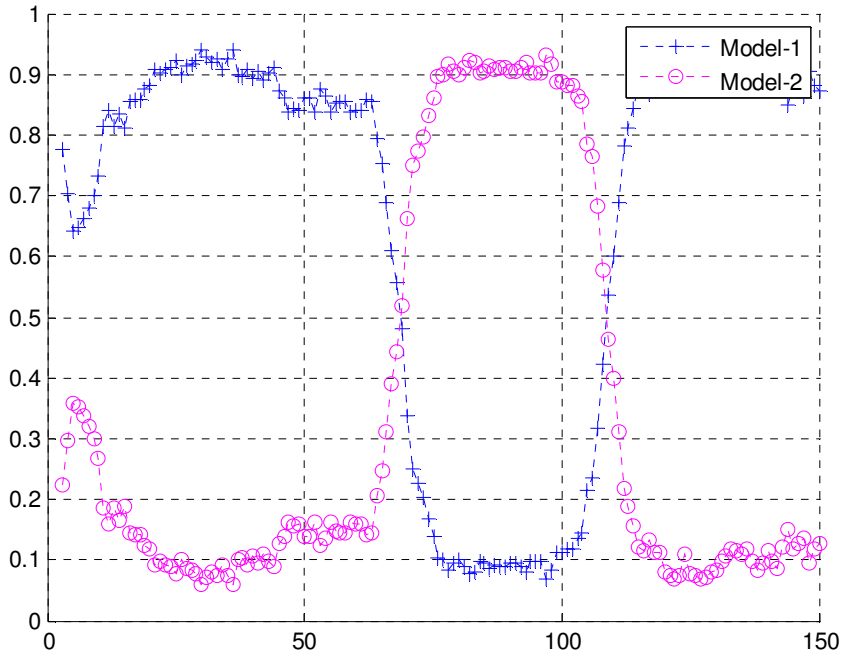

Figure 2.20 IMM-PF-500 Model Probabilities

In IMM there are two models. For linear motion the WNAM model has high probability. When the target starts coordinated turn the probability of WNAM starts decreasing and the probability of the CTM starts increasing as given in figures above.

In Figure 2.21 the average estimation error vs. computation time graph is given. The IMM-KF and IMM-PF-500 has very close average estimation error but the computation time of IMM-PF-500 is much greater than IMM-KF.

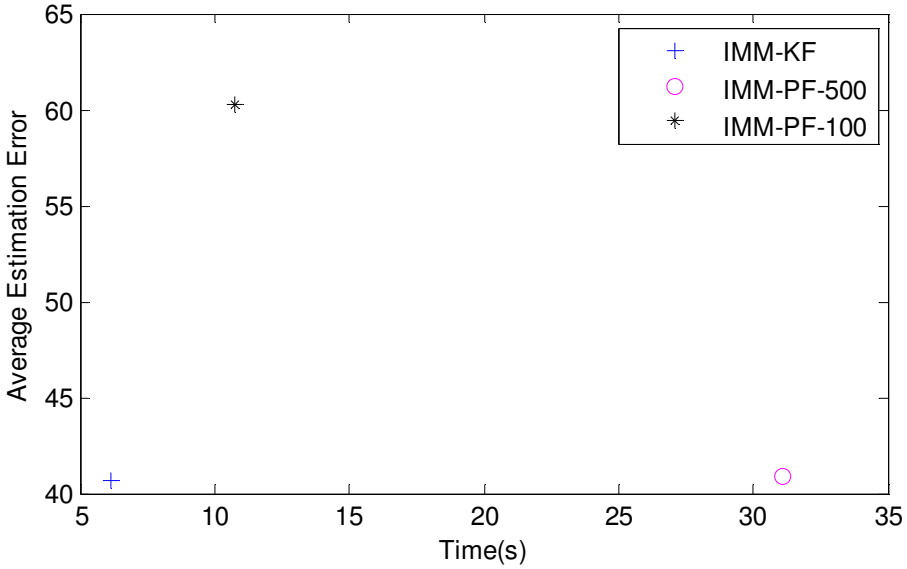

Figure 2.21 Performance vs. Time Comparison

CHAPTER 3

ACOUSTIC MODELLING

3.1. Introduction

In this chapter acoustic modeling is described for the purpose of generating simulated sonar observations. Acoustic modeling includes the mathematical modeling of underwater effects like the propagation of sound waves within layers of different temperature and salinity, underwater noise sources, reverberation as well as other effects. To obtain a faithful mathematical model for all underwater effects is a vast research area and we do not have any claim to have exhaustively covered the field. There are many physical uncertainties in an underwater environment. For example underwater currents, temperature differences, salinity differences, sea surface waves, shape of the sea bottom have fundamental effects in how the sound waves propagate and interact with the environment. It is very difficult to attack and model all these effects at once in acoustic modeling. Therefore, we follow an incremental approach and try to capture in our model some interesting and basic phenomena while trying to keep the complexity under control. This results in some assumptions and simplifications.

In underwater observation systems, the most common and widely used sensor is the Sonar. Sonar uses sound waves to measure the location (and sometimes speed) of

an underwater object. In underwater environment, the radar sensor can not be used due to the fact that high frequency electromagnetic waves used in radar systems are attenuated severely in this medium. Therefore it is not feasible to use electromagnetic waves for measurement for any practical application. The alternative to the electromagnetic wave is sound waves. Sound waves are faster and travel much larger distances in underwater medium as compared to the air due to the high density of water.

At the beginning of this chapter we start by reviewing sound propagation models. This is followed by a description of the noise model for underwater environment. Then the reverberation phenomenon is described. In the first three subsections of the present chapter, models used in the literature are described and their advantages and disadvantages are discussed. Then the sonar model that we have used in this study is described. Finally the simulation results for this sonar model are given to illustrate the behavior of the model.

A frequently used signal strength unit in acoustic modeling is decibel (dB). Decibel is used to compare the relative strengths of two signals in a logarithmic scale, specifically in this case the power at the source and at the receiver. At this point it is useful to note the definition of dB as given below.

$$Power\ gain = 10\log_{10}\left(\frac{P_{source}}{P_{receiver}}\right) \tag{3.1}$$

3.2. Propagation of Sound in Underwater Environment

As discussed in the introduction of this chapter the underwater environment is composed of the very complex interaction of many physical phenomena that is effecting the propagation of sound. This is a very dispersive medium due to sea floor, sea bottom internal waves and other effects. When sound waves travel in this

medium, the wave shape is distorted significantly. The amplitude of traveling sound wave attenuates and phase delays are introduced. The attenuation of the magnitude of sound wave is expressed by the term transmission loss (TL) [29]. The transmission loss defines a fundamental one of many effects of the considered underwater environment and will be described in more detail in the following section.

3.2.1.Transmission Loss

The transmission loss is defined as a quantitative measure of the attenuation of the sound amplitude between a point 1 meter from sound source center and a receiver placed at a variable distance from that point [29, 30, 31]. Let I_s denote the intensity of sound at a point 1 meter from the center of sound source and I_r is the intensity at the receiver. The intensity is defined as the energy which is carried by the sound wave per second (power) crossing a unit area[29]. The formulation of intensity is given as following.

$$I = \frac{p^2}{\rho c} \tag{3.2}$$

where p denotes pressure (in Pa or N/m^2), ρ denotes water density and c denotes the speed of the sound in the considered medium.

Then TL is defined as following.

$$TL = 10 \log \frac{I_s}{I_r} \text{ (dB)} \tag{3.3}$$

The TL can be considered as the sum of losses due to spreading and absorption. The spreading and attenuation losses are discussed shortly in the following sub sections. Further information can be found in [29, 30].

3.2.1.1. Spreading Loss

The spreading of sound can be divided into two categories: Spherical spreading and cylindrical spreading. The most basic spreading form is spherical spreading. In

spherical spreading it is assumed that the sound sources are located in an unbounded (i.e. the sea floor and sea surface are ignored) and lossless medium. With these assumptions the spreading loss can be calculated as the following: Let r be the distance of the receiver to sound source in meters. Then SL can be calculated as following.

$$SL = 20\log r \tag{3.4}$$

The main assumption of the spherical spreading is that the medium is unbounded. If the medium is bounded than cylindrical spreading model is used. Consider that the medium is bounded from upper and lower by two parallel planes which have a distance H (in meters) between them. Again let r (in meters) is the distance of the point to the sound source. Then SL due to cylindrical spreading is given by

$$SL = 10\log r \tag{3.5}$$

The detailed information about deriving the formulations presented above can be found in [29].

3.2.1.2. Absorption Loss

The main causes of absorption of a spreading sound wave are viscosity and molecular relaxation. These effects are described in detailed in [29]. In this study only the resulting model of these effects is given. Let a denote attenuation coefficient which gives the total absorption loss in the terms of *dB/km*. As an approximation the attenuation coefficient for standard sea water can be calculated by the following equation.

$$a = 0.05 f^{1.4} \tag{3.6}$$

where f is frequency of the sound waves in kHz.

Therefore, the total transmission loss TL is the sum of spreading loss (SL) and absorption loss (a). The TL can be calculated as the sum of spherical spreading loss and absorption loss and is given by the following

$$TL = 20\log r + ar\times 10^{-3} \qquad \text{(dB)} \qquad (3.7)$$

3.2.2. Speed of Sound

The speed of sound waves in underwater is one of the fundamental elements that affect the propagation of sound wave in this medium. The speed of sound depends on the magnitude of three main factors: the temperature, the pressure (depth) and the salinity of the water [29]. There are several calculation equations for speed of sound. Each one has its own applicable ranges and its own standard errors. In references [32, 33, 34, 35, 36] some of the developed speed calculation approaches are given each with different number of terms and complexity. The most basic equation which has 6 terms is presented in [36], and the most complex equation which has 23 terms is given in [32]. In the present study we use the speed of sound calculation equation that is given by [33]. This is the simplified version of the equation in [32] and is given by the following equation:

$$\begin{aligned} c = {} & 1492.9 + 3(t-10) - 6\times 10^{-3}(t-10)^2 - 4\times 10^{-2}(t-18)^2 \\ & + 1.2(s-35) - 10^{-2}(t-18)(s-35) + \frac{h}{61} \end{aligned} \qquad (3.8)$$

where we have

c = Speed of sound (m/s)

t = temperature (˚C)

s = salinity (ppt)

h = depth (m)

In Table 3.1, the ranges of parameters for which the equation (3.8) is applicable are summarized.

Table 3.1 Speed of Sound Formulation Parameter Ranges

Temperature Range (˚C)	Salinity Range (ppt)	Depth Range (m)	Standard Error (m/s)
-2 to 24.5	30 to 42	0 to 1000	0.2

3.2.3.Propagation Models

The propagation of the sound waves in underwater environment is described mathematically by solutions of the wave equation. All propagation models that are used in acoustic modeling are based on the wave equation. The derivation of wave equation can be found in [37]. The wave equation itself is given by

$$\frac{\partial^2 p}{\partial t^2} = c^2 \left(\frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} + \frac{\partial^2 p}{\partial z^2} \right) \tag{3.9}$$

where c is the speed of sound, p is acoustic pressure and x, y, z are the coordinates. The propagation of the sound is described by solving equation(3.9). There are several approaches for solving wave equation. These approaches are: ray theory, normal mode, multi-path expansion, fast field and parabolic equation techniques. In the following sub-sections these techniques are explained. More information about the theory of these methods can be found in [29, 30, 31].

3.2.3.1.Ray Theory

In ray theory sound waves are indicated as a ray starting from the source and propagating on a linear trajectory. The transmission loss, TL, is computed during the "tracing of the ray". Ray theory does not provide a good solution when the frequency of the sound wave is low or the pressure change is significant over the distance of one wavelength [29]. The detailed formulation of ray theory can be

found in [29, 30]. In ray theory the transmission loss is accurately modeled as a function of frequency, temperature, salinity and depth and it changes continuously during the tracing of ray (along the linear trajectory of wave propagation).

One of most important result of ray theory is Snell's law and is also extensively used for light propagation models. This law describes the refraction of sound waves between two different medium where the speed of sound differs. The refraction of a sound wave between different layers of underwater medium is illustrated in Figure 3.1. Each layer in the water is distinguished by a boundary line and the sound wave refracts when passing from one medium to another as a function of the incidence and exit angles and the velocities in the two medium.

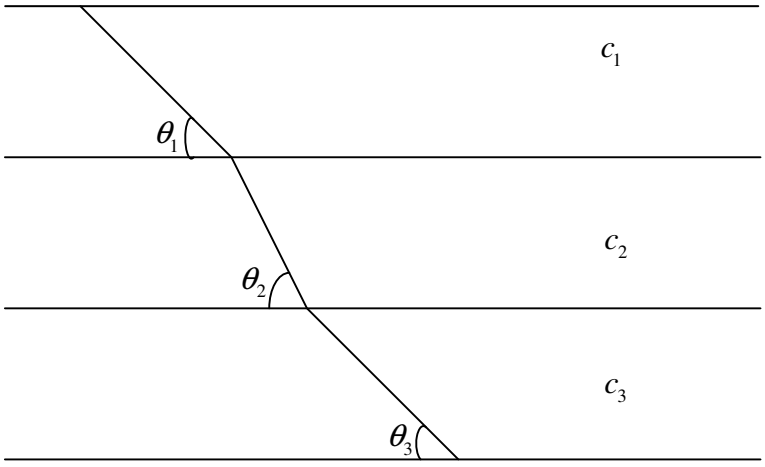

Figure 3.1 Refraction of Sound Between Different Layers

In the above figure c_1, c_2, c_3 the sound speeds in three different layers and the angles are the three propagation angles with the horizontal. Then the Snell's law is written by

$$\frac{\cos\theta_1}{c_1} = \frac{\cos\theta_2}{c_2} = \frac{\cos\theta_3}{c_3} \tag{3.10}$$

In case where arbitrary boundaries are present, the law should be written for each interface in terms of the incidence and exit angles with the boundary normal. For our experiments we are assuming that the assumptions for the validity of ray theory hold and mainly use ray theory to model sound propagation in multi-layer scenarios. The ray theory can also be adopted to handle the reflections from sea bottom and sea surface but these are considered out of the scope of the present work.

3.2.3.2.Normal Mode

In normal mode the propagation of sound waves in underwater medium is characterized by normal modes which are used for the solution of the wave equation. One of the normal mode equations is depth equation which describes the standing wave portion of solution and the other is called range equation which describes the traveling wave portion of the solution [30]. We are not further elaborating on this approach since it is not adopted in the present study.

3.2.3.3.Other Propagation Models

In the previous subsections ray theory model and normal mode models are described. These two are the most commonly used models. The remaining propagation models are multipath expansion models, fast field models and parabolic equation models. The detailed theory of these models can be found in [30] but will be briefly described here for the sake of completeness.

In [30] the applicable usage areas of all of these models are summarized. In the Table 3.2 the models and their domains of applicability are given.

Table 3.2 Domains of Applicability of Underwater Propagation Models

Model Type	Applications							
	Shallow Water				Deep Water			
	Low Frequency		High Frequency		Low Frequency		High Frequency	
	RI	RD	RI	RD	RI	RD	RI	RD
Ray Theory	-	-	⊥	+	⊥	⊥	+	+
Normal Mode	+	⊥	+	⊥	+	⊥	⊥	-
Multipath Expansion	-	-	⊥	-	⊥	-	+	-
Fast Field	+	-	+	-	+	-	⊥	-
Parabolic Equation	⊥	+	-	-	⊥	+		⊥

Low Frequency (<500 Hz) RI: Range Independent environment
High Frequency (>500 Hz) RD: Range-dependent Environment
\+ This model is both applicable and practical
⊥ There are limitations on the performance of model
\- Neither applicable nor practical

As discussed above, the propagation of the sound in the underwater environment is not straight forward. While sound propagates its speed and direction changes, it is reflected from sea bottom and sea surface. In Figure 3.2 the propagation of a sound wave is demonstrated. The figure is created using the Rayson software [38] which is an underwater communications simulator based on ray theory acoustic propagation model. In the figure the sound source is located at depth 400m. In the figure there are 4 lines each has a different ray angle. As it can be seen from the figure the propagation of the sound in underwater medium is considerably complicated due to underwater effects. The sea surface and sea bottom reflections are also seen from the figure.

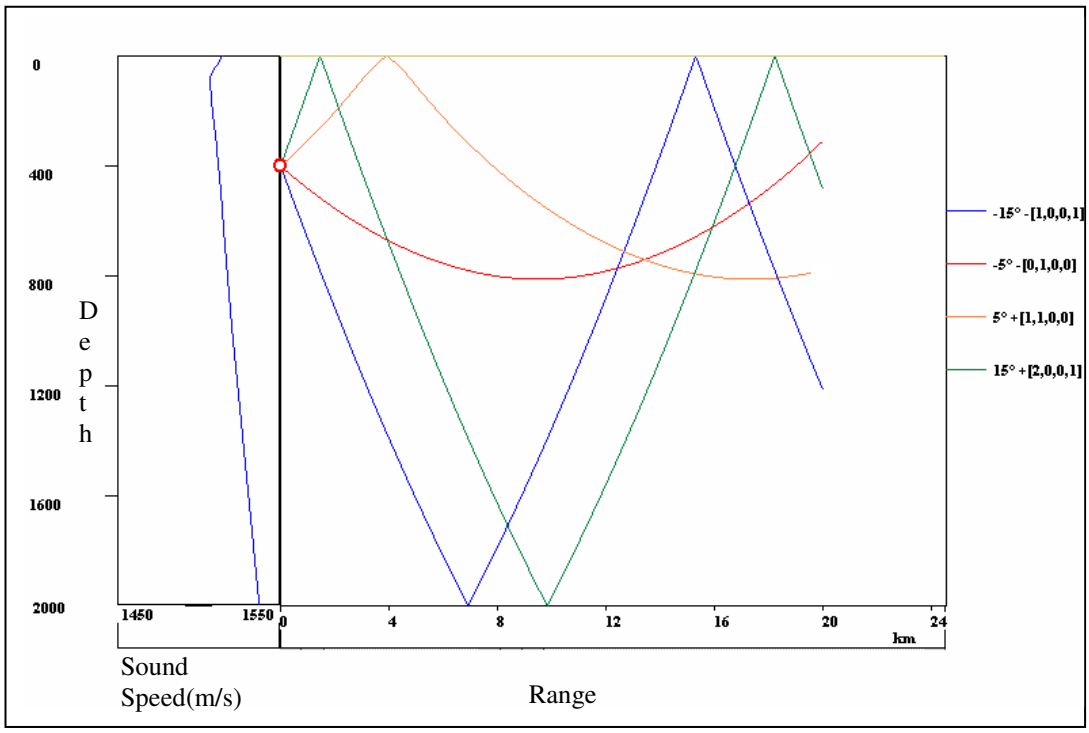

Figure 3.2 Propagation of Sound Wave in Underwater Environment. Output of the Rayson underwater communications software[38].

3.3. Noise

The sea is a chaotic environment. There are many noise sources in the sea itself in addition to the transducer noise. In sonar measurement even if the sonar is designed perfectly (i.e. there is not any noise generated by sonar itself) the errors will be high from noise generated by the sea physical effects. When using sonar there are three main types of noise sources including the noise generated by the sonar transducer and electronics. The noise sources can be categorized as:

1. Thermal noise
2. Ambient noise
3. Vessel noise

These noise types are discussed in the following subsections.

3.3.1.Thermal Noise

The thermal noise is the noise that is generated by sonar itself. An underwater transducer energy exchanges take place with the sea through motional resistance of the transducer. This energy exchange results thermal noise. The thermal noise comes from the thermal agitation of the molecules of water producing pressure fluctuations at the face of the hydrophone. Hydrophone has a similar meaning with antenna that is used in radar.

The thermal noise is also a problem for radars. All the research for the thermal noise in radars is also applicable for sonar. A general approximation for finding thermal noise of sonar is formulated below [31].

$$N_{thermal} = -15 + 20\log f \tag{3.11}$$

where $N_{thermal}$ is in dB and f is in KHz. Thermal noise can only be important when the sonar operating frequency is above 30 KHz but practically the thermal noise effect is shown when operating at least 100 KHz. The thermal noise spectrum is given in Figure 3.3.

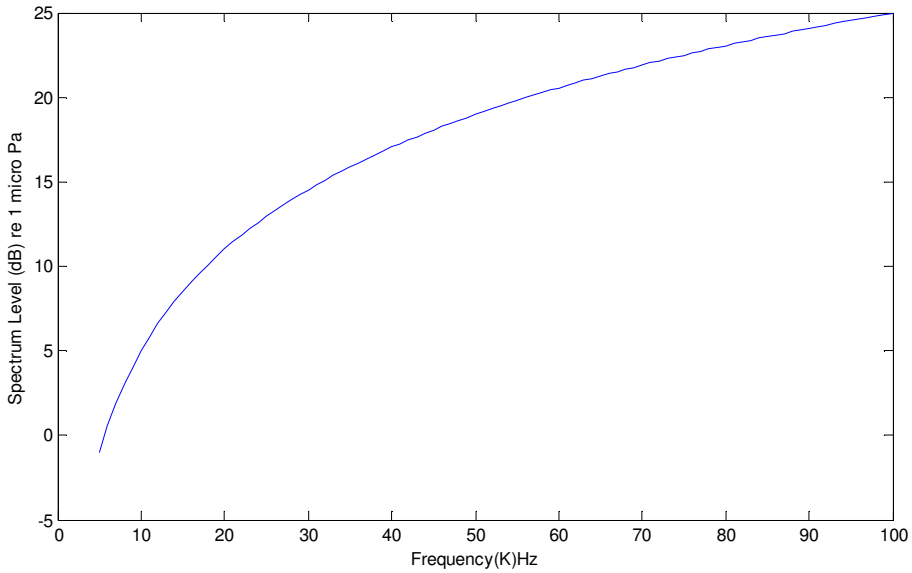

Figure 3.3 Thermal Noise characteristic

3.3.2.Ambient Noise

Ambient noise is an important noise that affects significantly the received sound wave. The ambient noise only includes the noise components which are generated by the sea itself. The noise components are sourced by the sea surface waves, sea interaction with nearby shores, marine organisms, etc. Ambient noise may have very different characteristics according to sea conditions depending on depth, season, time of day and weather conditions.

The ambient noise is mostly affected by the sea state. Sea state a is scale that categorizes the force of progressively higher seas by wave height. This scale is mathematically co-related to the Pierson-Moskowitz scale and the relationship of wind to waves. Thus, increase in the sea state increases the ambient noise. The seat state is described by the wind speed and wave height. In the Table 3.3 Sea state, wave and wind speed is given.

Table 3.3 Sea state, wave and wind speed

Sea State	0	1	2	3	4	5	6
Wind Speed (knots)	≤1	5	13	16	19	22	28
Wave height (m)	0	0.05	0.4	0.7	1.3	2	3

The ambient noise is the dominant noise source in sonar. A typical spectrum of ambient noise as a function of frequency for different sea states is given in Figure 3.4 [31].

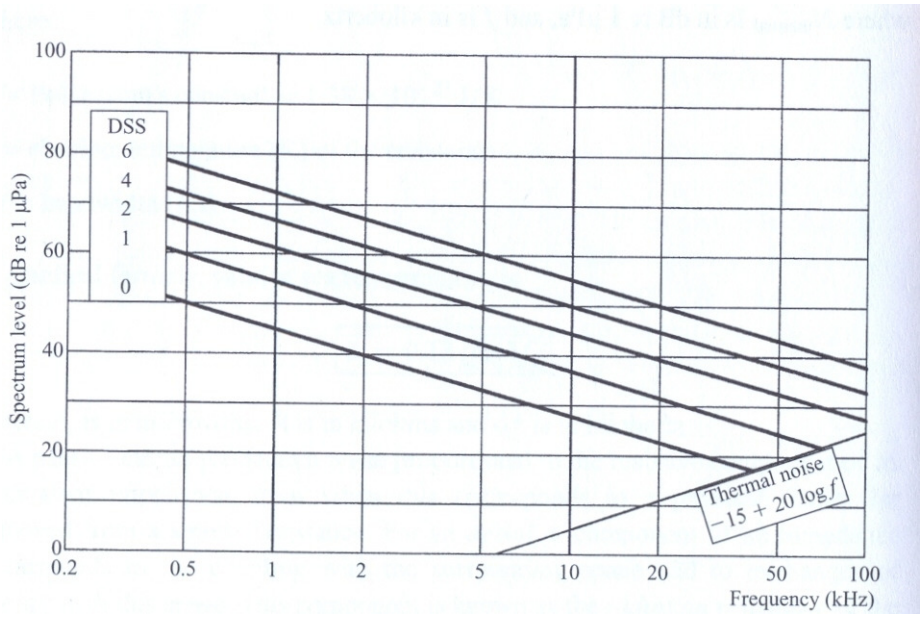

Figure 3.4 Ambient noise of the Sea

3.3.3.Other Noise Sources

The other common noise sources are shipping generated noises, specific biological noises (e.g. whale communication) and arctic noises. These noises sometimes affect the sound wave significantly but generally they are negligible.

Shipping noises are important in the operating frequencies below 500Hz. In shallow waters like harbors and coastal areas, the dominant noise may be the shipping noise and the noises that are caused by industrial activity.

Biological noises are caused by the marine organisms. The main contributors to biological noise are certain shellfish like shrimps, fish and marine mammals like whales and dolphins. The biological noise generally has a quite flat spectrum between 500 Hz and 20 Khz. Generally it can be neglected but sometimes its value can be high like 70 dB re 1μPa.

Arctic noises are caused by the ice cover on the ocean. The effect of the arctic noise is different from normal sea ambient noise. The ice cover affects the ambient noise significantly.

In this subsection noise sources for sea are discussed. As it is pointed out that the most significant noise is the ambient noise. The detailed information can be found in [29, 30]. In this study we consider a noise model that considers a non-Gaussian noise.

The total effects of the noise where two or more sources exist combine in an additive manner such that it can be computed by a summation of each independent noise source. This is given below in equation(3.12).

$$P_T = 10\log(10^{P_1/10} + 10^{10P_2/10}) \tag{3.12}$$

where P_T the total power of the noise and P_1 and P_2 are two independent noise power originally given in the dB scale.

3.4. Reverberation

There are physical effects that cause the sound wave transmitted from the sonar to reflect unintentionally and return to sonar again while traveling random paths. This unintentional reflection effect apart from the main reflection from the target is called scattering. The main sources of the scattering can be divided into two groups. One group is the boundaries of the sea and the other is all other scatterers being part of the ocean volume. The boundaries of the sea are sea surface and sea bottom. The sound waves are scattered when they hit the sea surface or bottom. The second group that causes scattering is the ocean volume. In the underwater environment there are a lot of inhomogeneities of different kinds. Some of the inhomogeneities in underwater environment are: the microorganisms, tiny particles of dust, fish etc.

The sum of scattering from both sea boundaries and ocean volume is called reverberation.

Reverberation is a large area of study but is not the primary focus of this study. Detailed information about reverberation can be found in [29,30].

3.5. Sonar Simulator Development for the Present Study

The sonar modeling is a large area of research due to the complex and chaotic environment of the underwater medium. In this study we have attempted to develop a sonar model which is not very complex but which can still reflect some of the dominant and interesting effects of underwater environment. We had to leave out many other important effects in order to define a manageable scope for our study. However, this is still an incremental approach where the model can be further developed in the future. In the following subsections the acoustic model that is used in the sonar simulations is described in more detail.

3.5.1.Propagation Model

The propagation of sound wave in underwater environment is subject to much more complicated effects as compared to the propagation of electromagnetic waves in air, at least for the majority of applications. One of the primary differences is the presence of distinct and highly different (in terms of velocity of sound) water layers within the region of interest for the sonar measurements. Although similar effects might be present for sonar, it might be argued that their relative influence is low and/or they are not dominant within the regions measurements are taken.

In this work we have assumed that the sea environment is composed of a layered layout where the salinity, temperature or pressure differs in each layer. All of these factors change the velocity and direction of propagation of the sound wave.

In this work we have assumed that the sound waves move linearly in the underwater environment except for a finite number of layer transitions that are explicitly modeled. In each distinct layer, the velocity and direction of the sound wave is assumed constant. Its velocity only changes when passing from one layer to another resulting in reflection and refraction. In this study, the reflected waves and the associated energy loss is ignored for the time being. The result of refraction of the sound wave is obtained by using the Snell's law. A typical example of the propagation of sound used in this simulator is given in Figure 3.5.

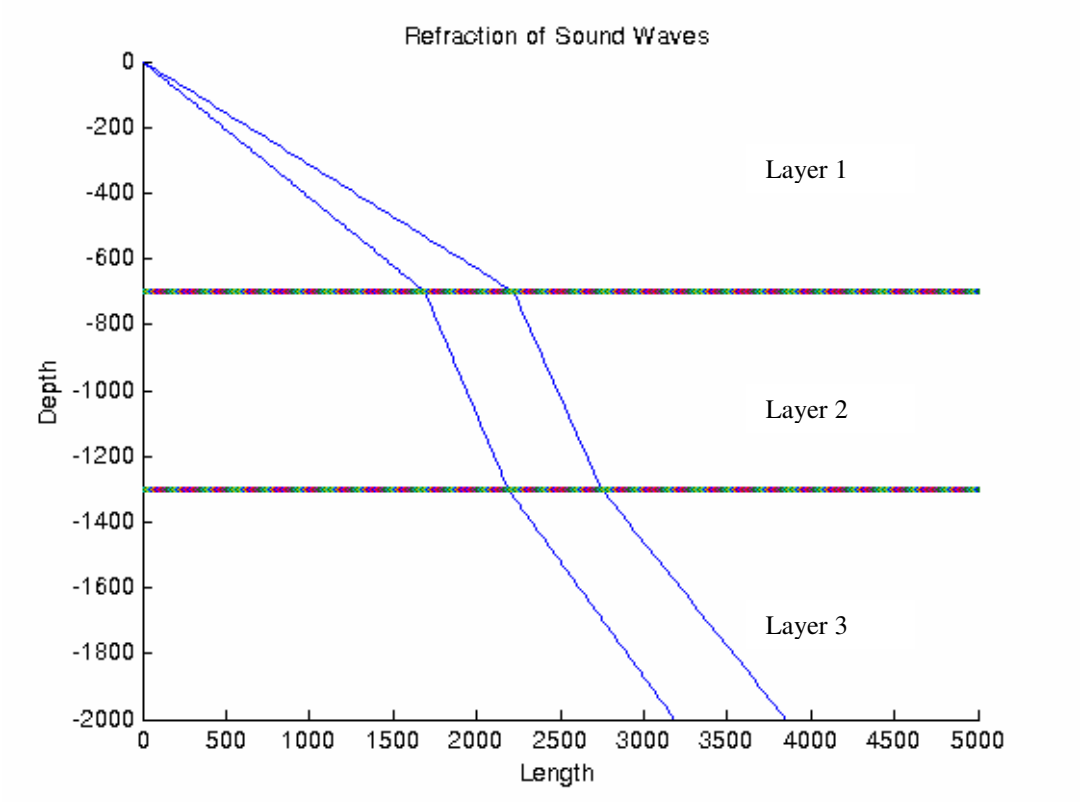

Figure 3.5 Propagation of Sound Wave Between Layered Underwater Environment. . Salinity of Layer 1: 35, Temperature of Layer 1: 20, Salinity of Layer 2: 100, Temperature of Layer 2: 50, Salinity of Layer 3: 50, Temperature of Layer 3: 40.

In Figure 3.5 sonar is located at (0,0) and sound wave propagation is illustrated with a piecewise linear plot. It is assumed that the transmitted wave propagates with an expansion angle (its limits illustrated in the figure) and signal power is maximum in the middle. The power is reduced by a given profile towards the edges of the fan angle. The layer properties used to generate the plot in Figure 3.5 are given in the caption.

3.5.2.Measurement Model

In the previous subsection the sound wave propagation model is given. During sonar operation, bearing and range measurements are obtained by the processing the returned sound wave reflected from a target. When the sound wave is reflected by an object and comes back to the sonar transceiver, it is subjected to attenuation and added noise. In this study if a target exists within the transmission fan angle, it is assumed that there will be a reflection and it will be picked up by the transceiver. Under these assumptions the received signal from the transceiver can be modeled as an attenuated version of the original pulse with a diversity of added noises. In this study we modeled the noise as a uniformly distributed white noise. Although other non-uniform and non-Gaussian noise models are also tried, these are not part of the focus of the thesis.

Using this model of attenuation and noise, a received signal is generated. The signal is then processed to the time delay between transmitted and received signal. By using this time delay, the range of the target is obtained. For obtaining the bearing of the target, the bearing of transmitted “ray” which has generated the detection is used. Assume that 4 transmitted rays are reflected back by the target. Then we use a weighted mean of the reflection magnitudes to obtain the bearing. In the following subsections we will further elaborate on these target detection and ranging models.

3.5.2.1.Transmitted Pulse Model

In this subsection the transmitted pulse model for the sonar is given. We used a simple sine wave of finite duration for the transmitted pulse. The frequency and the amplitude of the sound wave are configurable. The frequency is important when calculating the transmission loss. The transmitted pulse hence can be generated as follows:

$$S_{tranmitted} = A * \sin(2 * \pi * f * t) \tag{3.13}$$

where A is amplitude of sound wave, f is the frequency and the t is the time. For the digital representation of the pulse, the most important key point is the sampling period. Due to the Nyquist criteria, the sampling period must be at least $2*f$ for a lossless representation of the signal,. An partial illustration of the transmitted pulse (10 periods) is shown in Figure 3.6

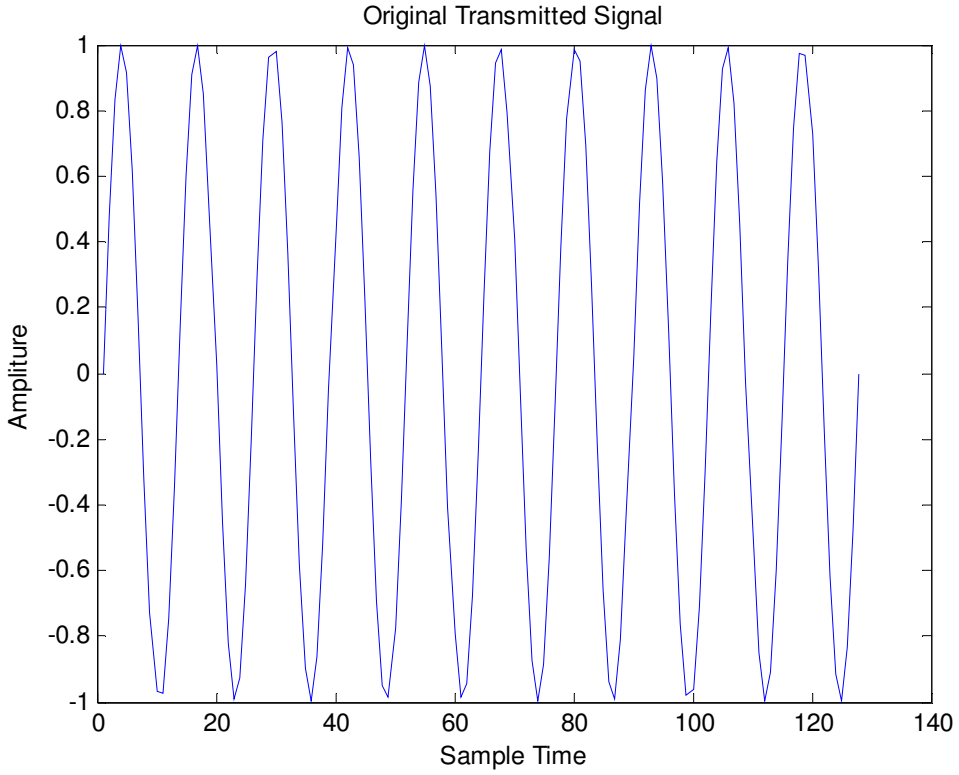

Figure 3.6 Transmitted Signal

3.5.2.2.Received Signal Generation Model

The sonar measures the range and bearing using the time difference between transmitted pulse and its reflection from the target. When a signal is transmitted from the sonar it is reflected from obstacles, sea surface, etc. and returned back to the sonar. Once the pulse delay is measured by using signal processing techniques, the range can be obtained by multiplying delay with the sound speed. Since the sound speed differs according to depth, salinity and temperature, these properties of the water must be taken into account in different layers when converting the total delay into distances corresponding to these particular layers.

While computing the time delay, we consider the effect of noise on the delay estimate. As described in earlier sections there are a lot of sources for noise. As given in [39] the noise in the underwater environment generally non-Gaussian. Thus the measurement noise is in practice non-Gaussian. In this study, to keep the noise model as simple as possible, we have used a uniform distribution to perform our simulation experiments. This noise has an effect on the detection of the return pulse and therefore on the delay estimation, which in turn adds noise to the range estimate.

The sound wave attenuates while propagating in underwater. The propagation loss is given in equation (3.7). To model the received signal, the propagation loss is calculated first. Using this propagation loss value the attenuation of the original transmitted pulse is determined. After the attenuated pulse amplitude is obtained, the noise is added. The obtained signal is the considered as the received signal subject to pulse detection. Using this noise and attenuation operations, the signal at the received end will be similar to the one given in Figure 3.7. Clearly the pulse is buried under noise and the original pulse is not visually apparent.

In this example it is assumed that the pulse is transmitted at time 0. Then the measurements are taken for 1664 samples. The pulse echo appears at sample 1024 and is of 128 samples duration.

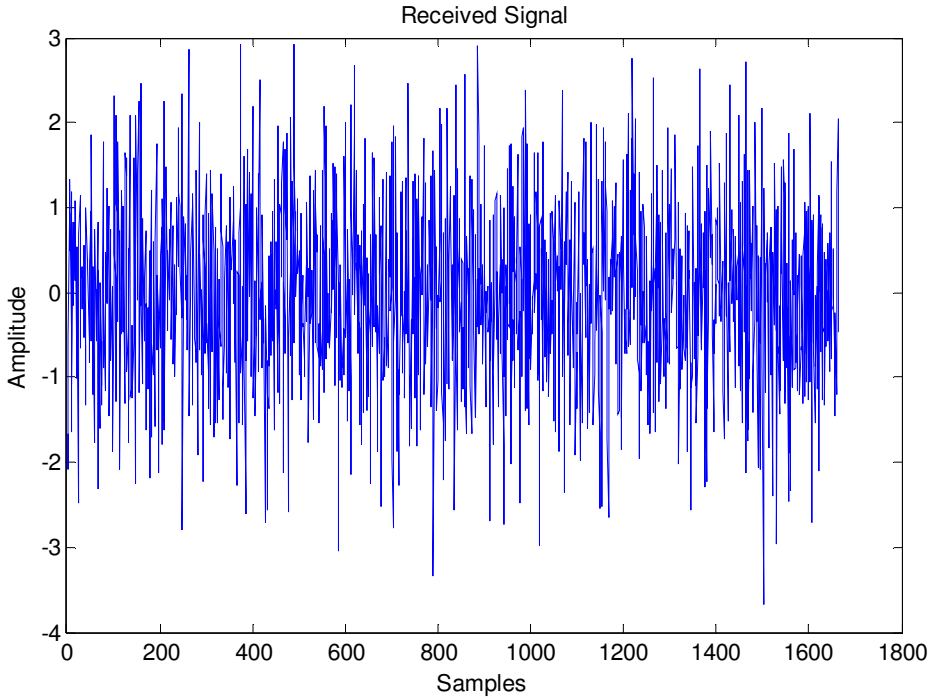

Figure 3.7 Received Signal

To find the location of the pulse echo under heavy noise, the received signal must be filtered. There are several filtering techniques to eliminate the noise. In this study a simple matched filter is used to enhance the echo for detection.

3.5.2.3. Matched Filtering

Matched filtering is a technique that is used to maximize the SNR (Signal-to-Noise Ratio) to enhance a known signal under noise. The detection of a signal can be improved by maximizing the SNR. For the derivation and theoretical background of

the matched filter, refer to [40]. Matched filter is obtained by time reversing the known signal that is to be detected. This is an optimal filter to detect the known signal since it can be shown that result of the convolution of the matched filter with the noisy signal attains its maximum power at the end of the pulse echo location. The matched filter is the optimal linear filter for maximizing the signal to noise ratio (SNR) in the presence of additive stochastic noise. If the pulse to be detected is $p[n]$, the corresponding matched filter is given by $h[n]=p[-n]$. Hence the result of the matched filtering is given by the following convolution:

$$y[n] = \sum_{k=-\infty}^{\infty} h[n-k]x[k] \tag{3.14}$$

where the selected matched filter $h[n]$ is convolved by the noisy signal $x[n]$. At the end of the filtering there will be peaks in the filtered signal which denotes the presence of original signal. We will discuss one way to detect these peaks.

In Figure 3.8 the filtered signal is illustrated. As it can be seen from the figure, the maximum signal energy is around sample 1100. This is our expected pulse echo location and should be detected automatically as follows.

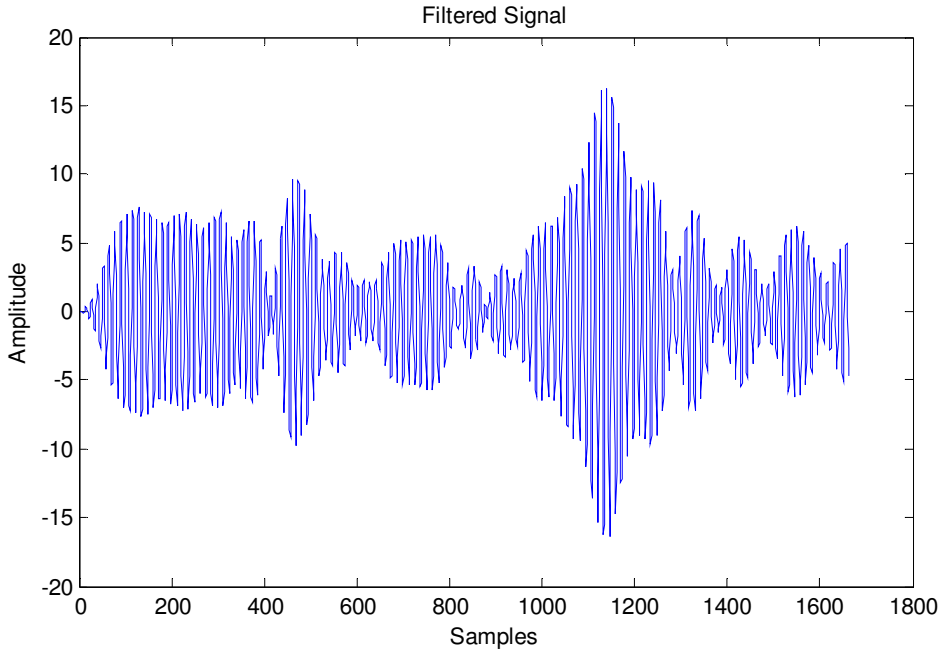

Figure 3.8 Filtered Received Signal

There are several techniques of detecting signal energy peaks. In this study we use polynomial (quadratic) fitting to find the location of the pulse echo.

To handle the negative values in the signal and represent the energy, the absolute value of the filtered signal is used. This is followed by choosing *n* largest values from the resulting signal around the signal maxima. Then these *n* samples values are used to find the best quadratic approximation as given below.

$$\tilde{f}(x) = ax^2 + bx + c \tag{3.15}$$

This approximating polynomial can be found using the '`polyfit`' command of MATLAB®. The '`polyfit`' command returns the coefficients *a*, *b* and *c* in equation (3.15). Then using these coefficients the energy peak location is found as

$$P_{loc} = \frac{-b}{2a}. \tag{3.16}$$

For this example, the value P_{loc} is found to be 1152 which is the exact location of the pulse echo ending.

The time difference between transmitted pulse and received pulse echo can be found by using the equation

$$\Delta t = \frac{(S_R - S_T)}{S_n} \tag{3.17}$$

where S_R denotes the received pulse echo sample number, S_T denotes transmitted pulse sample number and S_N denotes the sampling period. In the example given above, S_R is 1152, S_T is 0 and S_N is 128. Thus Δt=9 seconds.

3.5.2.4.Specifying the Sound Wave That is Reflected From Reflector

In our simulation the propagation of the sound wave is modeled as 2 lines which are the boundaries of horizontal beam-width. These 2 lines are propagated in the environment using the propagation characteristic of sound in the underwater. In the Figure 3.9 a screenshot from the simulator is given. As it is seen from figure, the sound waves are shown by 2 lines and the lines are refracted if a layer with

different salinity or temperature occurs. In the figure above blue lines show the sound wave boundaries and the red circle is target.

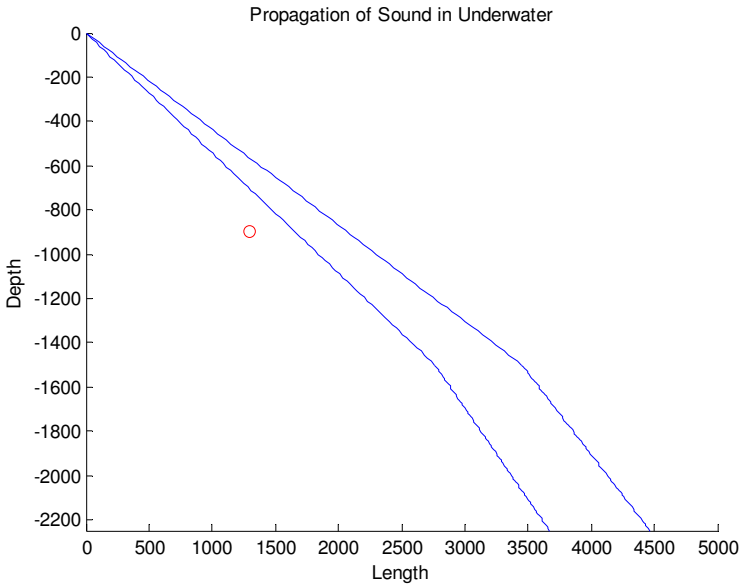

Figure 3.9 Propagation of Sound in Underwater Environment

In the figure above the target is out of boundaries so the sound wave only reflects from bottom. If a target is detected between these boundaries the time delay is computed by finding the line which thoroughly passes from the target.

In the Figure 3.10 the target is detected between two sound lines. There are two layers in the following figure. The line passes through the target which is drawn by red gives us the length of sound wave. By using the length of the line and using sound speed the time delay can be computed. The line that passes through the target is identified by $(x_s,y_s,x_l,y_l,x_t,y_t)$ where x_s and y_s are sonar coordinates, x_l and y_l are the coordinates that the line intersects with layer l and x_t are y_t coordinates of target. All parameters except x_l are known.

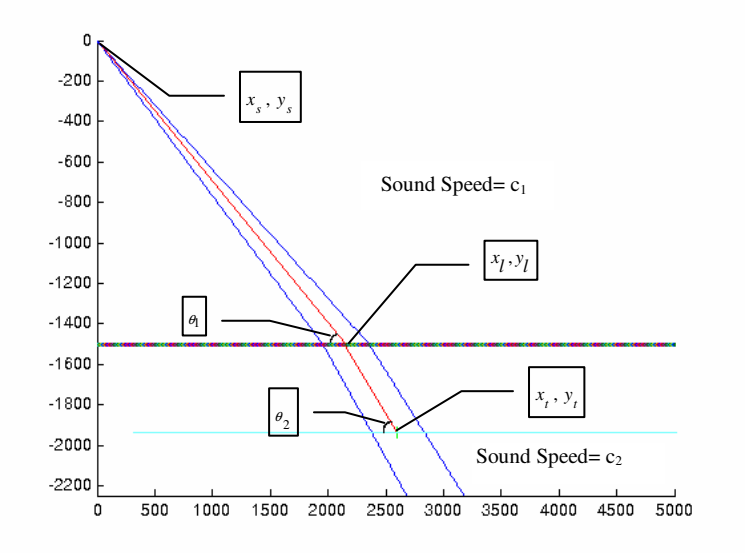

Figure 3.10 Sound Wave Reflected From Target

To find the length of line first x_l must be calculated. The x_l is calculated using Snell's law given in equation(3.10). The parametric values of $\cos(\theta_1)$ and $\cos(\theta_2)$ are given as follow.

$$\cos(\theta_1) = \frac{(x_l - x_s)}{\sqrt{(x_l - x_s)^2 + (y_l - y_s)^2}} \tag{3.18}$$

$$\cos(\theta_2) = \frac{(x_t - x_l)}{\sqrt{(x_t - x_l)^2 + (y_t - y_l)^2}} \tag{3.19}$$

Using equation (3.10) we can write

$$\frac{\dfrac{(x_l - x_s)}{\sqrt{(x_l - x_s)^2 + (y_l - y_s)^2}}}{c_1} = \frac{\dfrac{(x_t - x_l)}{\sqrt{(x_t - x_l)^2 + (y_t - y_l)^2}}}{c_2} \tag{3.20}$$

The equation (3.20) is arranged is follows

$$c_2 \times \frac{(x_l - x_s)}{\sqrt{(x_l - x_s)^2 + (y_l - y_s)^2}} = c_1 \times \frac{(x_t - x_l)}{\sqrt{(x_t - x_l)^2 + (y_t - y_l)^2}} \tag{3.21}$$

If we expand equation (3.21) we obtain a third order equation which has 36 terms. This equation is solved by using MATLAB's '`solve`' command. After x_l is found the total time can be as follows.

$$\Delta t = 2\times\left(\frac{\sqrt{(x_l - x_s)^2 + (y_l - y_s)^2}}{c_1} + \frac{\sqrt{(x_t - x_l)^2 + (y_t - y_l)^2}}{c_2}\right) \tag{3.22}$$

In equation (3.22) the time difference is obtained by summing the traveling time of sound in layer 1 and layer and then multiplying by 2. The sum is multiplied by 2 because the sound wave travels 2 times on the same path, propagation and reflection.

3.5.2.5.Range Calculation

The transmitted sound wave comes back when it is reflected from a reflector. The reflectors can be target, sea surface, sea bottom, etc. The range of the reflector is found by using the time difference between transmitted and received signal. After obtaining the time difference between the transmitted and received signal the range of the reflector can be found as following.

$$R_{reflector} = \frac{\Delta t}{2}\times c_s \tag{3.23}$$

where $R_{reflector}$ denotes the range of the reflector to the sonar receiver, Δt denotes time difference between transmitted and received signals (as given in equation (3.22)) and c_s the speed of sound. The errors in detecting the original signal in the noisy received signal causes wrong time differences which yields the range measurement to be noisy.

3.5.2.6.Bearing Calculation

The sonar transmits the sound waves as in radar. There is a sound wave generator which is also called “loudspeaker” in the sonar head. The loudspeaker converts electrical energy into sound waves. The loudspeaker transmits the sound waves at specific vertical and horizontal beam widths. The loudspeaker turns 360 degrees in sonar head. It turns with small step size like 0,5˚-1,5˚. The bearing of the reflector is found by using the bearing of the loudspeaker. At the boundaries of horizontal and vertical beam width the power of sound wave is smaller than the centre of boundaries. The bearing is then weighted according to the distance of target to the boundaries of sonar waves. The reflection will be less when the target is far from the centre of sound wave boundaries and will be higher when the target is near to the centre. The weights are assigned according to the distance of target to the. The weights are assigned using a cosine function. Cosine function is a monotone decreasing function between [0,pi/2]. The distances (distance between sound wave boundaries and distance between centre of sound wave boundaries and target) are first scaled to [0,pi/2] and then the weight is calculated as follows.

$$w = \cos(d_{target} / d_{sound}) \tag{3.24}$$

where w denotes the weight, d_{target} denotes the scaled distance between the target and center of sound wave bounds, d_{sound} denotes the scaled distance between sound wave boundaries.

Assume that we have detect targets at the bearing angles $\theta_i, i = 1,...N$ where θ_i denotes the bearing of loudspeaker. Then the final bearing is calculated as follows

$$\theta_{target} = \sum_{i=1}^{N} w_i^n \theta_i \tag{3.25}$$

where the θ_{target} is the bearing of the target and the w_i^n is the normalized weight of θ_i. The normalized weights are calculated as follows.

$$w_i^n = \frac{w_i}{\sum_{k=1}^{N} w_k} \tag{3.26}$$

where w_i denotes the weight of θ_i .

3.5.2.7.Post Processing Stage

The measurements for bearing and range can not be used directly if there are at least 2 layers in underwater environment. Although the sonar gives the target position directly using the range and bearing measurement in reality the target is at different position because of refraction of sound. This situation is illustrated in Figure 3.11

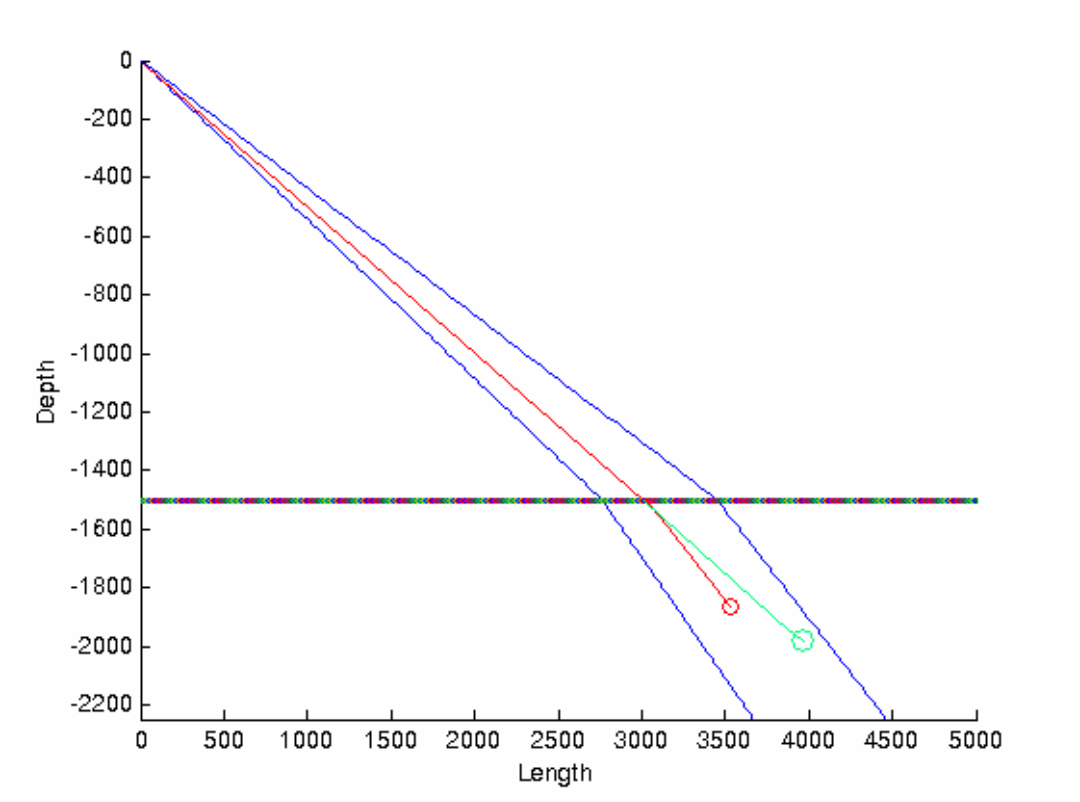

Figure 3.11 Ghost Image of The Target

The final step is post processing of measurements. In this step the measurements are updated by using the properties of underwater environment. Using the time difference between transmitted and received sound waves the corrected measurement is calculated. The calculation is described as follows.

1. Find the time that is passed until the sound wave travels between sonar up to layer 2 with bearing that is calculated by equation (3.25).
2. Find the new bearing angle for layer 2 using (3.10)
3. Subtract the time that is found in step 1 from total time difference.
4. Propagate the sound for the time that is found in step 3.
5. If a layer is detected while propagation go to step 2.
6. Update the target position

Notice that this calculation also yields an error in the measurement.

CHAPTER 4

UNDERWATER TARGET TRACKING SIMULATION AND RESULTS

4.1. Introduction

In this section several target tracking simulation scenarios are considered for underwater environment and results are given. As discusses in section 3.5.2.2, the noise in underwater environment is non-Gaussian as well as the sensor-apparent target system dynamics is nonlinear and hence these have an impact on some of the tracking algorithms, in particular the Kalman filter. In the following simulations the primary objective is to assess the effects of the considered underwater environment on target tracking algorithm performance..

The considered scenarios are based on different water layer structures and different target motions. The motions considered are according to the well known constant velocity (linear) and coordinated turn (nonlinear) dynamic models. The measurements taken from the sonar simulator are range and bearing. These measurements are updated as described in section 3.5.2.7.

In the following sections we first start from the simplest layer structure and motion and gradually add on both water layer complexity and target motion dynamics into the experiments, each presented as a separate experimental scenario.

4.2. Scenario 1

This scenario aims to compare the performance of Kalman Filter with a basic Particle Filter (PF-1000 with 1000 particles) in single water layer acoustic scenario and constant velocity target motion. In this experiment it is assumed that the underwater environment is homogeneous (there is only one water layer). The sonar is located at (0,0) and the target is located at (0,-300). The target moves horizontally in the x-direction with a speed of 5m/s. The process noise for KF and PF is chosen as white Gaussian with a variance of 1. The measurement noise is taken as uniformly distributed in the interval [0, 10]. The number of particles is set to 1000. The true trajectory of the target is given in Figure 4.1.

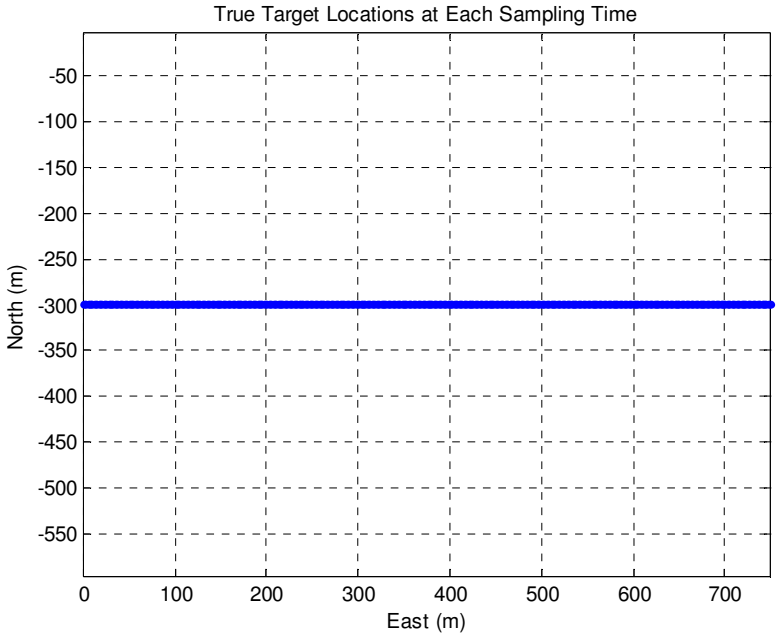

Figure 4.1 True Trajectory of target

In the following three figures, we illustrate the tracking performance of the considered filters. In Figure 4.2 the RMS position errors, in Figure 4.3 the Normalized RMS position errors and in Figure 4.4 the RMS velocity errors are given for KF and PF-1000.

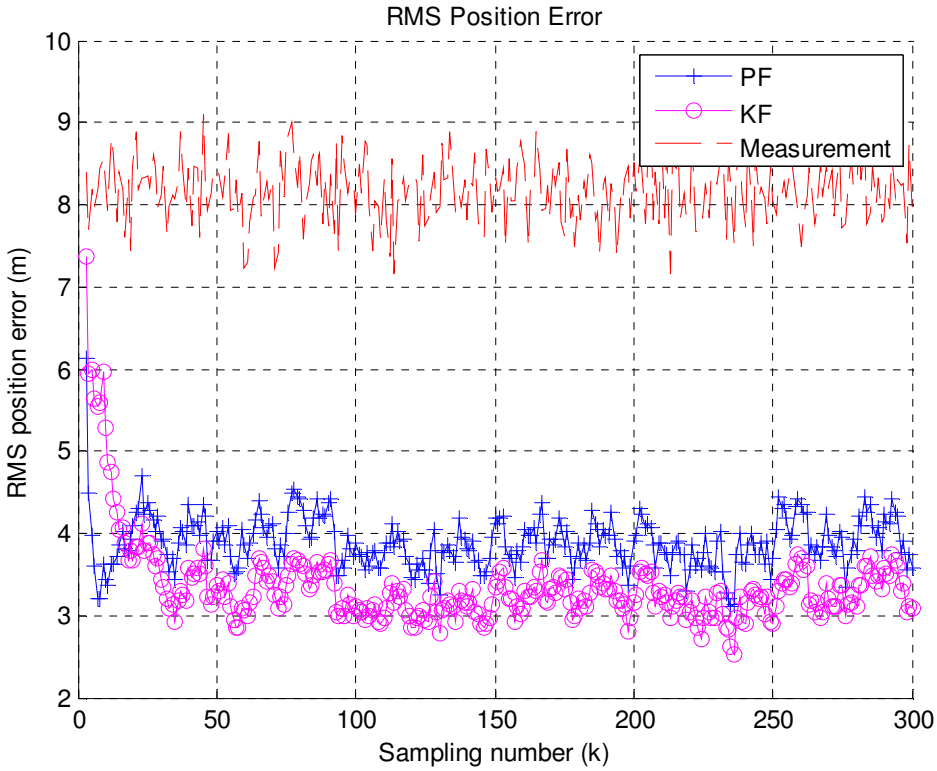

Figure 4.2 RMS Position Error

The figures illustrate that both KF and PF-1000 show close performance. Both result in estimates which are much better than plain observations from the sensor as expected. This is our baseline scenario where much of the underwater effects that we consider are absent and therefore plain KF performance is satisfactory. The particle filter also performs closely however with a much higher computational cost.

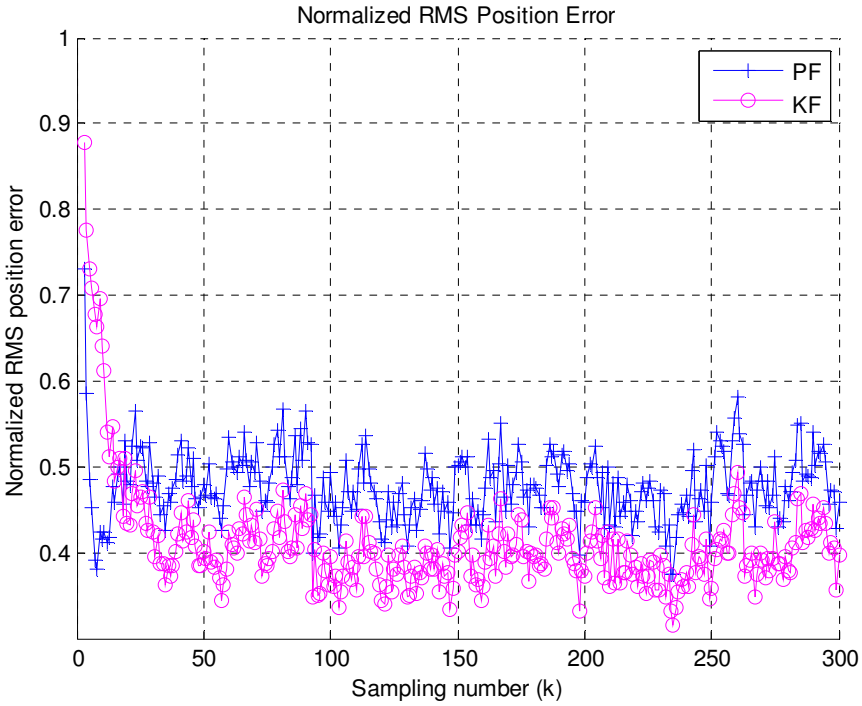

Figure 4.3 Normalized RMS Position Error

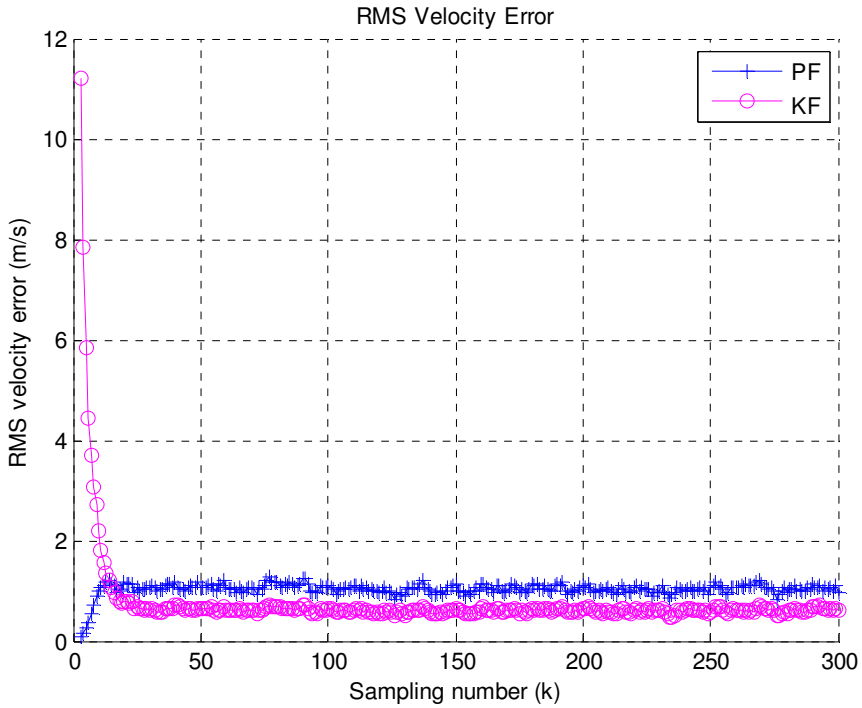

Figure 4.4 RMS Velocity Error

4.3. Scenario 2

This scenario aims to consider nonlinear (coordinated turn) motion and compare the performance of KF, EKF and PF-500 again under homogeneous single water layer underwater environment. Therefore in this experiment it is assumed that the underwater environment is homogeneous (there is only one layer). The salinity of the layer is taken as 35 ppt and the temperature is 20 ˚C. In this scenario performance comparison of KF, EKF and PF-500 are obtained. The sonar is located at (0,0) and the target is located at (0,-1800). The initial velocity of the target is along the y axis (north) and its direction and magnitude is -50m/s. The target then makes a nearly coordinated turn with turn rate of 1.2 degrees for 110 seconds. A positive turn rate implies a left turn and a negative one implies a right turn. Since the motion model is non-linear the process noise variance of KF is chosen larger to handle the turn maneuver. In this scenario the process noise variance is chosen as 3 for KF, 1 for EKF and PF-500. As the name implies the number of particles of the particle filter is 500. The model that is used in both PF and EKF is also the CTM, hence matching the actual motion dynamics. The model that is used in KF, since it has to be linear is WNAM. The number of Monte Carlo runs is 100. The sampling period for the sonar is 1 s meaning that the sonar has a full angular scan once every 1 s. In Figure 4.5 the true trajectory of the target is given. The subsequent figures present the tracking performance results for the cases of the three filters.

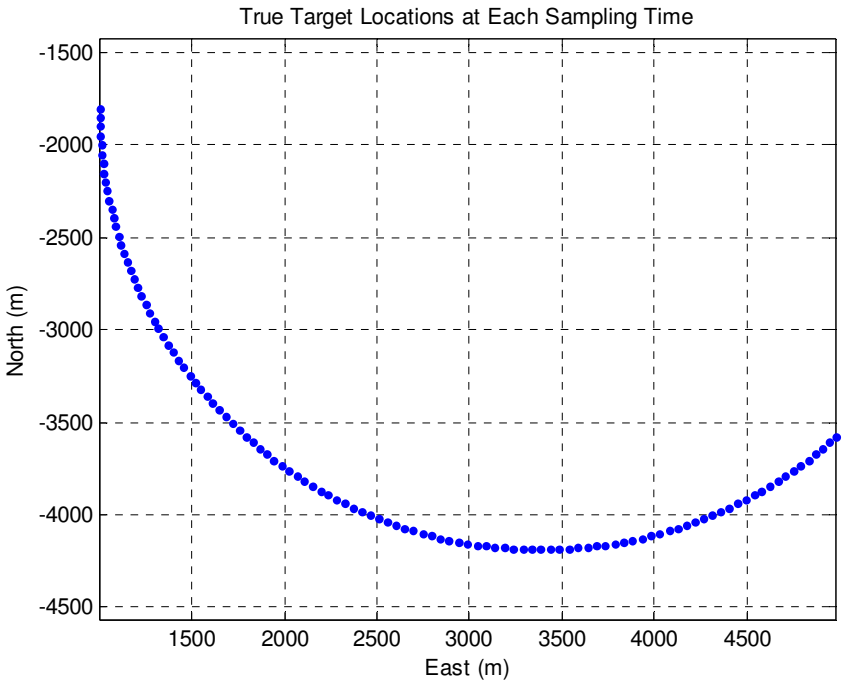

Figure 4.5 True Target Trajectory

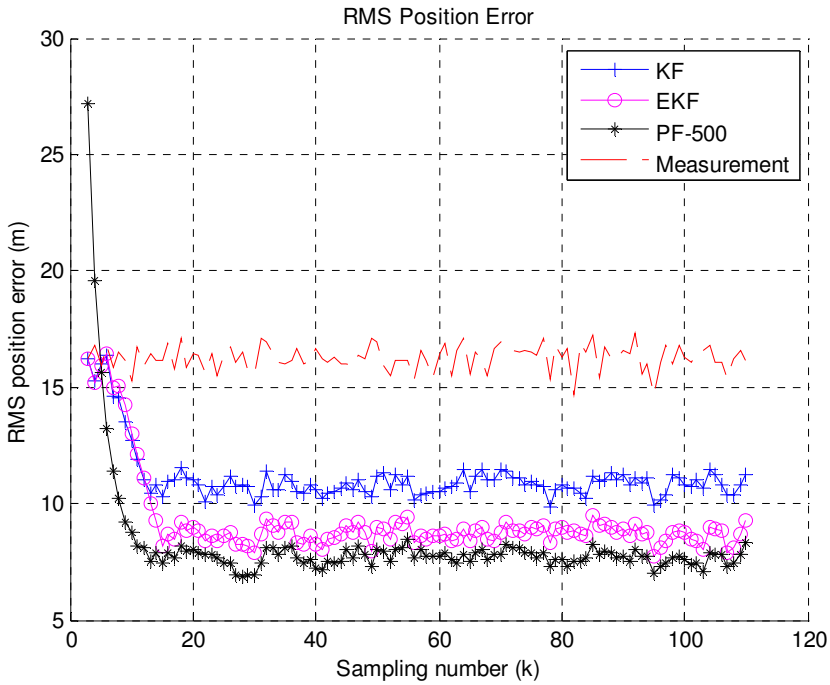

Figure 4.6 RMS Position Errors

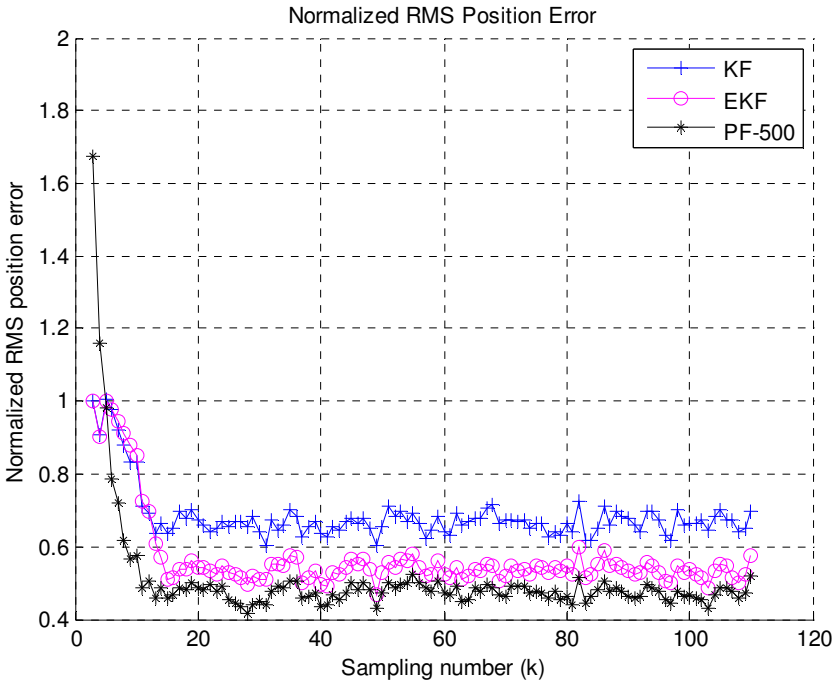

Figure 4.7 Normalized RMS Position Errors

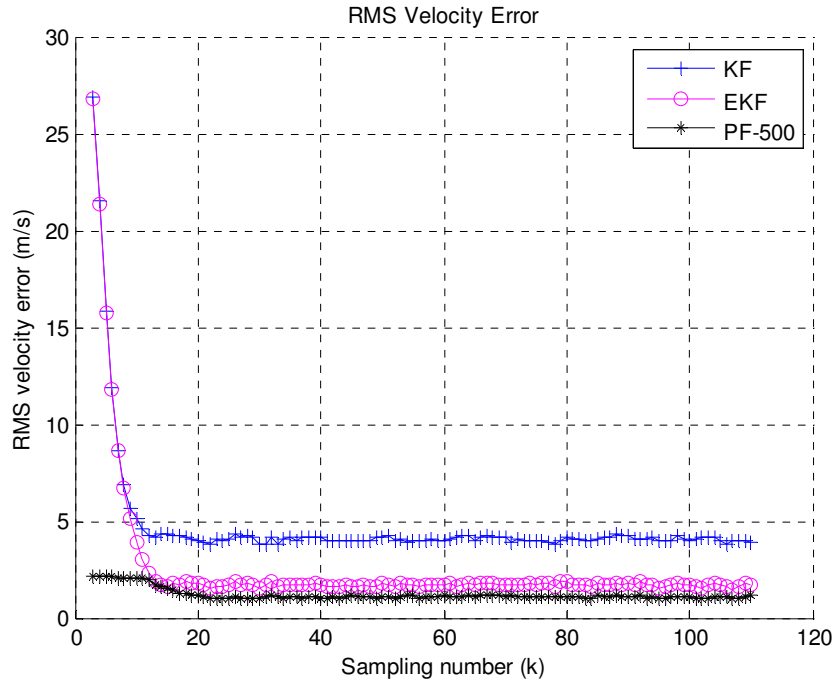

Figure 4.8 RMS Velocity Errors

As it can be seen from the figures above the performance of the KF is the worst among the tree filters. EKF and PF-500 are close in performance and PF-500 is marginally better than EKF. Note that in this underwater scenario the measurement noise is generated as non-Gaussian. This violates one of the main assumptions of KF and EKF. For the KF, both linearity and Gaussian assumption fails. Therefore this filter suffers the greater impact. For EKF which is able to partially deal with nonlinear dynamics, only the non-Gaussian noise is a violation. However, this violation itself pushes the EKF performance below the PF as compared to the experiments with Gaussian noise where EKF was the better performing filter. (See Section 2.8.2.) Also note that the performance of the PF can be further increased by increasing the number of particles (500 in this case) but this effect is not further analyzed in this scenario.

4.4. Scenario 3

The primary focus of this scenario is to investigate the presence of multiple layers as well as introducing multiple dynamics (change of behavior) for the target. Therefore, in this experiment it is assumed that the underwater environment is not homogeneous (there are now two distinct layers). The salinity of layer 1 is 35 ppt and the temperature is 20 ˚C. The salinity of layer 2 is 50 ppt and the temperature is 30 ˚C. The combined effect of salinity and temperature difference is a noticeable change in speed of sound between the layers, having the discussed effects on sound propagation. Since the target dynamics is changing, the Interacting Multiple Model (IMM) versions of the filters need to be utilized. In this scenario we compare the performances of IMM-KF (which in fact includes KF and EKF), IMM-PF-100 and IMM-PF-500. The sonar is again located at (0,0) and the target is initially located at (1000,-4000). The target initially moves linearly in the y-axis. The target then switches to the near coordinated turn motion which is followed again by a linear portion. The parameters of the target motion are as follows: the target moves linearly along the y-axis (north) for 40 seconds with a speed of 50m/s, then makes a

near coordinated turn at a turn rate of -3 degrees for 50 seconds and then moves linearly again for 60 seconds at 50 m/s. Since multiple modes of motion are present, multiple models are present in the IMM filters. For the KF/EKF based IMM filter, KF with WNAM is used to model the linear part of the motion while EKF with CTM is used to model the turn and the IMM structure handles the mixing of the two models. For the PF based IMM filter, two models (again WNAM and CTM) with identical particle numbers are utilized. We have experimented with 100 (IMM-PF-100) particles and 500 (IMM-PF-500) particles. The process noise variance for both the WNAM and CTM models are taken as 4. The layers in the underwater medium affect the measurements and so the measurement noise becomes further distorted and still non-Gaussian. In Figure 4.9 the true trajectory of the target is given. The subsequent figures present the tracking performance results for the cases of the three filters for 100 monte-carlo runs.

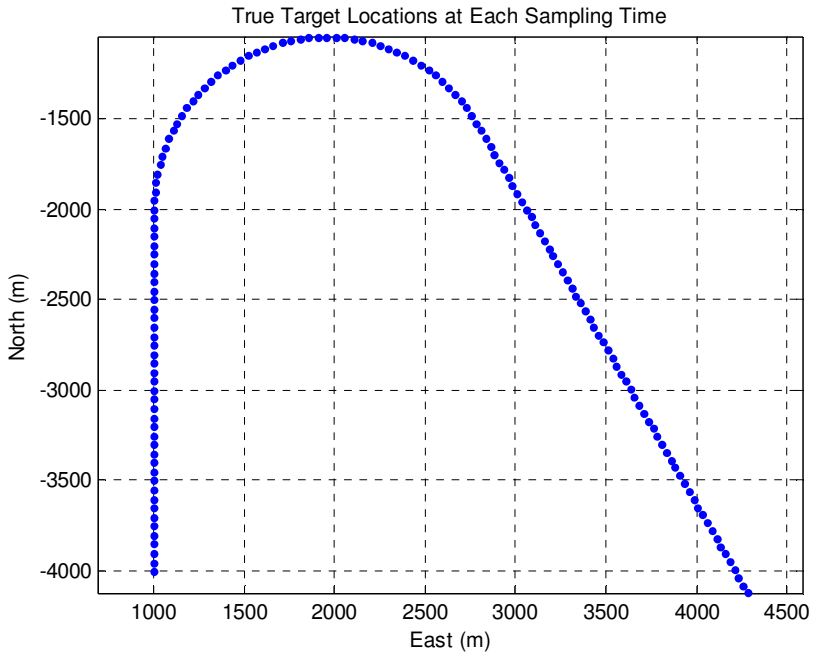

Figure 4.9 True Trajectory of Target

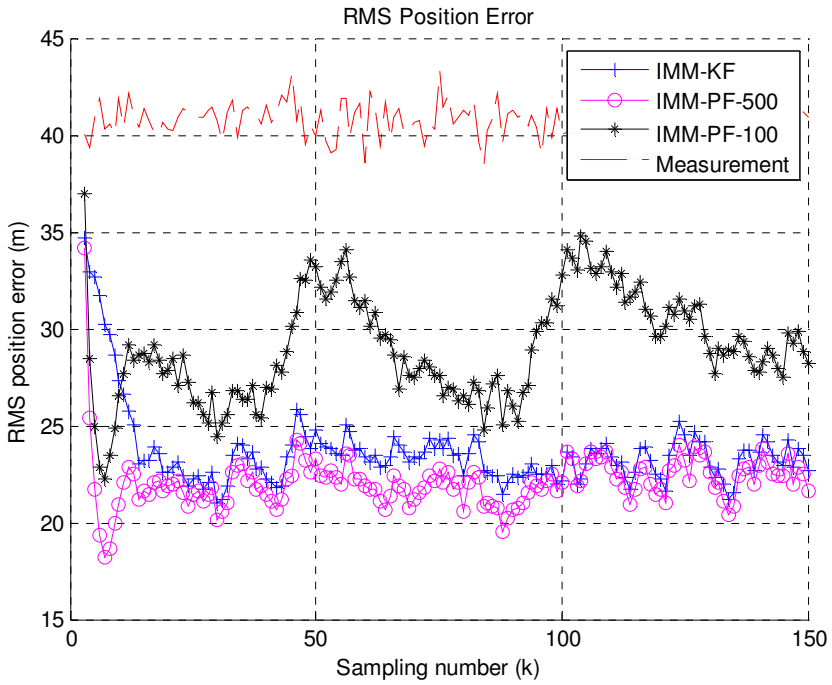

Figure 4.10 RMS Position Errors

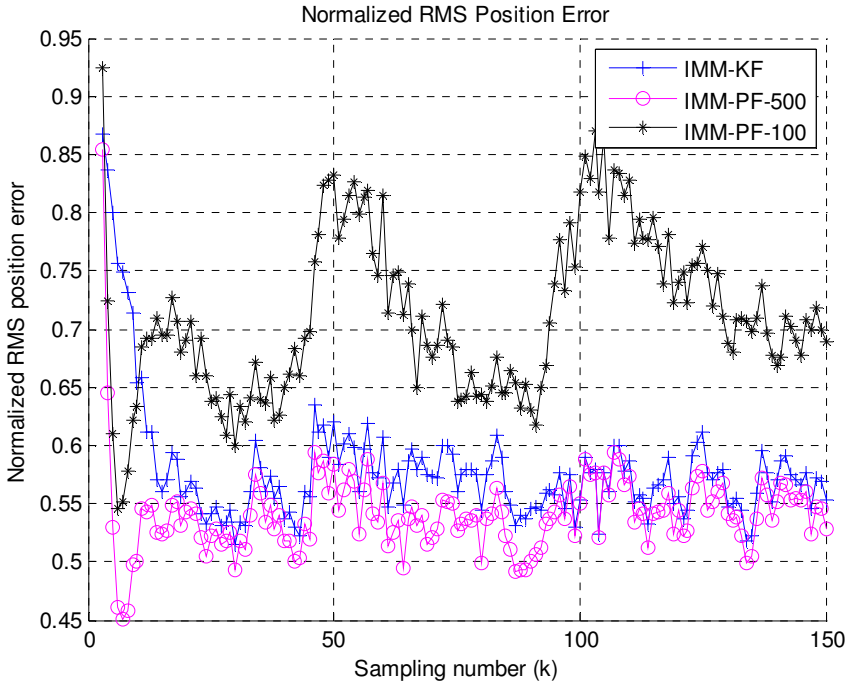

Figure 4.11 Normalized RMS Position Errors

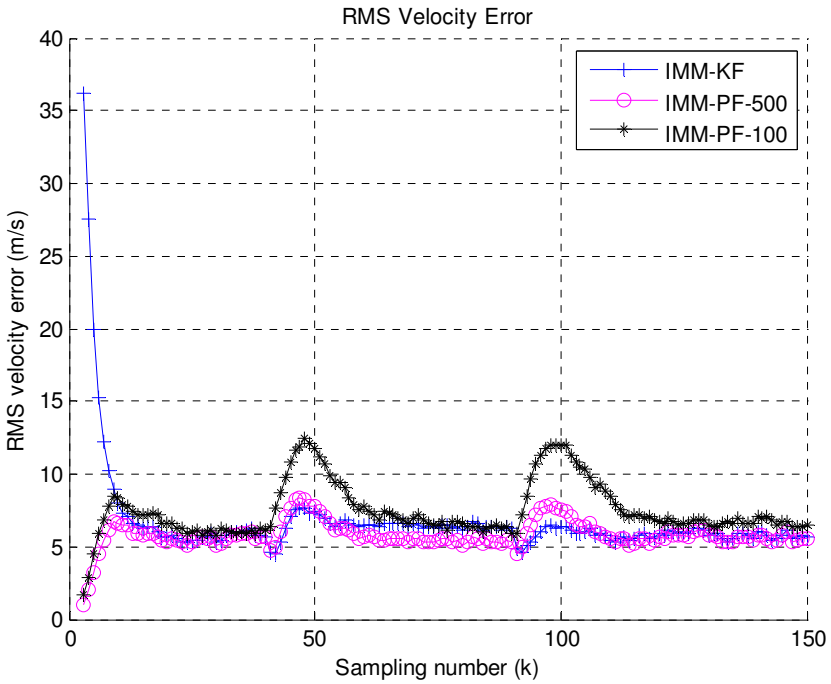

Figure 4.12 RMS Velocity Errors

In both the position and velocity errors, the characteristic adaptation of the IMM filter can easily be observed. Here, the estimation errors temporarily increase while the target changes its dynamics and the IMM filter adjusts the model probabilities after which the error is again brought down. Both the steady state performance differences as well as performance during these transitions can be inspected to compare two filters.

In terms of steady state performance one can note that a particle filter with appropriate (large enough particle number) is able to give the best estimation performance while an inappropriately chosen parameter set for the PF (too small number of particles) can also show the worst performance. The Kalman based IMM-KF is still able to provide acceptable performance in the presence of non-Gaussian noise that we consider.

In Figure 4.13 the mode probabilities of IMM-KF are given. In this figure WNAM model is denoted as 'Model 1' and CTM model is denoted as 'Model 2'. The model probabilities denote which model is more reliable at current time step.

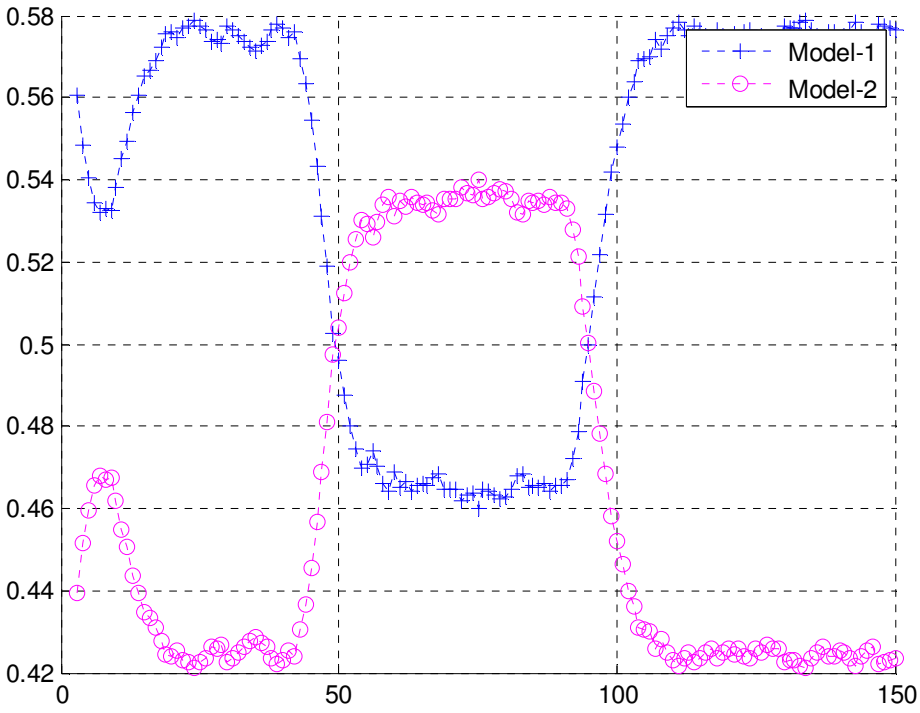

Figure 4.13 IMM-KF Model Probabilities

In Figure 4.14 the mode probabilities of IMM-PF-50 are given. In this figure WNAM model is denoted as 'Model 1' and CTM model is denoted as 'Model 2'.

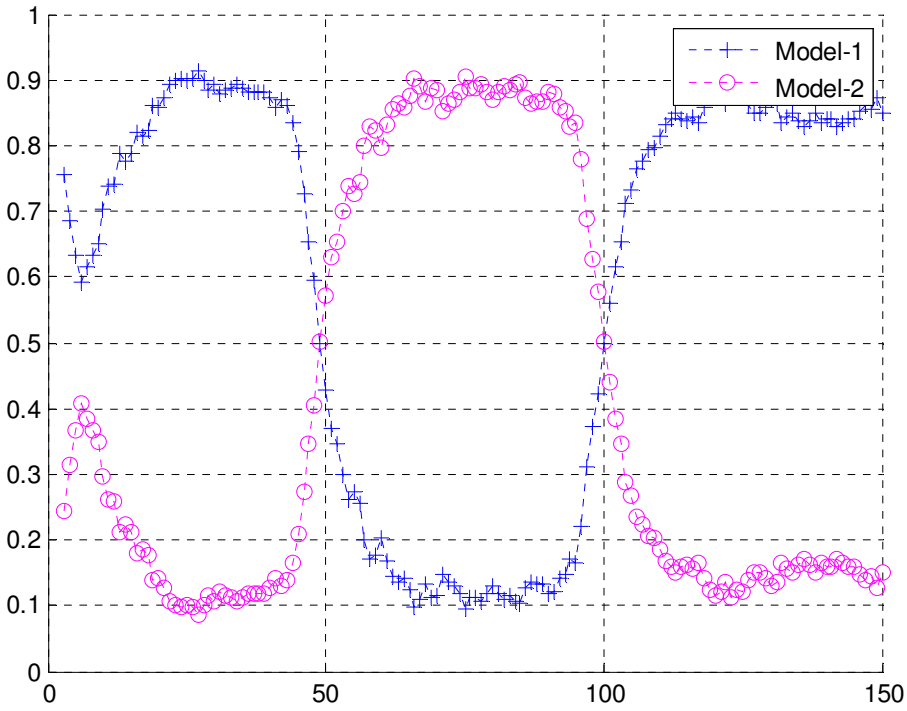

Figure 4.14 IMM-PF-50 Model Probabilities

In Figure 4.15 the mode probabilities of IMM-PF-500 are given. In this figure WNAM model is denoted as ‘Model 1’ and CTM model is denoted as ‘Model 2’.

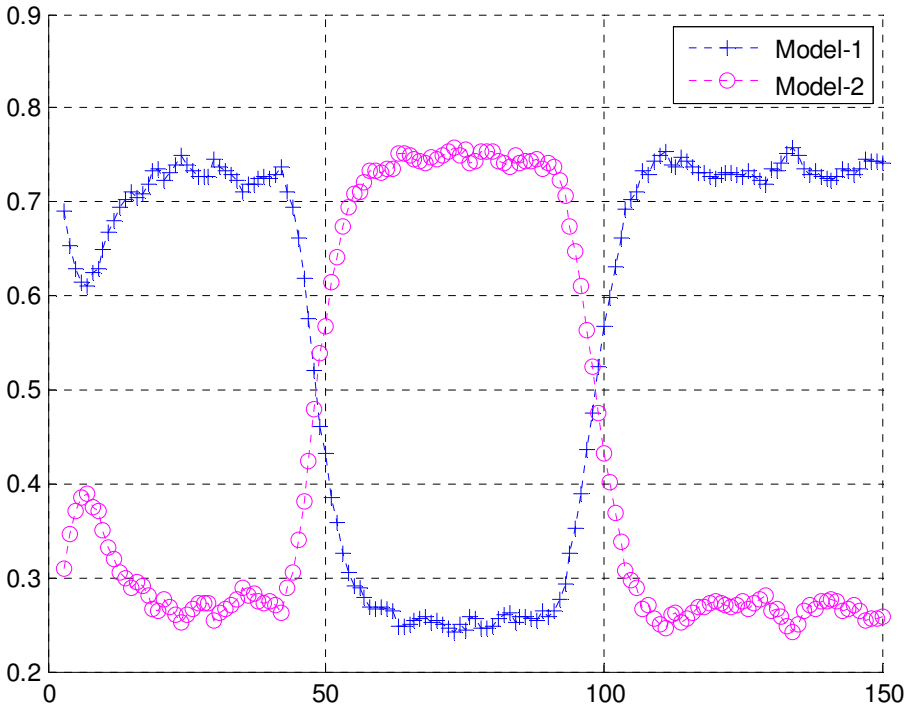

Figure 4.15 IMM-PF-500 Model Probabilities

4.5.Discussion

As mentioned in Chapter 2, the KF and its variations assume the noise is Gaussian. But due to chaotic environment of underwater, the noise is generally non-Gaussian. This effect decreases the performance of the KF and its extensions. In this situation we must use non-Gaussian estimators. In the experiments given above performance comparisons of KF, EKF, IMM-KF, PF and IMM-PF are given for several underwater environment scenarios.

For scenario 1, the target's motion is linear and the measurement noise is non-Gaussian. As it is mentioned in Chapter 2 KF is best linear estimator for non-Gaussian case. The expected result of the scenario 1 is to observe that the

performance of KF is the best. As it can be seen from the Figure 4.2 the KF is the best.

For scenario 2, the target's motion is non-linear and the measurement noise is non-Gaussian. In this case the main assumptions of KF and EKF are violated. In this scenario the performances of KF, EKF and PF are obtained for a homogeneous underwater environment. Since the assumptions are violated it is expected that the performance of non-Gaussian estimators are better. As it can be seen from the Figure 4.6 the performance of PF is better than KF and EKF.

For scenario 3, performance comparison of IMM-KF, IMM-PF-100 and IMM-PF-500 for a maneuvering target is obtained. In this scenario the underwater medium contains 2 layers. The measurement noise is non-Gaussian. The expected results is IMM-PF-500 is to be better than the other two filters. As it can be seen from the Figure 4.10 IMM-PF-500 is best.

The noise in the underwater environment is non-Gaussian. The non-Gaussian noise violates the one of the main assumptions of KF and its extensions. So in underwater environment using non-Gaussian estimators gives better performance for non-linear targets motions. In the above scenarios it is shown that the performance of non-Gaussian estimators are better than KF and extension for underwater environment.

CHAPTER 5

CONCLUSION

5.1.Discussion of Results

In this study we attempt to partially model the effects of underwater environment to target tracking algorithms. In the aerial environment with the assumption of Gaussian noise, the KF and its nonlinear extension the EKF are the most reliable estimators and these assumptions are usually reasonable. However, some of these assumptions fail for the underwater environment with the associated performance impacts.

In our study, we first reviewed the estimators used in target tracking applications and characterized their performances for aerial applications where the assumptions for the KF and EKF hold. In the air, the noise is commonly Gaussian and so the most accurate results are obtained by KF and its extensions. Secondly we have developed a sonar model for underwater measurements. And then the performances of the same set of estimators are inspected this time for the underwater environment.

The underwater environment introduces strong effects to disturb the Gaussian and linearity assumptions. There are numerous noise sources that effect the received signal significantly. As a collection of these major effects, the noise in an

underwater environment is known to be non-Gaussian. In this study we show that the Gaussian estimators suffer from this fact. Thus in this environment, non-Gaussian estimators such as sequential Monte-Carlo techniques (also known as the Particle Filter) can be used for improved tracking performance.. In this study PF and IMM-PF is used as the considered non-Gaussian estimators. Main disadvantage of PF is its increased computational complexity. In this filter, the performance increase come with increased number of particles and with this also come the increased computational complexity. With increased computational power available in recent years for real-time applications, this additional complexity has become manageable and this category of filters has become an option for target tracking applications. However, whether the computational cost can be tolerated should be decided for a particular platform and application area. Our study has also shown that within the considered simple models of underwater acoustics, the EKF algorithm (and also its IMM counterpart) can be used with reasonable success.

Since underwater acoustics is a vast field of study, we have limited ourselves to the most basic effects of the underwater medium to expose the differences of performance between the tracking filters considered. We cannot claim to have considered more complex phenomena and conditions in this medium. However the considered models were sufficient to expose a range of differences in performance.

5.2. Future Work

The most important future work of this study is to incrementally develop more detailed sonar acoustic models for the considered underwater environment.

In the present study, the layer structure of the medium is supposed to be known exactly and this information is required obtaining range and bearing measurements. However, it is usually impossible to get this information exactly and one relies on

other measurements techniques (salinity, temperature). Examining the filter performance degradations under layer structure estimation errors is an interesting line of future work. This would give a sensitivity analysis with respect to layer structure uncertainties.

Another important group of underwater environment effects which are not modeled in the present study are multi-paths and reverberation. The multi-paths and reverberation can cause false detections. The false detections can be considered as a form of “underwater clutter” and may decrease the performance of the estimator significantly. Handling of such effects can be attempted by adapting filters from the radar literature such as PDAF.

APPENDIX A

In Appendix A TafSim (Tracking and Fusion Simulator) is described. TafSim is originally developed by Murat Şamil ASLAN. This software is developed for Multisensor-Multitarget tracking. At the beginning of this study the TafSim was able to handle KF and IMM-KF for WNAM model. During this study WPAM, CTM models, EKF, IMM-KF, PF and IMM-PF are implemented also. Tafsim uses a radar model to obtain observations. During this study also sonar model is included.

Scenarios for target tracking are defined by a configuration file. An example configuration file is given below.

```
clear global; clc;
global params

% % ------------------------------------------------------------------------------
%                  SIMULATION ENVIRONMENT PARAMETERS
% ------------------------------------------------------------------------------
% params.simulation.title      = ['Simulation Title'];
%Define the number of monte carlo runs
params.simulation.number_of_Monte_Carlo_runs            =50;
%Define simulation start and end times(seconds)
params.simulation.start_time                            = 0;
params.simulation.stop_time                             = 150;
%Define borders
params.simulation.survailance_region.borders   = [0 20 0 20]';
params.simulation.survailance_region.clutter_density = 12*(1e-6);
params.simulation.clutter_generation_region= 'r'; % either s or r, corresponding
surveilance region or radar

% ------------------------------------------------------------------
%                 TARGET PARAMETERS
%       Pe        Ve        Ae        Pn        Vn        An
% ------------------------------------------------------------------
%Define the initial state of target [Pos_x Vel_x Acc_x Pos_y Vel_y Acc_y]
params.target{1}.initial_state = [  0 5 0 -300 0 0  ]';
%Define the motion of the target
params.target{1}.motion_description = { 't = 0:  WNAM, 0,''''N/A'''
't = 60:  CTM , 0, pi/150'
't = 120: WNAM, 0, ''''N/A'''              };
```

```
%Define the target detection probability
params.target{1}.probability_of_detection   = 1;

%\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
%            FILTER PARAMETERS
% \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
%Connstruct the filter
KF1     = KF_Constructor('WNAM',2,[1 1]');
PF1     = PF_Constructor('CTM',2,[1 1]',1000,pi/50);
EKF1     = EKF_Constructor(CTM,2,[0.5 0.5]',pi/60);
EKF2     = EKF_Constructor('WNAM',2,[0.5 0.5]',pi/60);
KF2     = KF_Constructor('WPAM',2,[0.6 0.6]');
IMMKF1=IMMKF_Constructor({EKF2,KF2},[0.5 0.5]',[0.9 0.1; 0.33 0.67]);

params.filter{1} = PF1;
params.filter{2} = KF1;
params.filter{3} = EKF1;
params.filter{4} = IMMKF1;

clear KF1 KF2 IMMKF1 PF1;


% -------------------------------------------------------------------
%                   RADAR PARAMETERS
% -------------------------------------------------------------------
% % This parameter can be either 'auto' or 'manual'. If 'auto' is selected,
% then the associated measurement covariance matrix is automatically
% computed by using pulse width and beam width parameters of the radar.  In
% the 'auto' case it is assumed that the radar provides measurements in
% polar coordinates. The 'auto' option overrides the measurement space
% parameter.

params.radar{1}.measurement_covariance_specification  = 'manual';
params.radar{1}.measurement_space                   = 'cartesian'; % This can be
either 'cartesian' or 'polar'
params.radar{1}.measurement_covariance         = diag([10^2 10^2]);
params.radar{1}.pulse_width                    = 1e-6; % s
params.radar{1}.beam_width                     = 1.3;  % deg
params.radar{1}.scanning_parameters.starting_time        = 0;
params.radar{1}.scanning_parameters.minimum_duration     = 0.5;
params.radar{1}.scanning_parameters.maximum_duration     = 0.5;
params.radar{1}.location                                 = [0 0]';
params.radar{1}.operate_clutter_free                     = 1;
params.radar{1}.clutter_density                       = 13*(1e-6);
params.radar{1}.tracking_filter_list                  = [1 2];
params.radar{1}.modelnames_to_be_initialized          = {'WNAM', 'WPAM', 'CTM'};
params.radar{1}.number_of_initialization_measurements = 2;
% -------------------------------------------------------------------
%                   SONAR PARAMETERS
% -------------------------------------------------------------------

%Define whether sonar or radar is used
params.isSonar = 0;
%Initialize Underwater Environment
%InitializeUnderwaterEnvironment(SeaSurfaceStart_x,SeaSurfaceStart_y, Length,
Depth,LayerNumber)
%AddLayer(LayerIndex,LayerStart_y, LayerDepth, Salinity, Temperature)
InitializeUnderwaterEnvironment(0,0, 5000, 2000,3);
AddLayer(1,0, 700, 35, 20);
AddLayer(2,-700, 600, 100, 50);
AddLayer(3,-1300, 700, 50, 40);
params.Sonar.BeamWidth =5;
params.Sonar.Position = [0 0];
params.Sonar.StepSize=1,5;
params.Sonar.StartAngle = -20;
params.Sonar.EndAngle = -65
params.Sonar.Sound.Frequency = 4; %must be in kilohertz
params.Sonar.Sound.Amplitude = 1;
SONAR_FIGURE=figure;
```

REFERENCES

[1]. Y. Bar-Shalom, X.-R. Li, T. Kirubaajan, "Estimation with Applications To Tracking and Navigation", John Wiley & Sons Inc., 2001

[2]. M. S. Grewal, A. P. Andrews, "Kalman Filtering: Theory and Practice Using MATLAB", 2nd Edition, ", John Wiley & Sons Inc., 2001

[3]. H. W. Sorenson, "Kalman Filtering: Theory and Application", IEEE Press Newyork, 1985

[4]. Y. Bar-Shalom, T. Fortmann, "Tracking and Data Association", volume 179 of Mathematics in Science and Engineering, Academic Press, 1988

[5]. R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems", Trans. ASME, J. Basic Engineering, 82:34-45, 1960

[6]. R. E. Kalman, R. Bucy, "New Results in Linear Filtering and Prediction Theory", Trans. ASME, J. Basic Engineering, 83:95-108, 1961

[7]. B. Anderson, J. Moore, "Optimal Filtering", Prentice Hall, Englewood Cliffs, 1979

[8]. D.T. Magill, "Optimal Adaptive Estimation of Sampled Stochastic Processes", IEEE Trans. on Automatic Control, 10:434-439, 1965

[9]. H.A.P. Blom, "An Efficient Filter for Abruptly Changing Systems", In Proc. 23rd IEEE Conf. Decision and Control, Las Vegas, 1984

[10]. M.S. Arulampalam, S. Maskell, N. Gordon, T. Clapp, "A Tutorial on Particle Filters for on-line Non-linear / Non-Gaussian Bayesian Tracking", IEEE Trans. on Signal Processing, 50(2):174-188, 2002

[11]. R. G. Brown, P. Y. C. Hwang, "Introduction To Random Signals and Applied Kalman Filtering", Second Edition, John Wiley & Sons Inc., 1992

[12]. G. Welch, G.Bishop, “An Introduction to The Kalman Filter”, Course 8, SIGGRAPH, Los Angeles, CA, 2001

[13]. B. D. O. Anderson, J. B. Moore, “Optimal Filtering”, Prentice-Hall, Englewood-Cliffs, NJ, 1979

[14]. P. Dyer, S. McReynolds, “Extensions of Root-Square filtering to Include Process Noise”, Journal of Optimization Theory and Applications, Vol 3, pp:444-458, 1969

[15]. C.B. Chang, M. Athans, “State Estimation of Discrete System with Switching Parameters”, IEEE Trans. Aerospace Electronic Systems, Vol. AES-14,pp:418-425, 1978

[16]. J.K. Tugnait, “Detection and Estimation for Abruptly Changing Systems”, Automatica, Vol. 18 No:5, pp:607-615, 1982

[17]. H. A. Blom, “An Efficient Filter for Abruptly Changing Systems”, Proc. 23rd IEEE Conf. Decision and Control, Las Vega, NV, pp:656-658

[18]. H. A. P. Blom, Y. Bar-Shalom, “The Interacting Multiple Model Algorithms for Systems with Markovian Switching Coefficients”, IEEE Trans. Automatic Control, Vol. AC-33 No.8,pp:780-783, 1988

[19]. S. Blackman, R. Popoli, “Design and Analysis of Modern Tracking Systems”, Artech House, 1999

[20]. Y. Bar-Shalom, W.D. Blair, “Multitarget-Multisensor Tracking: Application and Advances”, Vol. 3, Artech House, 2000

[21]. I. M. Rekleitis, “A Particle Filter Tutorial for Mobile Robot Localization”, Technical Report TR-CIM-04-02, Centre for Intelligent Machines, McGill University, Montreal,2004

[22]. B. Ristic, S. Arulampalam, N. Gordon, “Beyond the Kalman Filter Particle Filters for Tracking Applications”, Artech House, 2004

[23]. A. Doucet, N. Freitas, N. Gordon, “Sequential Monte Carlo Methods in Practice”, Springer, Newyork, 2001

[24]. S. Arulampalam, S. Maskell, N. Gordon, T. Clapp, “A Tutorial on Particle Filters for On-line Non-linear/Non-Gaussian Bayesian Tracking”, IEEE Trans. on Signal Processing, Vol. 50, No. 2,pp:174-188, 2002

[25]. N. J. Gordon, D. J. Salmond, A. F. M. Smith, “Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation”, IEE Proceedings-F, Vol. 140 No.2, pp: 107-113, 1993

[26]. T. Higuchi, “Monte Carlo Filter Using the Genetic Algorithm Operators”, Journal of Statistical Computation and Simulation, Vol. 59, No.1, pp:1-23

[27]. J. S. Liu, R. Chen, “Sequential Monte Carlo Methods for Dynamic systems”, Journal of the American Statistical Association, Vol.93 No. 430, pp:1032-1044

[28]. J. Carpenter, P. Clifford, P. Fearnhead, “Building Robust Simulation-Based Filters for Evolving Data Sets”, (unpublished)Department of Statistics, Oxford University, 1999

[29]. R. J. Urick, “Principles of Underwater Sound for Engineers”, McGraw Hill Inc. USA, 1967

[30]. P. C. Etter, “Underwater Acoustic Modeling and Simulation”, Third Edition, Spon Press, USA, 2003

[31]. A. D. Waite, “SONAR for Practising Engineers”, Third Edition, John Wiley & Sons Inc., 2002

[32]. W. D. Wilson, “Equation for The Speed of Sound in Sea Water”, Journal of Acoustic Society of America, Vol.32, No.10, p.1357, 1960

[33]. C. C. Leroy, “Development of Simple Equations for Accurate and More Realistic Calculation of the Speed of Sound in Seawater”, Journal of Acoustic Society of America, Vol.46, pp:212-226, 1969

[34]. V. A. Del Grosso, “New Equation for The Speed of Sound in Natural Waters (With Comparisons to Other Equations)”, Journal of Acoustic Society of America, Vol. 56, pp:1084-1091, 1974

[35]. K.V. Mackenzie, “Nine Term Equation for Sound Speed in the Oceans”, Journal of Acoustic Society of America, Vol. 70, pp:807-812, 1981

[36]. H. Medvin, “Speed of Sound in Water: A Simple Equation For Realistic Parameters”, Journal of Acoustic Society of America, Vol.58, pp:1318-1319, 1975

[37]. L. E. Kinsler, A. R. Frey, A. B. Coppens, J. V. Sanders, “Fundamentals of Acoustics”, Third Edition, John Wiley & Sons Inc., New York, 1982

[38]. C. Viala, C. Noel, G. Lapierre, “RAYSON : a real time underwater communication simulator and performance estimator”

[39]. H. Amiri, H. Amindavar, M. Kamarei, “Underwater Noise Modeling and Direction-Finding Based on Heteroscedastic Time Series”, EURASIP Journal on Advances in Signal Processing, vol. 2007, Article ID 71528, 10 pages, 2007

[40]. G. S. Kino, “Acoustic waves: Devices, Imaging and Analog Signal Processing”, Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1987

[41]. N. Yang, W. Tian, Z. Jin, “An Interacting Multiple Model Particle Filter for Manoeuvring Target Location”, Institue of Physics Publishing, Measurement Science and Technology, vol. 17, pp:1307-1311, 2006