BEARINGS-ONLY TRACKING

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

BY

HALUK ERDEM BİNGÖL

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

FEBRUARY 2011

Approval of the thesis:

BEARINGS-ONLY TRACKING

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

Prof. Dr. Canan Özgen _____
Dean, Graduate School of **Natural and Applied Sciences**

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

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

Examining Committee Members

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

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

Prof. Dr. Erol Kocaoğlan _____
Electrical and Electronics Engineering Dept., METU

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

Dr. H. Burak Kaygısız _____
Intelligent Systems Division, KAREL

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: Haluk Erdem BİNGÖL

Signature :

ABSTRACT

BEARINGS-ONLY TRACKING

Bingöl, Haluk Erdem
M.Sc., Department of Electrical and Electronics Engineering
Supervisor: Prof. Dr. Mübeccel Demirekler

FEBRUARY 2011, 102 Pages

The basic problem with angle-only or bearings-only tracking is to estimate the trajectory of a target (i.e., position and velocity) by using noise corrupted sensor angle data. In this thesis, the tracking platform is an Aerial Vehicle and the target is simulated as another Aerial Vehicle. Therefore, the problem can be defined as a single-sensor bearings only tracking. The state consists of relative position and velocity between the target and the platform. In the case where both the target and the platform travel at constant velocity, the angle measurements do not provide any information about the range between the target and the platform. The platform has to maneuver to be able to estimate the range of the target. Two problems are investigated and tested on simulated data. The first problem is tracking non-maneuvering targets. Extended Kalman Filter (EKF), Range Parameterized Kalman Filter and particle filter are implemented in order to track non-maneuvering targets. As the second problem, tracking maneuvering targets are investigated. An interacting multiple model (IMM) filter and different particle filter solutions are designed for this purpose. Kalman filter covariance matrix initialization and regularization step of the regularized particle filter are discussed in detail.

Keywords: Bearings-only tracking, Kalman Filter,Modified Spherical Coordinates, Particle Filter, Multiple Model

ÖZ

KERTERİZ-AÇISI İZLEME

BİNGÖL, Haluk Erdem
Yüksek Lisans., Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Mübeccel Demirekler

ŞUBAT 2011, 102 Sayfa

Kerteriz açısı izlemenin temel sorunu hedefin gezingesini ve hız profilini hatalı ölçüm yapan sensörün verdiği açı verisini kullanarak tahmin etmektir. Bu tezde izleme platformu bir hava aracıdır, hedef ise başka bir hava aracı olarak sentetik edilmektedir. Bu bağlamda, problem tek-sensörle kerteriz açısı izleme olarak tanımlanabilir. Durum değişkenleri hedef ve platform arasındaki nispi pozisyon ve hızdan oluşmaktadır. Hedefin ve platformun sürekli hızda ilerlediği durumlarda açı ölçüleri hedef ve platform arasındaki menzil hakkında herhangi bir bilgi vermemektedir. Hedefe olan menzilin tahmin edilebilmesi için platformun manevra yapması gerekmektedir. Bu çerçevede, iki ayrı problem araştırılmakta ve simüle veri kullanarak test edilmektedir. İlk problem manevrasız veya çok az menevra yapan hedeflerin izlenmesidir. Manevra yapmayan hedefleri izlemede Genişletilmiş Kalman Filtre (EKF), Menzil Parametreli Kalman Filtre ve parçacık filtresi uygulanmaktadır. Tez kapsamında ikinci bir problem olarak manevra yapan hedeflerin izlenmesi araştırılmaktadır. Bu bağlamda, etkileşimli çoklu model (IMM) filtresi ve farklı parçacık filtreleri tasarlanmıştır. Kalman filtresi kovaryans matrisi ilkleme problemi ve düzenleştirilmiş parçaçık filterisinde düzenleştirme adımı ayrıntılı olarak incelenmiştir.

Anahtar Kelimeler: Kerteriz açısı izleme, Kalman Filtre, Modifiye Küresel Koordinatlar, Parçacık Filtresi, Çoklu Model

To my family,

To my love

ACKNOWLEGMENTS

I am most thankful to my supervisor Prof.Dr. Mübeccel Demirekler for sharing her invaluable ideas and experiences on the subject of my thesis. Thanks to her advices and helpful criticisms, this thesis is completed.

I would like to extend my thanks to all lecturers at the Department of Electrical and Electronics Engineering, who greatly helped me to store the basic knowledge onto which I have built my thesis.

I am very grateful to TÜBİTAK-SAGE for providing tools and other facilities throughout the production of my thesis.

I would like to forward my appreciation to all my friends and colleagues who contributed to my thesis with their continuous encouragement.

I would like to express my deep gratitude to my family, who has always provided me with constant support and help. My dear mother, I have always felt your endless love and best wishes with me.

Special thanks to my love for all her help and showing great patience during the writing process of my thesis. I have always felt her endless support with me.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

CAR-EKF : Cartesian Coordinates Extended Kalman Filter

CV : Constant Velocity

EKF : Extended Kalman Filter

IMM : Interacting Multiple Model

MSC: Modified Spherical Coordinates

MPF : Marginalized Particle Filter

MCMC : Markov Chain Monte Carlo

MISE : Mean Integrated Square Error

MM-PF : Multiple Model Particle Filter

MSC-EKF : Modified Spherical Coordinates Extended Kalman Filter

PF : Particle Filter

PDF : Probability Density Function

RPF: Regularized Particle Filter

RP-EKF : Range Parameterized Extended Kalman Filter

RTAMS : Root Time Average Mean Square

SIR : Sampling Importance Resampling

SMCM : Sequential Monte Carlo Methods

UKF : Unscented Kalman Filter

CHAPTER 1

INTRODUCTION

Bearings only tracking has drawn some attention in the last decades due to its practical value as well as some of its theoretical aspects like unobservability of its states. The focus of this work is to track a single target with a single sensor using only bearings angle. In this chapter, first we will present the background information and the existing work in the literature. Thereafter we will formulate the problem and mention the limitations and the context of the thesis work. Finally, the thesis is outlined.

1.1 Background

The problem of bearings-only tracking is encountered in several important practical applications, since the use of passive sensors has the advantage of not revealing the position of its own platform. Submarine tracking, aircraft surveillance and electronic warfare are some of the important applications in bearings-only tracking [21, 25, 33]. The problem is also named as target motion analysis.

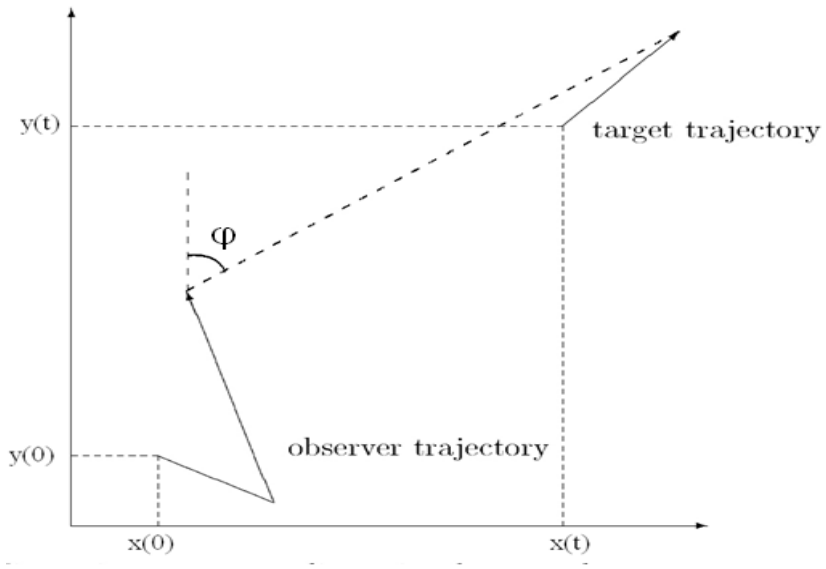

Figure 1-1 A Typical two-dimensional target-observer geometry

The aim of bearings-only tracking problem is to track the kinematics (position, velocity, acceleration) of the target using noisy measurements. Measurements are angle values between the observer and the target platform. A Typical two-dimensional target-observer geometry illustrating the bearings-only tracking problem is shown in Figure 1-1.

In the case of single observer bearings-only tracking problem, which is the focus of this thesis, same measurements can be obtained for different range values. Consider the case where the target moves at a constant velocity (Figure 1-2). The lines tr_1 and tr_2 represent possible targets that produce exactly the same bearing measurements. The figure shows three angle measurements that have been received, represented by three lines, namely y_1, y_2 and y_3. Two possible target trajectories, tr_1 and tr_2, are also depicted. The target can either be close to the sensor and move slowly or be far away and move fast. From the figure it is clear that infinite number of tracks can be generated by using only the bearing angle sequence.

“

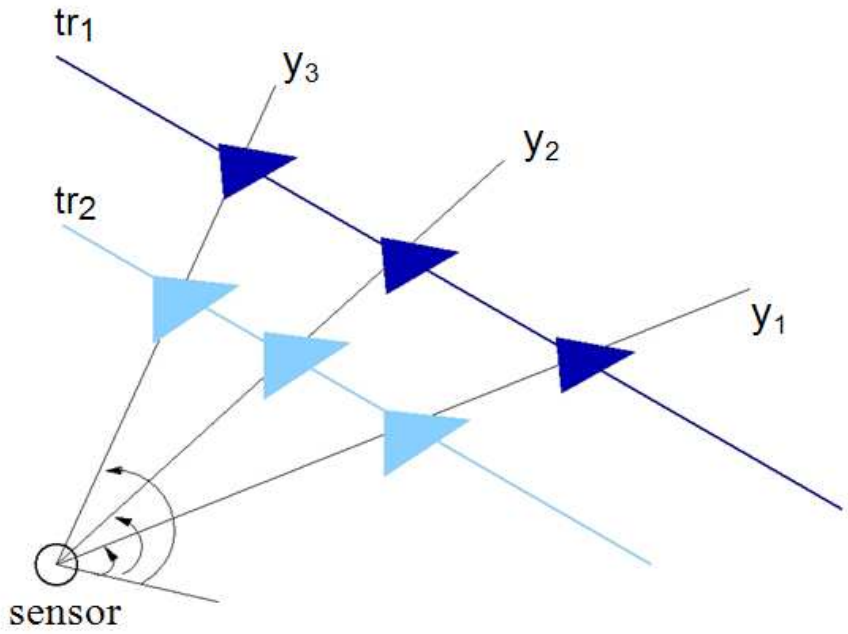

Figure 1-2 Single-sensor case .Two possible trajectories, tr_1 and tr_2 corresponding to the same angle measurement

Tracking a maneuvering target using only bearings measurements is much more difficult. Figure 1-3 demonstrates from which the two trajectories system receive the same measurements as in Figure 1-2, but now the problem is how to distinguish a maneuvering target from a non-maneuvering target. In a target tracking system with full observability an inaccurate model of the target’s dynamics can be compensated with accurate measurements. In case of unobservable range it is essential to have accurate dynamic models. Since the measurements do not provide the system with information of the range, the system must rely on the model. On the other hand it is hard to design a model that describes the dynamics of both maneuvering and non-maneuvering targets. Instead it is common to use multiple models, where each model describes one possible target maneuver (including no maneuver).

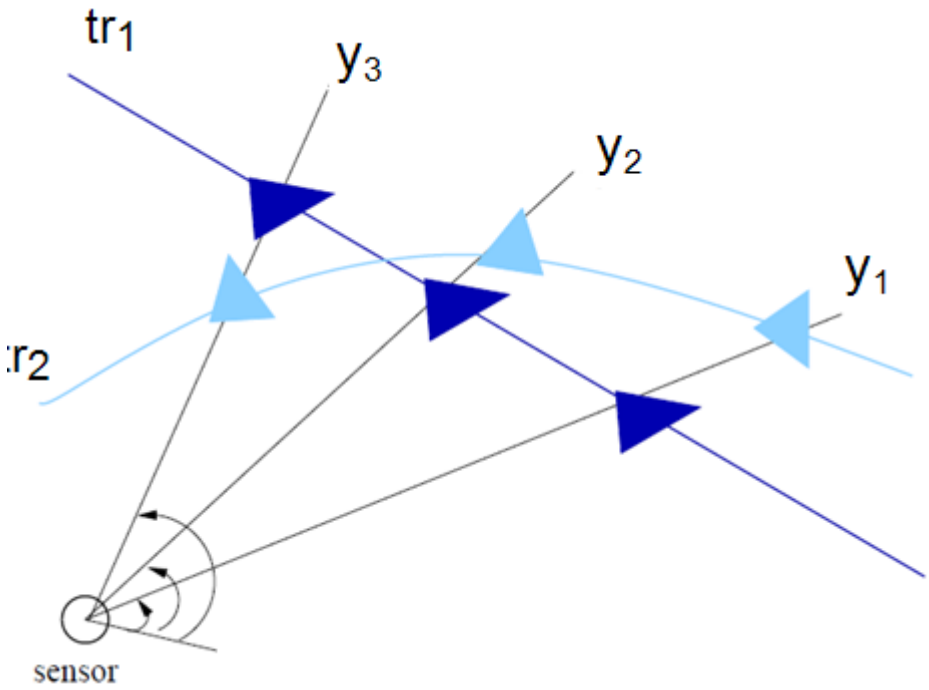

Figure 1-3 Two possible trajectories, tr_1 and tr_2 corresponding to the same angle measurement.

1.2 Literature Survey

The study on bearings-only tracking problem can be classified into two classes, namely non-maneuvering target tracking and maneuvering target tracking.

Most of the researchers in the field of bearings-only tracking have concentrated on the non-maneuvering target tracking (constant velocity). These solutions/algorithms that they provide can mainly fall into two categerios: batch processing type and recursive type. Since the batch processing type algorithms solve the problem with a delay and are computationally demanding, they are not studied in this work. Among recursive type algorithms, the Extended Kalman Filter (EKF) is one of the most widely used methods [1]. EKF in modified spherical coordinates (MSC) [2,3] which defines the problem in another coordinate system is the most popular refinement. Shifted Rayleigh Filter is another Extended Kalman Filter modification [4]. Modified Gain Exteneded Kalman Filter is also applied to bearings-only tracking problem [5]. Besides EKF, Unscented Kalman

Filter (UKF) solution is proposed in [6,7]. It is claimed in these studies that UKF is much more robust than the EKF.

The EKF variants mentioned above can only track a single mode of the posterior probability distrubition function (pdf) of the target state and thus suffer from the same problem as the EKF, i.e., they can potentially track an inaccurate mode of the posterior pdf and hence become inconsistent [8]. To mitigate the aforementioned issue, running parallel filters was proposed in [8] to track multiple hypotheses of the target state, namely the Range-Parameterized Extended Kalman Filter (RP-EKF). The RP-EKF makes an assumption about the minimum and maximum distances between the sensor and target and divides this range interval into a number of subintervals, each representing a hypothesis regarding the true range of the target. A bank of independently operating range-parameterized EKFs is thus created, each designed for one of the hypotheses and receiving the same bearing measurement. Recently, considerable attention has been paid to the Sequential Monte Carlo Methods (SMCM) [9, 10, 11], due to its capability of solving nonlinear estimation problems with multimodal pdfs. In the SMCM, each particle represents a hypothesis of the target state, weighted by its measurement likelihood. If the particles sample the state space sufficiently, the SMCM will converge to the true distribution. However, the particles are usually initialized randomly, and if far from a mode of the pdf, their weights will decay quickly and lead to particle depletion.

For the bearings-only tracking of a maneuvering target the problem is much more diffucult and less study has been published in the literature in this regard. Interacting Multiple Model Filter was proposed in [12, 13]. The idea behind multiple model is to use multiple filters running in parallel, where each filter uses a model describing one possible maneuver. Le Cadre and Tremois [13] modeled the maneuvering target tracking using the contant velocity (CV) model with process noise and developed a tracking filter in the hidden Markov model framework. Particle filter algorithms are also proposed for tracking maneuvering targets. In [15] Karlsson e.a. propose an Auxiliary Particle Filter for maneuvering target tracking. Also, in [16] Andrieu e.a. use Marginilized Particle Filter. This study is interesting from the point of using the optimal importance density.

The observer manuever is another problem which is not studied well enough in the literature. In [17] Jauffret e.a. examine the cases for which maneuver of the observer is necessary and/or sufficient to obtain observability in passive target tracking. They show that it is necessary for the observer platform to maneuver with non-zero acceleration in the case of a target of constant speed, but this is not sufficient. The platform must have a non-zero velocity component perpendicular to the line of sight. Necessary and sufficient conditions for observability are given in [18]. In [19] platform maneuvers are statistically analyzed in accordance to their affect on the target tracking. Optimal control theory which is applied for determination of optimal observer maneuver is discussed in [19].

Another area that the problem of bearings-only target tracking arises is Simultaneous Localization And Mapping (SLAM). Here the problem is different from bearings-only target tracking since the “target” does not move. Little work has been presented regarding bearing-only SLAM due to the difficulty of feature initialization [34, 35]. Reference [34] presents a solution to the feature initialization. Cartesian EKF solution is used for bearings only problem used. In Reference [35] Deans e.a. study the multiple sensor case problem again using EKF.

1.3 The Scope of the Thesis

In this thesis, a general framework is given for the bearings-only tracking. Two different problems are investigated. The first problem is to track a constant velocity target. For this problem different methods are simulated and their performances are compared. The methods include Extended Kalman Filter using Cartesian Coordinates (CAR-EKF), Extended Kalman Filter using Modified Spherical Coordinates (MSC-EKF), a bank of Extended Kalman Filters, Range-Parameterized Extended Kalman Filter (RP-EKF) and particle filter (PF).

The second problem is tracking a maneuvering target. This problem is relatively less analyzed in the literature compared to the first one. As the solution to this problem IMM and PF methods are implemented.

1.4 Outline of the Thesis

In Chapter 2, the basis of bearings-only target tracking is presented and the observability problem is discussed. Thereafter target and sensor models and the coordinate systems used in the thesis are presented. In Chapter 3, the theory related to bearings only tracking is explained. In Chapter 5, firstly methodology of performance evaluation is discussed and several measures to be used in simulations are presented. After that implementation of the filters is described, the simulation scenarios are presented and the results are discussed. In chapter 6, the thesis is summarized and conclusions from the simulations as well as ideas for the future work are presented.

The theory that has been omitted in the main text is included in appendices A and B.

CHAPTER 2

BEARINGS ONLY TARGET TRACKING

The problem with bearings-only target tracking is to estimate target position with unknown parameters of its dynamics using angle measurements between the observer and the target. The state primarily consists of position and velocity of a target or variables related with them for example the range. It is possible to include other quantities into the state as well, such as angular velocity, turning rate, turn mode, etc. according to the model that is used.

This chapter starts with explaining target motion in two different coordinate systems, namely Cartesian coordinates and Modified Spherical Coordinates (MSC). Thereafter target's dynamics for constant velocity (CV) are investigated. Finally, dynamics of coordinated turn model are discussed.

2.1 Coordinate Systems

2.1.1 Cartesian Coordinate System

A common choice of coordinate system in any tracking application is to use Cartesian coordinates as illustrated in Figure 2-1 in which the observer is assumed to be an airplane. The x-axis is pointing through the nose of the observing platform, the y-axis points through the left wing and the z-axis points through the upper part of the observer. The state vector in 3D Cartesian coordinates is denoted by x_{car} and is given by:

$$X_{car} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} \xi \\ \eta \\ \varsigma \\ \dot{\xi} \\ \dot{\eta} \\ \dot{\varsigma} \end{bmatrix} \tag{2.1}$$

Here, instead of x-y-z notation ξ-η-ζ is used avoiding confusion between state variable, measurement and position [1].

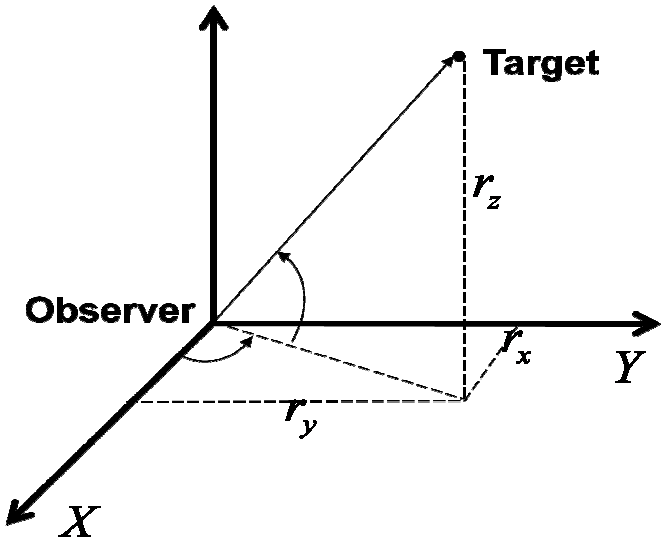

Figure 2-1 3D Cartesian Coordinates

For the 2D case z components are dropped.

2.1.2 Modified Spherical Coordinates

It has been shown that the estimation algorithms for the angle-only target motion analysis (TMA) problem formulated in Cartesian coordinates have resulted in unstable and biased estimates [2]. Specifically, the Cartesian coordinate EKF exhibits filter divergence while the Pseudo-Linear estimator shows biased characteristics. To overcome these difficulties, an extended Kalman filter was

proposed in [2], which was formulated in a different coordinate system called the modified spherical coordinates (MSC). This coordinate system was shown to be well suited for angle-only target motion analysis because it automatically decouples observable and unobservable components of the estimated state vector. Such decoupling prevents covariance matrix to be an ill-conditioned matrix, which is the primary cause of filter instability. The MSC state vector is comprised of the following six components: azimuth angle, azimuth angle rate, elevation angle, elevation angle rate, range rate divided by range, and the reciprocal of the range. MSC system is illustrated in Figure 2.2, and the state variables are given in Equation 2.2. In theory, the first five states can be determined by using a single-sensor angle data without an observer maneuver; the sixth component, however, remains unobservable until a maneuver occurs. In [2], Aidala et al did not insert the process noise in their formulation. Another derivation of the MP Extended Kalman Filter was proposed in [3], where Singer Model target acceleration noise is modeled in state-space representation.

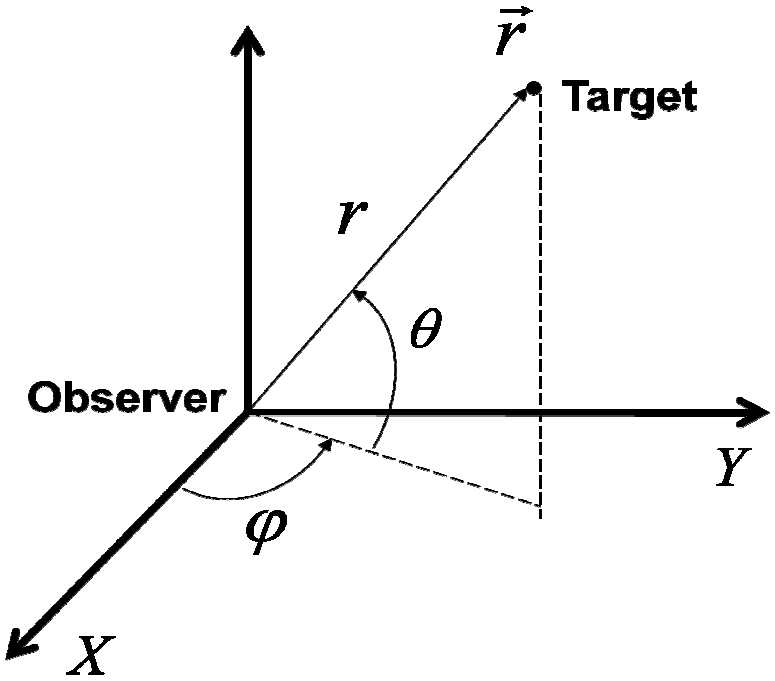

Figure 2-2 Modified Spherical Coordinates

$$Y_{msc} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \dot{y}_1 \\ \dot{y}_2 \\ \dot{y}_3 \end{bmatrix} = \begin{bmatrix} \frac{1}{r} \\ \varphi \\ \theta \\ \frac{\dot{r}}{r} \\ \vartheta \\ \dot{\theta} \end{bmatrix} \tag{2.2}$$

where ϑ is $\dot{\varphi}\cos(\theta)$

2.2 Target Dynamics

Models, which describe targets' dynamics, are intrinsically continuous. Besides, differential equations are used generally for describing these dynamics. Consider a general dynamic model written in the state space form as follows:

$$\dot{x}(t) = f_c(x(t), u(t), w(t)) \tag{2.3}$$

where *x(t)* is the state of the system. *u(t)* is a known input vector to the system and *w(t)* is the noise. For our case *u(t)* may correspond to the observing platform's maneuver that is assumed to be known without errors. How the state vector propagates in time is described by the function *f*. Process noise which is denoted by *w(t)* corresponds to non modeled part of the motion of the target.
In discrete time the state equation can be written as:

$$x_{k+1} = f_d(x_k, u_k, w_k) \tag{2.4}$$

2.2.1 Constant Velocity State Equations

Model of a target whose relative motion is approximately linear, i.e. the relative velocity is constant, is given in this section.

For a system modeled in Cartesian coordinates the constant velocity discrete time state equation is written as:

$$x_{k+1}^{car} = F_k x_k^{car} + G_k w_k \tag{2.5}$$

$$x_{k+1}^{car} = \begin{bmatrix} 1 & 0 & 0 & T & 0 & 0 \\ 0 & 1 & 0 & 0 & T & 0 \\ 0 & 0 & 1 & 0 & 0 & T \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} x_k^{car} + \begin{bmatrix} T^2/2 & 0 & 0 \\ 0 & T^2/2 & 0 \\ 0 & 0 & T^2/2 \\ T & 0 & 0 \\ 0 & T & 0 \\ 0 & 0 & T \end{bmatrix} w_k \tag{2.6}$$

As seen from the formulation, state equations are linear and time invariant. Measurements of the system are the bearing angles which are modeled as:

$$z_k = \begin{bmatrix} \varphi \\ \theta \end{bmatrix} = \begin{bmatrix} \arctan(\frac{x_2}{x_1}) \\ \arctan(\frac{-x_3}{\sqrt{x_1^2 + x_2^2}}) \end{bmatrix} \tag{2.7}$$

Here the extended definition of arctan(x) which describes angles in all four quadrants is used. The observables, namely φ and θ, are coupled with the unobservable range. As seen from Equation 2.7 measurement equation it is highly nonlinear.

In MSC the state equation can be written as [3]:

$$
\begin{aligned}
\dot{y}_1 &= -y_1 y_4 \\
\dot{y}_2 &= \frac{y_5}{\cos(y_3)} \\
\dot{y}_3 &= y_6 \\
\dot{y}_4 &= y_6^2 + y_5^2 - y_4^2 - y_1(a_{Rt} - a_{Ro}) \\
\dot{y}_5 &= y_5(-2y_4 + y_6 \tan(y_3)) - y_1(a_{Ht} - a_{Ho}) \\
\dot{y}_6 &= -2y_4 y_6 - y_5^2 \tan(y_3) + y_1(a_{Vt} - a_{Vo})
\end{aligned}
\tag{2.8}
$$

where a_{Rt}, a_{Ht} and a_{Vt} are target acceleration in radial, horizontal and vertical directions and a_{Ro}, a_{Ho} and a_{Vo} are the corresponding observer platform acceleration.

The derivation of plant equations for MSC state equation can be found in Appendix A.

It can be seen from Equation (2.8) that when neither the target nor the platform maneuvers, the last five states are decoupled from the first. This means that theoretically all states except the first (the inverse of range) may be observable. In practice, however, nonzero angular velocity is required to observe the fourth state, namely $\dot{r}/r$ according to [21].

The advantage of implementing a target tracking filter in MSC is that the lack of range estimate does not degrade the estimates of the observable states in the case of no maneuver [2]. If the target is not maneuvering, an accurate range estimate is produced by using the maneuver of the observer. The disadvantage of using MSC is that solving and discretization of Equation 2.8 will yield a highly nonlinear dynamic model.

The measurement equation in MSC is:

$$
z_k = \begin{bmatrix} \varphi \\ \theta \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} y_K \tag{2.9}
$$

2.2.2 Constant Velocity Process Noise

Process noise of constant velocity model is usually taken as the acceleration of the target that can be modeled in several ways such as white-noise model, polynomial model, Singer model etc. [22]. We used Singer model in the simulations since for manned aerial vehicles it is the one that is most commonly used.

The Singer model assumes [23] that the target acceleration *a(t)* is a zero-mean first-order stationary Markov process with autocorrelation function $R_a(\tau) = E(a(t+\tau)a(t)) = \sigma^2 e^{-\alpha|\tau|}$ or equivalently the power spectrum of the acceleration is $S_a(w) = 2\alpha\sigma^2/(\omega^2 + \alpha^2)$. The process *a(t)* can be modeled as the state process of a linear time-invariant system driven by white noise as follows.

$$\dot{a}(t) = -\alpha a(t) + w(t) \quad \alpha > 0 \tag{2.10}$$

The state space representation of the continuous time Singer model for one dimension is as follows.

$$\dot{x}(t) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -\alpha \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} w(t) \tag{2.11}$$

Its discrete time equivalent is

$$x_{k+1} = F_\alpha x_k + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} w_k = \begin{bmatrix} 1 & T & (\alpha T - 1 + e^{-\alpha T})/\alpha^2 \\ 0 & 1 & (1 - e^{-\alpha T})/\alpha \\ 0 & 0 & e^{-\alpha T} \end{bmatrix} x_k + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} w_k \tag{2.12}$$

In this equation w_k is a zero mean white noise sequence with variance $\sigma^2(1-\beta^2)$ where β is $e^{-\alpha T}$. The exact covariance of w_k is a function of α, T, σ which is:

$$
\begin{aligned}
Q(k) &= 2\alpha\sigma^2 \begin{bmatrix} q_{11} & q_{12} & q_{13} \\ q_{12} & q_{22} & q_{23} \\ q_{13} & q_{23} & q_{33} \end{bmatrix} \\
q_{11} &= \frac{1}{2\alpha^5}\left[1 - e^{-2\alpha T} + 2\alpha T + \frac{2\alpha^3 T^3}{3} - 2\alpha^2 T^2 - 4\alpha T e^{-\alpha T}\right] \\
q_{12} &= \frac{1}{2\alpha^4}\left[e^{-2\alpha T} + 1 - 2e^{-\alpha T} + 2\alpha T e^{-\alpha T} - 2\alpha T + \alpha^2 T^2\right] \\
q_{13} &= \frac{1}{2\alpha^3}\left[1 - e^{-2\alpha T} - 2\alpha T e^{-\alpha T}\right] \\
q_{22} &= \frac{1}{2\alpha^3}\left[4e^{-\alpha T} - 3 - e^{-2\alpha T} + 2\alpha T\right] \\
q_{23} &= \frac{1}{2\alpha^2}\left[e^{-2\alpha T} + 1 - 2e^{-\alpha T}\right] \\
q_{33} &= \frac{1}{2\alpha}\left[1 - 2e^{-\alpha T}\right]
\end{aligned}
\tag{2.13}
$$

The derivation of Equation (2.13) can be found in [23].

2.2.3 Coordinated Turn Model with Known Turn Rate

For maneuvering targets coordinated turn (CT) model is the most common model [13, 24, 25]. In this model, it is assumed that the target turns at a nearly constant angular velocity. In addition, the magnitude of the velocity vector is almost constant. The maneuver occurs in a plane. However, it does not mean that it has to be in *x-y* plane. For example, if maneuver plane is vertical, this leads to climbs and descents. Therefore, CT can actually model this maneuver, too. States in this model are the same as the Cartesian coordinates for non-maneuvering case. State transition matrix for CT model in 2-d coordinate system is shown in Equation 2.14-2.15.

$$
x_{k+1}^{car} = F_k^{CT} x_k^{car} + G^{CT} w_k \tag{2.14}
$$

$$x_{k+1}^{car} = \begin{bmatrix} 1 & 0 & \frac{\sin(\Omega T)}{\Omega} & -\frac{1-\cos(\Omega T)}{\Omega} \\ 0 & 1 & \frac{1-\cos(\Omega T)}{\Omega} & \frac{\sin(\Omega T)}{\Omega} \\ 0 & 0 & \cos(\Omega T) & -\sin(\Omega T) \\ 0 & 0 & \sin(\Omega T) & \cos(\Omega T) \end{bmatrix} x_k^{car} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} w_k \qquad (2.15)$$

where

$$Q = \mathrm{cov}(w_k) = \begin{bmatrix} \frac{2(\Omega T - \sin\Omega T)}{\Omega^3} & 0 & \frac{1-\cos\Omega T}{\Omega^2} & \frac{\Omega T - \sin\Omega T}{\Omega^2} \\ 0 & \frac{2(\Omega T - \sin\Omega T)}{\Omega^3} & \frac{\Omega T - \sin\Omega T}{\Omega^2} & \frac{1-\cos\Omega T}{\Omega^2} \\ \frac{1-\cos\Omega T}{\Omega^2} & -\frac{(\Omega T - \sin\Omega T)}{\Omega^2} & T & 0 \\ \frac{(\Omega T - \sin\Omega T)}{\Omega^2} & \frac{1-\cos\Omega T}{\Omega^2} & 0 & T \end{bmatrix} \qquad (2.16)$$

Turn rate Ω can be written as a nonlinear function of acceleration:

$$\Omega = \frac{\pm a_m}{\sqrt{\dot{x}_k^{\,2} + \dot{y}_k^2}} \qquad (2.17)$$

In Equation 2.17 $\pm a_m$ is the typical maneuver acceleration of the target. Different nonlinear models can be constructed for different values of a_m.

The coordinated turn model can give accurate results if the turn rate is known. However, this case is unrealistic for most of the practical applications. In this study we also studied unknown turn rate case which is much more realistic.

2.2.4 Coordinated Turn Model with Unknown Turn Rate

Coordinated turn model with unknown turn rate differs from the model above with the inclusion of the turn rate as a state variable [22]. The state transition matrix is given in Equation 2.18.

$$x_{k+1} = \begin{bmatrix} F^{CT}(\omega) & 0 \\ 0 & e^{-\tau/T} \end{bmatrix} x_k + \begin{bmatrix} G^{CV} & 0 \\ 0 & 1 \end{bmatrix} w_k \qquad (2.18)$$

The most commonly used models for ω are Wiener process model and first order Markov process model. We have used Markov process model in this study. The discrete time equation for turn rate is described in Equation 2.19.

$$\omega_{k+1} = e^{-\tau/T}\omega_k + w_{w,k} \tag{2.19}$$

where τ is the correlation time constant for the turn rate and w is zero-mean white noise.

2.2.5 Coordinated Turn Model with Polar Velocity and Unknown Turn Rate

Since the velocity vector is assumed to be nearly constant, using polar velocity instead of Cartesian velocities will simplify the transition matrix. Denote the magnitude of velocity as v and the heading angle (course) as φ. The new state vector becomes as given in Equation 2.19.

$$x = [x_1 \quad x_2 \quad V \quad \varphi \quad \Omega]^T \tag{2.20}$$

The discrete time state equation then becomes:

$$x_{k+1} = \begin{bmatrix} x_1 + (2/\Omega)v\sin(\Omega T/2)\cos(\varphi + \Omega T/2) \\ x_2 + (2/\Omega)v\sin(\Omega T/2)\sin(\varphi + \Omega T/2) \\ v \\ \varphi + \Omega T \\ \Omega \end{bmatrix} + G^{CTP} w_k^{CTP} \tag{2.21}$$

Where w_k is the corresponding process noise with the covariance matrix given below.

$$Q^{CTP} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & T^2\sigma_v^2 & 0 & 0 \\ 0 & 0 & 0 & T^3\sigma_\Omega^3/3 & T^3\sigma_\Omega^2/2 \\ 0 & 0 & 0 & T^3\sigma_\Omega^2/2 & T^3\sigma_\Omega^2/2 \end{bmatrix} \quad (2.22)$$

2.2.6 Sensor Error Model

Sensor model in the simulations are implemented as given by Equation 2.23:

$$z_m = z_t + v_k \quad (2.23)$$

where z_m is the measured signal, z_t is its true value and v_k is the measurement noise which is assumed to have a probability distribution function *N(0,R)*. *R* is usually selected as a diagonal matrix (i.e. azimuth and elevation measurements are independent from each other).

CHAPTER 3

TRACKING FILTERS

In target tracking the aim is to estimate the target's state vector by measurements. This chapter starts with filters designed to track non-maneuvering targets. For maneuvering targets multiple model filters are often used, which are presented at the end of the chapter.

3.1 General Information

In this thesis the aim is to obtain probablity distrubition (usually it is expressed as an estimate and its uncertainity) of the unknown state vector by using noisy measurements. Figure 3-1 presents a block diagram that illustrates the state estimation [25].

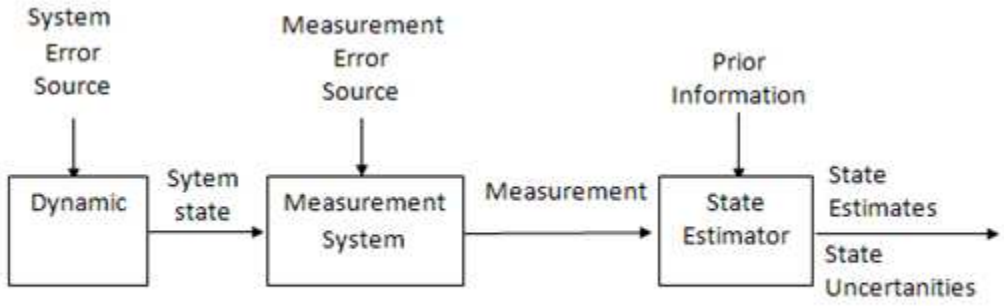

Figure 3-1 State Estimation (Copied from,[25])

A general state space model for a discrete time stochastic system can be defined as given by Equation 3.1 and Equation 3.2:

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

$$z_k = h_k(x_k, v_k) \tag{3.2}$$

Where:

x_k : state at time *k*

z_k : measurement at time *k*

u_k : known input to the dynamic system at time *k*

w_k : process noise at time *k*

v_k : measurement noise at time *k*

The aim is to estimate the posterior distribution of the state vector given past observations for the above model. The Bayesian solution can be written as:

$$\begin{aligned} p(x_k \mid z_{1:k}) &= \frac{p(z_k \mid x_k)\, p(x_k \mid z_{1:k-1})}{p(z_k \mid z_{1:k-1})} \\ p(z_k \mid z_{1:k-1}) &= \int_{R^{n_x}} p(z_k \mid x_k)\, p(x_k \mid z_{1:k-1}) dx_k \\ p(x_{k+1} \mid z_{1:k}) &= \int_{R^{n_x}} p(x_{k+1} \mid x_k)\, p(x_k \mid z_{1:k}) dx_k \end{aligned} \tag{3.3}$$

This update recursion is the optimal solution for non-linear filtering estimation problem. However, usually it is impossible to obtain the solution analytically as in our problem. Therefore, some suboptimal filters are used which are described in the rest of this chapter.

3.2 Non-Maneuvering Target Tracking Filters

3.2.1 Kalman Filter

For a linear model with additive noise the system given by the Equations 3.1 and 3.2 can be rewritten as:

$$x_{k+1} = Ax_k + Bu_k + Gw_k \tag{3.4}$$

$$y_k = Hx_k + v_k \tag{3.5}$$

where A, B, G *and* H are matrices, possibly depending on time, but independent of the state vector x.

Assume that the noises $\{w_k, v_k\}$ are all independent with Gaussian distribution with zero mean and covariance matrices $Cov(w_k) = Q_k$ and $Cov(v_k) = R_k$. Also, assume that the initial distribution of the state vector is Gaussian with mean x_0 and covariance matrice $Cov(x_0) = P_0$. Under these assumptions it can be shown that the optimal solution to the problem of estimating the state vector x based on the observations y is given by the Kalman Filter [11]. Below, we will give a very brief description of the Kalman Filter.

Let $\hat{x}_{k|k}$ be the estimate of the state vector for time t_k given the measurements up to time t_k, $Z^K=(z_1, z_2, ..., z_k)$. In the same manner let $\hat{x}_{k|k-1}$ denote the estimate of the state vector at time t_k based on the measurements up to time t_{k-1}. Let $\hat{P}_k$ be the estimate of the prediction covariance matrix for the state vector at time k. $\hat{P}_k$ represents the uncertainty in the estimated state vector.

$\hat{F}_k$ does not depend on the measurements Z^k meaning that it will not give any information on how well the estimates fits the measurements. Instead $\hat{F}_k$ gives a theoretical value on how well the filter performs if the assumptions of Gaussian

noise are valid. The parameter *K*, known as the Kalman gain, decides how much the innovation $\varepsilon_k = z_k - H\hat{x}_{k|k-1}$ should be taken into consideration while estimating the state. Large values of $\hat{P}_k$ means that the uncertainty in the estimate of the state is large. This causes *K* to be large, so the measurement will have a great impact on the correction of the estimate. This will give a fast filter that considers the measurements to be reliable. Small values of $\hat{P}_k$ and K on the other hand results in a slower filter that is more robust against measurement noise. $\hat{P}_k$ depends on its initiatialization and of the values in the covariance matrices Q and R. The designer of the filter has to balance the robustness to measurement noise against the fastness of the filter by choosing appropriate values for the Q and R matrices.

The Kalman filter is the optimal solution under the assumptions of Gaussian noise. In real applications the assumption of Gaussian noise might not be valid. However assuming that the specified covariance matrices, *Q,* R and $\hat{P}_k$ reflects the true second order moments is the best linear filter. A more detailed description of the Kalman Filter can be found in [9,26]. A single cycle of Kalman Filter is described in Table 3-1.

Table 3-1 One Cycle of Kalman Filter

1. **Time Update**

Predict the state vector and the prediction covariance matrix for the next time step

$$\hat{x}_{k+1|k} = A\hat{x}_{k|k} + Bu_k$$
$$\hat{P}_{k+1|k} = A\hat{P}_{k|k}A^T + Q_k$$

2. **Measurement Update**

Correct the predicted state vector and prediction covariance matrix according to the measurement

$$\varepsilon_k = z_k - H\hat{x}_{k|k-1}$$
$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K\varepsilon_k$$
$$\hat{P}_{k|k} = \hat{P}_{k|k-1} - KH\hat{P}_{k|k-1}$$
$$K = \hat{P}_{k|k-1}H^T S^{-1}$$

Where *S* is :

$$S = H\hat{P}_{k|k-1}H^T + R_k$$

3.2.2 Extended Kalman Filter

In the case of a non-linear state or measurement equations as in (3.1) and (3.2) an Extended Kalman Filter (EKF) can be used. The idea is to linearize the system and apply the Kalman filter. Unlike the Kalman filter EKF is not an optimal filter. The implemention can be done in several ways, one of which is discretized linearization, in which first the non-linear continuous system is linearized and thereafter the resultant system is discretized. A derivation of the discretized-linearized EKF can be found in [26]. The equations of EKF are presented in Table

3-2. When we compare EKF with KF we observe that the main difference is that the matrices *A* and *H* have been replaced with the Jacobians of the functions *f* and *h* in the update of $\hat{F}$.

Table 3-2 One Cycle of Extended Kalman Filter

1. **Time Update**

Predict the state vector and the prediction covariance matrix for the next time step

$$\hat{x}_{k+1|k} = f(\hat{x}_k, u_k)$$
$$\hat{P}_{k+1|k} = J_f \hat{P}_{k|k} J_f^T + Q_k$$

2. **Measurement Update**

Correct the predicted state vector and prediction covariance matrix according to the measurement

$$\varepsilon_k = z_k - h(\hat{x}_{k|k-1})$$
$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K\varepsilon_k$$
$$\hat{P}_{k|k} = \hat{P}_{k|k-1} - KJ_h \hat{P}_{k|k-1}$$
$$K = \hat{P}_{k|k-1} J_h^T S^{-1}$$

Where S is

$$S = J_h \hat{P}_{k|k-1} J_h^T + R_k$$

J_f and J_h are the Jacobians of the functions *f* and *h* respectivly.

3.2.3 Extended Kalman Filter for Bearings Only Tracking in Cartesian Coordinates for Non-Maneuvering Target Case

When the system is modeled in Cartesian coordinates for the non-maneuvering case, only the measurement equation comes out to be nonlinear. Therefore, only the evaluation of the Jacobian of *h* is required. Jacobian of *h* is shown in Equation (3.7).

Recall from Equation (2.7) that the measurement function in Cartesian Coordinates is:

$$z_k = \begin{bmatrix} \varphi \\ \theta \end{bmatrix} = \begin{bmatrix} \arctan\left(\dfrac{x_2}{x_1}\right) \\ \arctan\left(\dfrac{-x_3}{\sqrt{x_1^2 + x_2^2}}\right) \end{bmatrix} \tag{3.6}$$

Then it can be easily shown that the Jacobian of this function is:

$$J_h = \begin{bmatrix} \dfrac{-x_2}{(x_1^2+x_2^2)} & \dfrac{x_1}{(x_1^2+x_2^2)} & 0 & 0 & 0 & 0 \\ \dfrac{x_3 x_1}{\sqrt{x_1^2+x_2^2}(x_1^2+x_2^2+x_3^2)} & \dfrac{x_3 x_2}{\sqrt{x_1^2+x_2^2}(x_1^2+x_2^2+x_3^2)} & -\dfrac{\sqrt{x_1^2+x_2^2}}{(x_1^2+x_2^2+x_3^2)} & 0 & 0 & 0 \end{bmatrix} \tag{3.7}$$

3.2.4 Extended Kalman Filter for Bearings Only Tracking in Modified Spherical Coordinates for Non-Maneuvering Target Case

The idea behind the MSC-EKF is to use Cartesian coordinates in the time update but use MSC in the measurement update. In this way both the equations of motion and the measurement equation become linear. Let x^{car} be the state vector in the Cartesian coordinates and y^{msc} the state vector in the MSC. The algorithm of this approach is given below by Equation (3.8).

$$\begin{aligned} x_k^{\ car} &= f_x(y_k^{\ msc}) \\ x_{k+1}^{car} &= Ax_k^{car} + Bu_k + Gw_k \\ y_{k+1}^{msc} &= f_z(x_k^{\ car}) \\ z_{k+1} &= Hy_{k+1}^{msc} + v_k \end{aligned} \tag{3.8}$$

The first equation describes the dynamics of the system in Cartesian coordinates and the third equation gives the relation between the measurements and the state vector in MSC. f_x and f_z are functions that transform the state vector between Cartesian coordinates and MSC. The explicit expressions for f_x and f_z as well as their Jacobians can be found in Appendix B.

Figure 3-2 shows the block diagram of the filter. In the blocks *Time update* and *Measurement update* are performed in the same way as in the Kalman filter. The tasks for the blocks *MSC to Car* is to transform the state vector from MSC to Cartesian coordinates while the block *Car to MSC* do the opposite transformation. The algorithm for the MSC-EKF is given in Table 3-3 where the superscripts *car* and *msc* has been suppressed. The prediction covariance matrix *P* and the measurement noise covariance matrix *R* are described in MSC and the process noise covariance matrix *Q* is expressed in Cartesian coordinates.

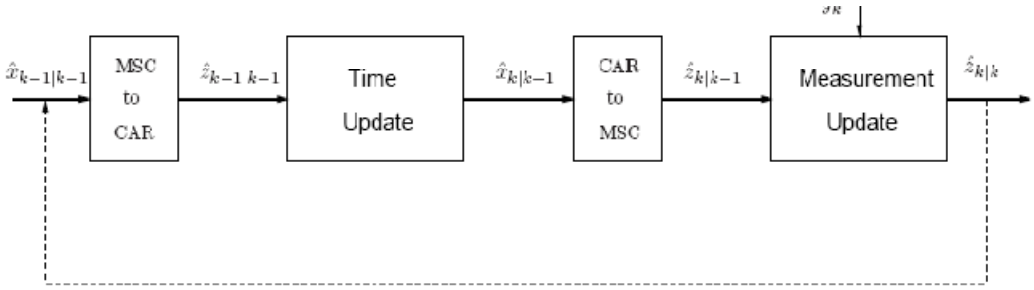

Figure 3-2 Block diagram for an extended Kalman filter using modified spherical coordinates (MSC-EKF)

3.2.5 Range-Parameterized EKF

For a particular EKF, the tracking performance is highly dependent on the Coefficient of Variation of the range estimate C_R given by σ_r/R where R and σ_r are the range estimate and its standart deviation respectively. It is this factor which determines the percentage change in the range estimate for a given change in the bearing estimate and, therefore, governs the stability of the tracker. Practically, for bearings-only tracking problem apriori knowledge of the initial target range can be

very poor. To handle this Range-Parameterized EKF use a a number of EKF trackers in parallel with a different range estimate [16]. By this way C_R is tried to be minimised. Also, C_R is the modelled to be for all filters.

If the range errors are uniformly distrubuted in each subinterval, subintervals can be subdivided using geometrical progression as in shown in Equation (3.9):

$$r_{\max} = r_{\min}\rho^{Nf} \tag{3.9}$$

The weights associated with each EKF are computed using the measurement as given in Equation 3.10.

$$w_k^i = \frac{p(z_k \mid i) w_{k-1}^i}{\sum_{j=1}^{Nf} p(z_k \mid j) w_{k-1}^j} \tag{3.10}$$

Here $\hat{z}_{k|k-1}^i$ denotes the predicted angles at k for filter *i*, and Σ_i is the innovation variance. In this thesis MSC is used for RP-EKF, so Σ_i is can be extracted from the covariance matrix $\hat{P}_{k|k-1}^i$.

Weights are propagated in time using the measurements as seen in Equation 3.10.

Table 3-3 One Cycle for MSC EKF

1. **Transform the state vector to Cartesian coordinates**

$$\hat{x}_{k-1|k-1} = f_x(\hat{z}_{k-1|k-1})$$

2. **Time Update**

$$\hat{x}_{k|k-1} = A\hat{x}_{k-1|k-1} + Bu_k$$
$$\hat{P}_{k|k-1} = \Phi_{k|k-1}\hat{P}_{k-1|k-1}\Phi_{k|k-1}^T + Q_{k-1}^{msc}$$

Φ is given by the chain rule for derivates

$$\Phi_{k|k-1} = \left.\frac{\partial_{z_k}}{\partial_{z_{k-1}}}\right|_{\hat{z}} = \frac{\partial_{z_k}}{\partial_{x_k}}\frac{\partial_{x_k}}{\partial_{x_{k-1}}}\frac{\partial_{x_{k-1}}}{\partial_{z_{k-1}}} = J_{fZ}(\hat{x}_k)AJ_{fX}(\hat{z}_{k-1})$$

J_{fZ} and J_{fX} are the Jacobian for the transformation function f_Z and f_X respectively evaluated in different time steps.

$$Q_{k-1}^{msc} = J_{fZ}(\hat{x}_{k|k})Q_{k-1}J_{fZ}(\hat{x}_{k|k})^T$$

Where Q_{k-1} is expressed in Cartesian coordinates.

3. **Transform the state vector to MSC**

$$\hat{z}_{k|k-1} = f_z(\hat{x}_{k|k-1})$$

4. **Measurement Update**

$$\varepsilon_k = y_k - H\hat{z}_{k|k-1}$$
$$\hat{z}_{k|k} = \hat{z}_{k|k-1} + K\varepsilon_k$$
$$\hat{P}_{k|k} = \hat{P}_{k|k-1} - KH\hat{P}_{k|k-1}$$
$$K = \hat{P}_{k|k-1}H^T S^{-1}$$

Where S is:

$$S = H\hat{P}_{k|k-1}H^T + R_k$$

$p(z_k \mid i)$ can be computed as:

$$p(z_k \mid i) = \frac{1}{(2\pi)^{M/2} \mid \Sigma \mid^{1/2}} \exp\left(-\frac{1}{2}\left((z_k - \hat{z}^i_{k|k-1})' \Sigma_i^{-1} (z_k - \hat{z}^i_{k|k-1})\right)^2 \right) \quad (3.11)$$

3.2.6 Sequential Monte Carlo Methods

In this part brief information about Sequential Monte Carlo Methods are given for completeness.

The Sequential Monte Carlo Methods, or particle filters perform particle representation of probability densities. To describe particle filters first Monte Carlo sampling methods are introduced. Then the idea is applied to recursive Bayesian Estimation [11].

Expectation of a function of a random variable can be written as an integral as given in Equation 3.12.

$$E\{f(x)\} = \int f(x)\, p(x) d(x) \quad (3.12)$$

In this expression $p(x)$ is the probability density function of *x*.

If the random variable $\tilde{x}$, is approximated by *N* samples (i.e. particles) in the sense that $p(x) = \sum_{i=1}^{N} \hat{w}_i \delta(x - x_i)$ then the sample mean of $f(x)$ can be written as [8]:

$$\hat{f} = \sum_{i=1}^{N} f(x^i) \hat{w}(x^i) \quad (3.13)$$

In the equation given above $\hat{f}$ stands for expectation of the function f. In the ideal case, samples are directly generated from $p(x)$ and the expectation is approximated as given in Equation 3.13. Suppose samples are generated from a density $q(x)$,which is similar to $p(x)$. The pdf $q(x)$ is called the importance or the proposal density which satisfies:

$$p(x) > 0 \Rightarrow \quad q(x) > 0 \quad \text{for all } x \in R^{n_x} \tag{3.14}$$

Then Equation (3.13) can be written as:

$$\hat{f} = \int f(x) \frac{p(x)}{q(x)} q(x) d(x) \tag{3.15}$$

Now the Monte Carlo (MC) estimate is:

$$\hat{f} = \sum_{i=1}^{N} f(x^i) \tilde{w}(x^i) \tag{3.16}$$

where $\tilde{w}(x^i)$ is the normalized weight obtained from the unnormalized weight as:

$$w(x^i) = \frac{\left(\frac{p(x^i)}{q(x^i)} \right)}{\sum_{i=1}^{N} \frac{p(x^i)}{q(x^i)}} \tag{3.17}$$

Here $w(x^i)$ are the importance weights. If unnormalized weight $p(x^i)/q(x^i)$ is taken as, $\tilde{w}(x^i)$ Equation (3.16) becomes:

$$\hat{f} = \frac{\sum_{i=1}^{N} f(x^i) \tilde{w}(x^i)}{\sum_{i=1}^{N} \tilde{w}(x^i)} = \sum_{i=1}^{N} f(x^i) w(x^i) \tag{3.18}$$

This approach forms a basis for most sequential MC filters if $p(x)$ is taken as the posterior density in Bayesian estimation. This technique is called sequential importance sampling (SIS).

3.2.6.1 Sequential Importance Sampling

Let $X_k = \{x_j, j = 0, ...k\}$ be a sequences of all target states up to time *k* and x_k be the target state at time *k*. Similarly let $Z_k = \{z_j, j = 0, ...k\}$ be the measurements up to time *k*. $p(X_k \mid Z_k)$ can be approximated using MC integration as follows:

$$p(X_k \mid Z_k) \approx \sum_{i=1}^{N} w_k^i \delta(X_k - X_k^i) \tag{3.19}$$

Here, $p(X_k \mid Z_k)$ is pdf of the complete state history.

Suppose the samples X_k^i are drawn from an importance density $q(X_k \mid Z_k)$, then the weights in Equation 3.19 can be written as:

$$w_k^i \propto \frac{p(X_k^i \mid Z_k)}{q(X_k^i \mid Z_k)} \tag{3.20}$$

In order to obtain a sequential method, the importance function is chosen such that:

$$q(X_k \mid Z_k) \triangleq q(x_k \mid X_{k-1}, Z_k) q(X_{k-1} \mid Z_{k-1}) \tag{3.21}$$

This factorization is the basic idea of sequential importance sampling.

Now, assume that at each iteration a particle set $\{X_k^i, w_k^i\}_{i=1}^N$ for $p(X^k \mid Z^k)$ is available, and a new set $\{X^i_{k+1}, w_{k+1}^i\}_{i=1}^N$ for $p(X_{k+1} \mid Z_{k+1})$ is obtained by using the observation z_{k+1}. Weight update equation can be written as:

$$
\begin{aligned}
p(X_{k+1} \mid Z_{k+1}) &= \frac{p(z_{k+1} \mid X_{k+1}, Z_k)\, p(X_{k+1} \mid Z_k)}{p(z_{k+1} \mid Z_k)} \\
&= \frac{p(z_{k+1} \mid X_{k+1}, Z_k)\, p(x_{k+1} \mid X_k)\, p(X_k \mid Z_k)}{p(z_{k+1} \mid Z_k)} \\
&= \frac{p(z_{k+1} \mid x_{k+1})\, p(x_{k+1} \mid x_k)\, p(X_k \mid Z_k)}{p(z_{k+1} \mid Z_k)}
\end{aligned}
\tag{3.22}
$$

Denominator can be viewed as a normalizing constant, so it can be said that posterior density is proportional to:

$$
p(X_{k+1} \mid Z_{k+1}) \propto p(z_{k+1} \mid x_{k+1})\, p(x_{k+1} \mid x_k)\, p(X_k \mid Z_k) \tag{3.23}
$$

By substituting Equation 3.21 and Equation3.23 into Equation 3.20 the weight update equation can then be shown to be:

$$
\begin{aligned}
w_{k+1}^i &\propto \frac{p(z_{k+1} \mid x_{k+1}^i)\, p(x_{k+1}^i \mid x_k^i)\, p(X_{k+1}^i \mid Y_k)}{q(x_{k+1}^i \mid X_k^i, Z_{k+1})\, q(X_k^i \mid Z_k)} \\
&= w_k^i \frac{p(z_{k+1} \mid x_{k+1}^i)\, p(x_{k+1}^i \mid x_k^i)}{q(x_{k+1}^i \mid X_k^i, Z_{k+1})}
\end{aligned}
\tag{3.24}
$$

Furthermore, if $q(x_{k+1} \mid X_k, Z_{k+1}) = q(x_{k+1} \mid x_k, z_{k+1})$, then the weights take the form:

$$
w_{k+1}^i \propto w_k^i \frac{p(z_{k+1} \mid x_{k+1}^i)\, p(x_{k+1}^i \mid x_k^i)}{q(x_{k+1}^i \mid x_k^i, z_{k+1})} \tag{3.25}
$$

Equation 3.25 is useful in terms of recursive solution. Then the posterior density becomes:

$$
p(x_{k+1} \mid z_{k+1}) \approx \sum_{i=1}^{N} w_{k+1}^i \delta(x_{k+1} - x_{k+1}^i) \tag{3.26}
$$

If $N \to \infty$ in Equation 3.26 $p(x_{k+1} \mid z_{k+1})$ approaches to the true posterior distribution. The SIS algorithm has two main properties [11]. Firstly, nonlinearities

in the state and measurement functions can be handled without any modification (remember Kalman Filter). Secondly, there is no need to parameterize the probability distribution, since any pdf can be approximated by particles. The SIS algorithm is summarized in Table 3-4.

Table 3-4: Filtering Via Sequential Importance Sampling, [11]

$[\{x_{k+1}^i, w_{k+1}^i\}_{i=1}^N] = SIS(\{x_k^i, w_k^i\}_{i=1}^N, z_{k+1})$

- For *i=1:N*

  -Draw $x_{k+1}^i \sim q(x_{k+1}^i \mid x_k^i, z_k)$

  -Update weight $\hat{w}_{k+1}^i = w_k^i \frac{p(z_{k+1} \mid x_{k+1}^i) p(x_{k+1}^i \mid x_k^i)}{q(x_{k+1}^i \mid x_k^i, z_{k+1})}$

- End
- For *i=1:N*

  -Normalize weight $w_{k+1}^i = \frac{\hat{w}_{k+1}^i}{\sum_{i=1}^N \hat{w}_{k+1}^i}$

- End

The SIS algorithm is the basis of most particle filters.

3.2.6.2 Resampling

The SIS algorithm updates particle weights recursively. The most common problem with this recursion is the sample depletion or degeneracy problem. After some iterations all particles but a few may have negligible weights [10]. Processing with these particles makes the algorithm ineffective and causes a degradation in the ability of the method to approximate the posterior density. This problem is named as degeneracy or sample depletion. To overcome this problem, particles with smaller weights can be eliminated, and instead of them higher weighted particles can be generated. This process is called resampling.

The degree of depletion can be represented by effective number of samples [11, 27], which is:

$$N_{eff} = \frac{N}{1 + N^2 Var(w_{k|k}^i)} \tag{3.27}$$

As Equation 3.27 implies N_{eff} becomes maximum ($N_{eff} = N$) when all weights are equal $1/N$. The lowest it can attain is 1, which is the case where only but one particle has the weight $w_{k|k}^i = 1$ and the others have weight 0.

A logical approximation of Equation (3.27) is [27]:

$$\hat{N}_{eff} = \frac{1}{\sum_{i=1}^{N} (w_{k|k}^i)^2} \tag{3.28}$$

In this work Equation 3.28 is used as a measure of finding the effectiveness of particles.

Resampling can be summarized as a mapping of pdf $\{x_k^i, w_k^i\}$ into a similar pdf $\{x_k^{i*}, 1/N\}$. There are a number of resampling algorithms in the literature [27]. The most commonly used resampling algorithm is strafied or systematic resampling which is used in this thesis. Systematic resampling minimizes the variation of $Var(w_k^i)$ and it has less computational complexity than the other resampling algorithms [11].

The systematic resampling assumes that the weights are continues random variables in the interval (0,1) which are randomly ordered. The grid points (u_j's) in each interval $[c_i, c_{i+1})$ are counted, and particles corresponding to that interval are either replicated or eliminated. The algorithm is given in Table 3-5.

After the resampling procedure, all weights are assigned as $1/N$.

Table 3-5: Systematic Resampling, [11]

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

- Construct the CDF
  - $c_1 = 0$
  - For *i=1:N*

    $c_{i+1} = c_i + w_k^i$

  - End For
- Start from the initial point of CDF: $i = 1$
- Draw a starting point : $u_1 \sim U[0, 1/N]$
- For $j = 1:N$
  - Move along the CDF: $u_j = u_1 + (j-1)/N$
  - While $u_j > c_j$

    $i = i + 1$

  - End While
  - Assign sample: $x_k^{j*} = x_k^i$
  - Assign weight : $w_k^{j*} = 1/N$
  - Assign parent : $i^j = i - 1$
- End For

3.2.6.3 Sampling Importance Resampling Particle Filter

Sampling Importance Resampling (SIR) Filter is actually SIS with the resampling step. SIR is illustrated in Figure 3-3 and the algorithm is summarized in Table 3-6.

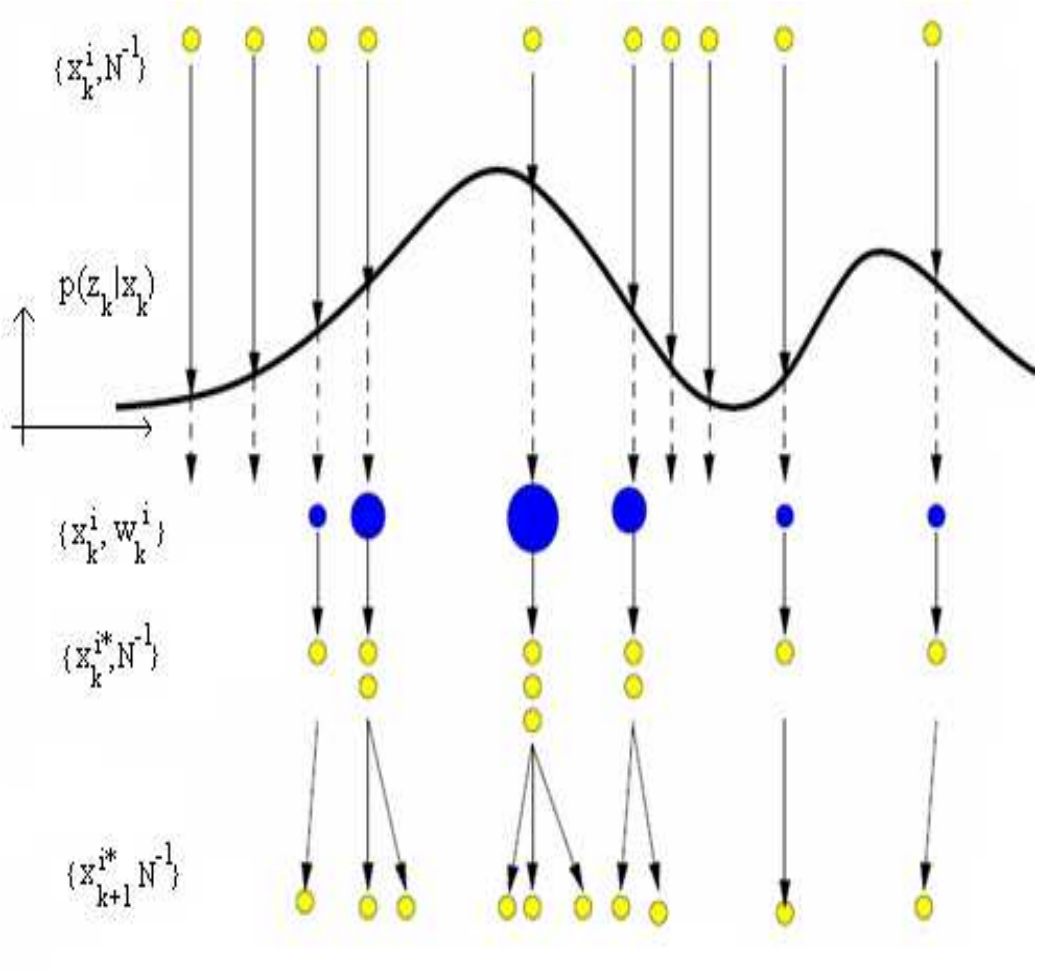

Figure 3-3 A single cycle of a Particle Filter, [11]

Table 3-6: Sampling Importance Resampling Algorithm, [11]

$[\{x_{k+1}^i, w_{k+1}^i\}_{i=1}^N] = SIR(\{x_k^i, w_k^i\}_{i=1}^N, z_{k+1})$

- For *i=1:N*

  -Draw $x_{k+1}^i \sim q(x_{k+1}^i \mid x_k^i, z_k)$

  -Update weight $\hat{w}_{k+1}^i = w_k^i \frac{p(z_{k+1} \mid x_{k+1}^i) p(x_{k+1}^i \mid x_k^i)}{q(x_{k+1}^i \mid x_k^i, z_{k+1})}$

- End For
- For *i=1:N*

  -Normalize weight $w_{k+1}^i = \frac{\hat{w}_{k+1}^i}{\sum_{i=1}^N \hat{w}_{k+1}^i}$

- End For
- Calculate the effective sample size: $\hat{N}_{eff} = \frac{1}{\sum_{i=1}^N (w_{k+1}^i)^2}$
- If $\hat{N}_{eff} < N_{Threshold}$ then Resample

$N_{Threshold}$ is a design parameter. If $N_{Threshold}$ is taken as N ,then resampling is carried out at each step. In this case there is no need to update the weights , since after resampling weights are taken equal.

3.2.6.4 Choice of Importance Density

The selection of importance density is one of the most critical issues in the design of a particle filter. The accuracy of the importance density affects the degeneracy of the particles. There are several importance density proposals in the literature [27]. Here, the functions, which are used in this work are presented.

Optimal Important Density

The optimal importance density function is the posterior function itself [28]:

$$q(x_{k+1} \mid x_k^i, z_{k+1})_{opt} = p(x_{k+1} \mid x_k^i, z_{k+1}) = \frac{p(z_{k+1} \mid x_{k+1}) p(x_{k+1} \mid x_k^i)}{p(z_{k+1} \mid x_k^i)} \tag{3.29}$$

If Equation (3.29) is substituted into the weight update equation (3.25), one obtains:

$$w_{k+1}^i \propto w_k p(z_{k+1} \mid x_{k-1}^i) = w_k^i \int p(z_{k+1} \mid x_{k+1}) p(x_{k+1} \mid x_k) dx_k \tag{3.30}$$

Equation 3.30 states that the importance weights at time *k* can be computed before the particles are propagated to time *k*.

The first drawback of using optimal important density is the unavailability of the density to draw samples. Second drawback is the calculation of the integral in Equation 3.30 [11].

However,the above formulation can be used for some special cases. The first case is when the state x_k is a member of a finite set. In that case it is possible to take samples from the posterior and the integral becomes a sum. In this work this solution is investigated for maneuvering target tracking. The second case is when the posterior density is Gaussian [3].

Prior Density as Importance Density

The standard choice for importance density is to use the conditional prior of the state vector:

$$q(x_{k+1} \mid x_k^i, z_k) = p(x_{k+1} \mid x_k^i) \tag{3.31}$$

Then the weight update equation becomes:

$$w_{k+1}^i = w_k^i p(z_{k+1} \mid x_{+1}^i) \tag{3.32}$$

When the prior density is chosen as the importance density, the current observation is neglected while propagating the state, which in turn may cause quick depletion of the particles.

3.2.7 Regularized Particle Filter

In Section 3.2.6, the basic theory behind the particle filter algorithm is presented. In this part, the Regularized Particle Filter (RPF) that is used in non-maneuvering target tracking problem is described.

Resampling was introduced in particle filter algorithm for reducing degeneracy. However, resampling stage causes another problem, which is called the sample impoverishment [11,28]. At the resampling stage, duplicated particles occupy the same place. If the process noise is small, these duplicated particles are not spread out and after a number of resampling steps all particles may collapse to the same point in the state-space. The reason for this problem is that samples are drawn from the discrete approximation of the posterior distribution.

Lets refresh that the SIR filter posterior distribution was represented as:

$$p(x_{k+1} \mid z_{k+1}) \approx \sum_{i=1}^{N} w_{k+1}^{i} \delta(x_{k+1} - x_{k+1}^{i}) \tag{3.33}$$

In regularized particle filter, this posterior distribution is approximated as:

$$\begin{aligned} p(x_{k+1} \mid z_{k+1}) &\approx w_{k+1}^{i} K_h(x_{k+1} - x_{k+1}^{i}) \\ K_h(x) &= \frac{1}{h^{nx}} K\left(\frac{x}{h}\right) \end{aligned} \tag{3.34}$$

Where $K_h(x)$ is the Kernel density, $h>0$ is the Kernel bandwidth, n_x is the dimension of the state-space.

The Kernel is used for interpolation, with each particle contributing to the estimate in accordance with its distance from x_{k+1}. The kernel and bandwidth are chosen to minimize the Mean Integrated Square Error (MISE) between the posterior PDF

and the corresponding kernel estimate. When the samples are equally weighted, the optimal kernel is the Epanechnikov kernel [11]:

$$K_{opt} = \begin{cases} \frac{n_x + 2}{2c_{n_x}}(1 - \| x \|^2) & \text{if } \|x\| < 1 \\ 0 & \text{otherwise} \end{cases} \quad (3.35)$$

c_{n_x} is the volume of the unit sphere in R^{n_x} .If the posterior density is Gaussian with a unit covariance matrix, the optimal choice for the bandwidth can be given as:

$$h_{opt} = AN^{\frac{-1}{n_x+4}} \text{ with } A = [8c_{n_x}^{-1}(n_x + 4)(2\sqrt{\pi})^{n_x}]^{\frac{1}{n_x+4}} \quad (3.36)$$

Regularized Particle Filter differs from the SIR filter only in resampling stage. During the resampling stage each particle is jittered by adding an extra term as shown in Equation 3.37:

$$x_k^{i*} = x_k^i + h_{opt} D_k \varepsilon_i \quad (3.37)$$

In the above equation, D_k is obtained from the empirical covariance matrix of the particles using Cholesky decomposition at time *k*, i.e. $D_k D_k^T = Cov(x_k^i, w_k^i)$ in Equation 3.37. h_{opt} is the optimal bandwidth and ε_i is a random number drawn from the corresponding Kernel. RPF effectively jitters the resampled values.

In this work Gaussian kernel is used for RPF, where the bandwidth becomes the variance of the Gaussian.

One cycle of RPF is given in Table 3-7.

The disadvantage of RPF is that variance of the posterior distribution increases. In order to overcome this problem Markov Chain Monte Carlo (MCMC) move step based on Metropolis-Hastings algorithm can be used [29]. The basic idea in Metropolis-Hastings algorithm is that a resampled particle x_k^{i*} is moved to a new

state if $u \leq \alpha$ where u is a uniform distributed random variable and α is the acceptance probability. If $u > \alpha$ then the move is rejected.

Suppose particle x_k^{i*} is generated by duplicating x_k^i by resampling step. If the importance density is chosen prior distribution, The Metropolis-Hasting acceptance probability is given by:

$$\alpha = \min\left\{1, \frac{p(z_k \mid x_k^{i*}) p(x_k^{i*} \mid x_k^{i*})}{p(z_k \mid x_k^i) p(x_k^i \mid x_k^i)}\right\} \tag{3.38}$$

This acceptance probability equation can be interpreted as: if at the resampling step generated particle does not belong to higher probability regions in terms of measurement, this particle is not accepted.

Table 3-7 Regularized Particle Filter, [11]

$[\{x_{k+1}^i, w_{k+1}^i\}_{i=1}^N] = RPF(\{x_k^i, w_k^i\}_{i=1}^N, z_{k+1})$

- **For *i=1:N***

  -Draw $x_{k+1}^i \sim p(x_{k+1}^i \mid x_k^i)$

  -Update weight $\hat{w}_{k+1}^i = p(z_{k+1} \mid x_{k+1}^i)$

- **End For**

- **For *i=1:N***

  -Normalize weight $w_{k+1}^i = \dfrac{\hat{w}_{k+1}^i}{\sum_{i=1}^N \hat{w}_{k+1}^i}$

- **End For**

- **Calculate the effective sample size:** $\hat{N}_{eff} = \dfrac{1}{\sum_{i=1}^N (w_{k+1}^i)^2}$

- **If** $\hat{N}_{eff} < N_{Threshold}$

  -Calculate the empirical covariance matrix S_{k+1} **of** $\{x_{k+1}^i, w_{k+1}^i\}$

  -Compute D_{k+1} **such that** $D_{k+1} D_{k+1}^T = S_{k+1}$

  -Resample using strafied sampling

  -For *i=1:N*
  - **Draw** $\varepsilon^i \sim K$ **from the Gaussian Kernel**
  - $x_k^{i*} = x_k^i + h_{opt} D_k \varepsilon_i$

  -End For

- **End If**

3.3 Maneuvering Tracking Filters

In this section, the algorithms will be explained that are used for tracking a target that makes a coordinated turn. Since the turn rate is not known the process model is nonlinear as well as the measurement model. A reasonable approach to overcome the difficulties that rose because of the uncertainty in the turn rate is to use multiple model approach. Below firstly the multiple model filter is explained and then it is applied it to track an object that makes a coordinated turn.

3.3.1 Interacting Multiple Model Filter

The idea behind interacting multiple model (IMM) is to have multiple filters, here denoted as subfilters, running in parallel. IMM is different than multiple model systems since a special interaction between the states of the paralel models is defined.Figure 3-4 shows the block diagram of IMM. For simplicity only two subfilters are illustrated. In IMM five different probability variables are used to calculate which model the system obeys. They are summarized in Table 3-8. If m_k^j denotes the event that the system obeys model m^j at time *k* then the probability that the system changes from model *i* to model *j*, $p^{j|i} = \mathrm{P}_r\{m_k^j \mid m_{k-1}^i\}$, is a priori probability that is considered as a design parameter.

Table 3-8 The Probability variables that are used in the IMM algorithm

μ_k^j	$\mathrm{P}_r\{m_k^j\}$	Probability that the system is in model *j* at time step *k*. Based on measurements Z_k up to time *k*
C_k^j	$\mathrm{P}_r\{m_{k+1}^j\}$	Probability that the system is in model *j* at time step *k+1*. based on measurements Z_k up to time *k*
$p^{j\|i}$	$\mathrm{P}_r\{m_{k+1}^j \mid m_k^i\}$	Apriori probability that the system will make the transition from model *i* to model *j*.
$\mu_k^{i\|j}$	$\mathrm{P}_r\{m_k^i \mid m_{k+1}^j\}$	Conditional probability that the system was in model *i* at time *k* given that it is in model *j* at time *k +1*. Based on measurements Z_k up to time *k*
λ_k^j	$p(z_k \mid Z_{k-1}, r_k = j)$	Conditioned likelihood function of the sytsem at time *k*

In the blocks called *Filter 1* and *Filter 2* each subfilter produces a new estimate of the state vector $x_{k\|k}^j$ according to the model that the subfilter is using. The likelihood of the model λ_k^j is also calculated. λ_k^j is used in the block *Probability Update* where the probability variables are updated. In the block *Merging* all $\hat{x}_{k\|k}^j$ are combined (merged). The estimates from the subfilters $x_{k\|k}^j$ are weighted by their probabilities μ_k^j to obtain the total estimated state vector $\hat{x}_{k\|k}$.

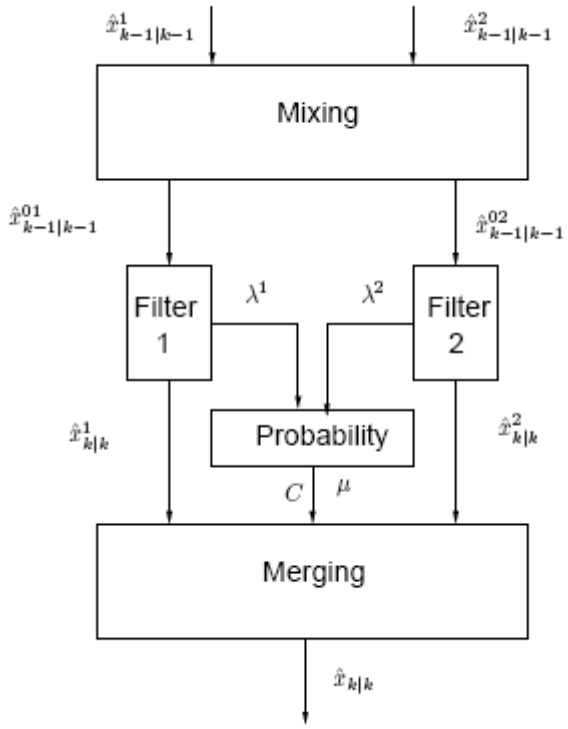

Figure 3-4 Block diagram of IMM Filter.

In the *Mixing* block the model interacts to produce the inputs for *Filter j*, called $\hat{x}^{0j}_{k-1|k-1}$. Assume that the system is in model *j* at time *k*. Since it is unknown in which model the system is at time *k − 1* the estimates of the previous time step are weighted together according to

$$\hat{x}^{0j}_{k-1|k-1} = \sum_i \mu^{i|j}_{k-1} \hat{x}^i_{k-1|k-1} \tag{3.39}$$

where $\mu^{i|j}$ is given by Bayes' theorem as follows:

$$\mu^{i|j}_{k-1} = P_r\{m^i_{k-1} \mid m^j_k\} = \frac{P_r\{m^j_k \mid m^i_{k-1}\} P_r\{m^i_{k-1}\}}{P_r\{m^j_k\}} = \frac{p^{j|i} \mu^i_{k-1}}{C^j_{k-1}} \tag{3.40}$$

The total IMM algorithm is summarized in Table 3-9.

Table 3-9 One Cycle of IMM Filter

1. **Mixing** Calculate the mixed initial state vectors and covariance matrices.

$$\hat{x}^{j0}_{k-1|k-1} = \sum_i \mu^{i|j}_{k-1} \hat{x}^i_{k-1|k-1}$$

$$\hat{P}^{j0}_{k-1|k-1} = \sum_i \mu^{i|j}_{k-1} (\hat{P}^i_{k-1|k-1} + (\hat{x}^i_{k-1|k-1} - \hat{x}^{j0}_{k-1|k-1})(\hat{x}^i_{k-1|k-1} - \hat{x}^{j0}_{k-1|k-1})^T)$$

2. **Filtering** For all subfilters *j* perform the time update and measurement update according to the algorithm of the subfilter. This yields $\hat{x}^j_{k|k}$ and $\hat{P}^j_{k|k}$.

3. **Mode Matching** Determine the likelihood λ^j_k for the observation given model *j*.

$$\lambda^j_k = \frac{1}{(2\pi)^{M/2} |\Sigma_j|^{1/2}} \exp\left(-\frac{1}{2} \left((z_k - \hat{z}^j_{k|k-1})' \Sigma_j^{-1} (z_k - \hat{z}^j_{k|k-1}) \right)^2 \right)$$

Σ and $(z_k - \hat{z}^j_{k|k-1})$ is calculated in the measurement update in the filter and *M* is the dimension of the measurement vector.

4. **Propability Update** Update the probability variables μ^j , $\mu^{i|j}$ and C^j

$$\mu^j_k = \frac{\lambda^j_k C^j_{k-1}}{\overline{C}}$$

$$C^j_k = \sum_i p^{j|i} \mu^i_k$$

$$\mu^{i|j}_k = \frac{\mu^i_k p^{j|i}}{C^j_k}$$

where $\overline{C} = \sum_j \lambda^j_k C^j_{k-1}$.

5. **Estimate and Covariance Combination** Calculate the output of the filter.

$$\hat{x}_{k|k} = \sum_j \mu^j_k \hat{x}^j_{k|k}$$

$$\hat{P}_{k|k} = \sum_j \mu^j_k (\hat{P}^j_{k|k} + (\hat{x}^j_{k|k} - \hat{x}_{k|k})(\hat{x}^j_{k|k} - \hat{x}_{k|k})^T)$$

For our problem, i.e., the maneuvering target tracking three sub filters are used; one for constant velocity (CV) motion, and the other two for the clockwise and counterclockwise coordinated turns. These models are implemented in MSC-EKF. CV motion sub filter is the model that is described in section 3.2.4. State transition matrix for coordinated turn is shown in Equation 3.41 which was described in

Section 2.2.4. The Jacobian corresponding to these transformations are given below.

$$F_k^{(j)} = \begin{bmatrix} 1 & 0 & \dfrac{\sin(\Omega_k^j T)}{\Omega_k^j} & -\dfrac{1-\cos(\Omega_k^j T)}{\Omega_k^j} \\ 0 & 1 & \dfrac{1-\cos(\Omega_k^j T)}{\Omega_k^j} & \dfrac{\sin(\Omega_k^j T)}{\Omega_k^j} \\ 0 & 0 & \cos(\Omega_k^j T) & -\sin(\Omega_k^j T) \\ 0 & 0 & \sin(\Omega_k^j T) & \cos(\Omega_k^j T) \end{bmatrix} \tag{3.41}$$

$$\underset{Fk}{\sim}^{(j)} = \begin{bmatrix} 1 & 0 & \dfrac{\partial f_1^{(j)}}{\partial \dot{x}_k} & \dfrac{\partial f_1^{(j)}}{\partial \dot{y}_k} \\ 0 & 1 & \dfrac{\partial f_2^{(j)}}{\partial \dot{x}_k} & \dfrac{\partial f_2^{(j)}}{\partial \dot{y}_k} \\ 0 & 0 & \dfrac{\partial f_3^{(j)}}{\partial \dot{x}_k} & \dfrac{\partial f_3^{(j)}}{\partial \dot{y}_k} \\ 0 & 0 & \dfrac{\partial f_4^{(j)}}{\partial \dot{x}_k} & \dfrac{\partial f_4^{(j)}}{\partial \dot{y}_k} \end{bmatrix} \quad \text{j= 2,3} \tag{3.42}$$

where

$$\frac{\partial f_1^{(j)}}{\partial \dot{x}_k} = \frac{\sin(\Omega_k^{(j)} T)}{\Omega_k^{(j)}} + g_1^{(j)}(k) \frac{\partial \Omega_k^{(j)}}{\partial \dot{x}_k} \tag{3.43}$$

$$\frac{\partial f_1^{(j)}}{\partial \dot{y}_k} = \frac{-(1-\cos(\Omega_k^{(j)} T))}{\Omega_k^{(j)}} + g_1^{(j)}(k) \frac{\partial \Omega_k^{(j)}}{\partial \dot{y}_k} \tag{3.44}$$

$$\frac{\partial f_2^{(j)}}{\partial \dot{x}_k} = \frac{-(1-\cos(\Omega_k^{(j)} T))}{\Omega_k^{(j)}} + g_2^{(j)}(k) \frac{\partial \Omega_k^{(j)}}{\partial \dot{x}_k} \tag{3.45}$$

$$\frac{\partial f_2^{(j)}}{\partial \dot{y}_k} = \frac{\sin(\Omega_k^{(j)} T)}{\Omega_k^{(j)}} + g_2^{(j)}(k) \frac{\partial \Omega_k^{(j)}}{\partial \dot{y}_k} \tag{3.46}$$

$$\frac{\partial f_3^{(j)}}{\partial \dot{x}_k} = \cos(\Omega_k^{(j)} T) + g_3^{(j)}(k) \frac{\partial \Omega_k^{(j)}}{\partial \dot{x}_k} \tag{3.47}$$

$$\frac{\partial f_3^{(j)}}{\partial \dot{y}_k} = -\sin(\Omega_k^{(j)} T) + g_3^{(j)}(k) \frac{\partial \Omega_k^{(j)}}{\partial \dot{y}_k} \tag{3.48}$$

$$\frac{\partial f_4^{(j)}}{\partial \dot{x}_k} = \sin(\Omega_k^{(j)} T) + g_4^{(j)}(k) \frac{\partial \Omega_k^{(j)}}{\partial \dot{x}_k} \tag{3.49}$$

$$\frac{\partial f_4^{(j)}}{\partial \dot{y}_k} = \cos(\Omega_k^{(j)} T) + g_4^{(j)}(k) \frac{\partial \Omega_k^{(j)}}{\partial \dot{y}_k} \tag{3.50}$$

with

$$g_1^{(j)}(k) = \frac{T\cos(\Omega_k^{(j)} T)\dot{x}_k^t}{\Omega_k^{(j)}} - \frac{\sin(\Omega_k^{(j)} T)\dot{x}_k^t}{(\Omega_k^{(j)})^2} - \frac{T\sin(\Omega_k^{(j)} T)\dot{y}_k^t}{\Omega_k^{(j)}} + \frac{(-1+\cos(\Omega_k^{(j)} T))\dot{y}_k^t}{(\Omega_k^{(j)})^2} \tag{3.51}$$

$$g_2^{(j)}(k) = \frac{T\sin(\Omega_k^{(j)} T)\dot{x}_k^t}{\Omega_k^{(j)}} - \frac{(1-\cos(\Omega_k^{(j)} T))\dot{x}_k^t}{(\Omega_k^{(j)})^2} + \frac{T\cos(\Omega_k^{(j)} T)\dot{y}_k^t}{\Omega_k^{(j)}} - \frac{\sin(\Omega_k^{(j)} T)\dot{y}_k{}^t}{(\Omega_k^{(j)})^2} \tag{3.52}$$

$$g_3^{(j)}(k) = -\sin(\Omega_k^{(j)} T)T\dot{x}_k{}^t - \cos(\Omega_k^{(j)} T)T\dot{y}_k{}^t \tag{3.53}$$

$$g_4^{(j)}(k) = \cos(\Omega_k^{(j)} T)T\dot{x}_k{}^t - \sin(\Omega_k^{(j)} T)T\dot{y}_k{}^t \tag{3.54}$$

$$\frac{\partial \Omega_k^{(j)}}{\partial \dot{x}_k} = \frac{(-1)^{j+1} a_m \dot{x}_k{}^t}{\left[(\dot{x}_k{}^t)^2 + (\dot{y}_k{}^t)^2\right]^{3/2}} \tag{3.55}$$

3.3.2 Multiple-Model Particle Filter

Like in Kalman-Filter based methods, multiple state-transition models can be utilized in particle filter algorithms to represent dynamic system more precisely [30, 31].

For multiple-model particle filter (MM-PF) state is extended with the regime variable. Therefore, the state vector consists of both a continuous valued part and a discrete-valued part. The augmented state vector is expressed by $y_k = [x_k^T \ \mathrm{r}_k]^T$ where x_k stands for target states (continuous part), and r_k stands for regime variable (discrete part) where $r_k \in S = \{1, 2, ..., s\}$.

In the MM-PF firstly, $\{r_{k+1}^n\}_{n=1}^N$ is generated based on $\{r_k^n\}_{n=1}^N$ and the Markov transitional probability matrix $\Pi = [\pi_{ij}]$, where $i, j \in S$ .This can be done by defining a new matrix Π^R such that $\Pi^R = \Pi D$. *D* is the upper triangular matrix with diagonal and the elements above the diagonal being equal to *1*. An example for *s=2* is shown below:

$$\Pi = \begin{bmatrix} 0.6 & 0.4 \\ 0.4 & 0.6 \end{bmatrix}$$
$$\Pi^R = \begin{bmatrix} 0.6 & 1 \\ 0.4 & 1 \end{bmatrix} \tag{3.56}$$

Suppose the particle *i* is at regime 1 at time *k.* At time *k+1* this particle will be at regime 1 with probability π_{11} and it goes to state 2 with probability π_{12}. To realize this transition we draw a sample from the uniform distribution $u_n \sim U(0,1)$. If u_n is in between 0 and π_{11} the new regime is assigned as 1. If it is between π_{11} and 1 (which is the $\Pi^R(1,2)$) it is assigned as 2. This procedure can be extended to *s* number of regimes straightforwardly.

At the next step, sampling importance-resampling algorithm is used based on regime variable of the particle. Table 3-10 summarizes the MM-PF algorithm. For our problem the components of continuous valued vector are target kinematic variables. The discrete valued part is turn model. The discrete valued part can take values 1,2,3 where 1 stands for constant velocity, 2 and 3 stands for clockwise and counter clockwise coordinated turn models, respectively.

3.3.3 Marginalized Particle Filter

The basic idea behind the Marginalized (Rao-Blackwellized) Particle Filter (MPF) is partitioning the state vector into two parts as linear and non-linear. By this way, state dimension for the particle filter solution is kept small, which in turn increases the accuracy of the particle filter solutions. Let the state vector be $x_k = (x_k^l, x_k^n)^T$ where x_k^l corresponds to the states that enters both the state and observation model linearly, x_k^n corresponds to the nonlinear states that has non-linear propagation and measurement equations. Essentially particle filter solution is applied to non-linear states, x_k^n . The linear states are propagated through Kalman Filter for each particle. More information about MPF can be found in [27]. In this work since none of the states has a linear part, a suboptimal solution namely the Extended Kalman Filter (EKF) is used for target kinematic variables and particle filter is used for regime transition variable. One cycle for MPF is given in Table 3-11.

For regime transition, r_k optimal importance density can be used for particle filter [9].

Table 3-10 Multiple Model Particle Filter, [11]

$[\{y_{k+1}^i, w_{k+1}^i\}_{i=1}^N] = MMPF(\{y_k^i, w_k^i\}_{i=1}^N, z_{k+1})$

- $[x_k{}^T \quad \mathrm{r}_k] = y_k$
- For *n=1:N*
  -Draw u_n from a uniform distrubition
  -Set $i = r_k^n$
  -*m=1*
  -While (Π^R(i,m) < u_n)
    - m=m+1

  -End While
  -set $r_{k+1}^n = m$
- End For
- Regime conditioned sampling importance resampling
  $[\{x_{k+1}^n, w_{k+1}^n\}] = RC - SIR[\{x_k^n, w_k, z_{k+1}\}]$
- Calculate the effective sample size: $\hat{N}_{eff} = \frac{1}{\sum_{i=1}^{N}(w_{k+1}^i)^2}$
- If $\hat{N}_{eff} < N_{Threshold}$

  -Resample
- End If
- $y_{k+1} = [x_{k+1}{}^T \quad \mathrm{r}_{k+1}]$

Table 3-11 Marginalized Particle Filter, [11]

$$[\{\hat{x}^i_{k+1|k+1}, P^i_{k+1|k+1}, r^i_{k+1}\}^N_{i=1}] = \text{Marginalized } PF(\{\hat{x}^i_{k|k}, P^i_{k|k}, r^i_k\}^N_{i=1}, z_{k+1})$$

- For *i=1:N*
  -Draw r^i_{k+1} from the optimal importance density
  -Compute the unnormalized weights $\tilde{w}^i_{k+1}$
- End For
- Compute the normalize weights w^i_{k+1}
- For i=*1:N*
  -Apply one-step-ahead EKF
  - $[\hat{x}^i_{k+1|k+1}, P^i_{k+1|k+1}] = EKF[\hat{x}^i_{k|k}, P^i_{k|k}, r^i_k, z_{k+1}]$
- End For
- Calculate the effective sample size: $\hat{N}_{eff} = \dfrac{1}{\sum_{i=1}^{N} (w^i_{k+1})^2}$
- If $\hat{N}_{eff} < N_{Threshold}$

  -Resample
- End If

3.3.4 Auxiliary Multiple Model Particle Filter

In multiple-model estimation remember that aim was to estimate, the posterior distribution of the state and the regime variable $p(x_{k+1}, r_{k+1} \mid Z_{k+1})$ [15]. For Sequential Monte Carlo Methods this density is presented by particles, so the posterior density for i^{th} particle at time $k+1$ can be written as [11]:

$$\begin{aligned} p(x_{k+1}, i, r_{k+1} \mid Z_{k+1}) &\propto p(z_{k+1} \mid x_{k+1}) p(x_{k+1}, i, r_{k+1} \mid Z_k) \\ &= p(z_{k+1} \mid x_{k+1}) p(x_{k+1} \mid i, r_{k+1}, Z_k) p(r_{k+1} \mid i, Z_k) p(i \mid Z_k) \\ &= p(z_{k+1} \mid x_{k+1}) p(x_{k+1} \mid x_k^i, r_{k+1}) p(r_{k+1} \mid r_k^i) w_k^i \end{aligned} \quad (3.57)$$

In Equation 3.57 $p(r_{k+1} \mid r_k^i)$ is an element transitional probability matrix Π . Since it is hard to sample from $p(x_{k+1}, i, r_{k+1} \mid Z_k)$ define a new importance sampling function $q(x_{k+1}, i, r_{k+1} \mid Z_k)$:

$$q(x_{k+1}, i, r_{k+1} \mid Z_{k+1}) \propto p(z_{k+1} \mid \mu_{k+1}^i(r_{k+1})) p(x_{k+1} \mid x_k^i, r_{k+1}) p(r_{k+1} \mid r_k^i) w_k^i \quad (3.58)$$

Here μ_{k+1}^i's are defined as support points and can be defined as:

$$\begin{aligned} \mu_{k+1}^i(r_{k+1}) &= E\{x_{k+1} \mid x_k^i, r_{k+1}\} \\ &= \mathrm{f}(\mathrm{x}_k^i, \mathrm{x}_{k-1}^o, \mathrm{r}_{k+1}) \end{aligned} \quad (3.59)$$

In this work importance sampling function $q(x_{k+1} \mid i, r_{k+1} \mid Z_{k+1})$ is selected as prior distribution:

$$q(x_{k+1}, i, r_{k+1} \mid Z_{k+1}) \triangleq p(x_{k+1} \mid x_k, r_{k+1}) \quad (3.60)$$

If Equation (3.58) is integrated with respect to x_{k+1} to get an expression for $q(i, r_{k+1} \mid Z_{k+1})$ Equation (3.61) is obtained.

$$q(i, r_{k+1} \mid Z_{k+1}) \propto p(z_{k+1} \mid \mu_{k+1}^i(r_{k+1})) p(r_{k+1} \mid r_k^i) w_k^i \quad (3.61)$$

In AUX-MMPF, firstly a random sample is selected from the distribution $q(x_{k+1}, i, r_{k+1} \mid Z_{k+1})$. This can be achieved by first sampling the regime variables for corresponding particle i.e, $\{i^j, r_k^j\}$ [11]. To do this, all N particles firstly are propagated for all regimes $R = \{1, 2...., s\}$. The number of particles is now increased to sN. Each of the sN particles is assigned a weight using Equation (3.64) and N particles with $\{i^j, r_k^j\}_{j=1}^N$ are sampled using obtained weights. The resultant triplet sample $\{x_k^j, i^j, r_k^j\}_{i=1}^N$ is a random sample from density $q(x_{k+1}, i, r_{k+1} \mid Z_{k+1})$. To use these samples to characterize $p(x_{k+1}, i, r_{k+1} \mid Z_{k+1})$, weights w_k^j are attached to each particle ,where w_k^j is ratio of (3.61) and (3.60). Notice that in Equation 3.62 importance sampling function is replaced by prior distribution.

$$\begin{aligned} w_k^j &= \frac{p(z_{k+1} \mid x_{k+1}^j)\, p(x_{k+1}^j \mid x_k^{i^j}, r_{k+1}^j)\, p(r_{k+1}^j \mid r_k^{i^j})\, w_{k-1}^{i^j}}{p(z_{k+1} \mid \mu_k^{ij}(r_k))\, p(x_{k+1}^j \mid x_k^{i^j}, r_{k+1}^j)\, p(r_{k+1}^j \mid r_k^{i^j})\, w_{k-1}^{i^j}} \\ &= \frac{p(z_{k+1} \mid x_{k+1}^j)}{p(z_{k+1} \mid \mu_k^{ij}(r_k))} \end{aligned} \tag{3.62}$$

By defining the augmented state vector as $y_{k+1} \triangleq [x_{k+1}^T, i, r_k]$, the distribution $p(x_{k+1}, i, r_{k+1} \mid Z_{k+1})$ can now be written as:

$$p(x_{k+1}, i, r_{k+1} \mid Z_{k+1}) = p(y_k) \approx \sum_{j=1}^{M} w_k^i \delta(y_k - y_k^j) \tag{3.63}$$

A single cycle of the AUX-MMPF is described in Table 3-12.

Table 3-12 Auxiliary Multiple Model Particle Filter,[11]

$[\{y_{k+1}^i, w_{k+1}^j\}_{i=1}^N] = AUX - MMPF(\{y_k^i, w_k^i\}_{i=1}^N, z_{k+1})$

- $[x_k^T \quad \mathrm{r}_k] = y_k$
- For *n=1:N*
  -For r_k =1:s
    - Compute support points $\mu_k^i(r_{k+1}) = f(x_k^i, x_{k-1}^o, x_k^o, r_{k+1})$
    - Compute weights $q(i, r_{k+1} \mid Z_{k+1}) \propto p(z_{k+1} \mid \mu_{k+1}^i(r_{k+1}))p(r_{k+1} \mid r_k^i)w_k^i$

  -End For
- End For
- Draw N samples $\{i^j, r_k^j\}_{j=1}^N$ from the Ns samples according to weights created in Step 1.
- Predict the selected N samples
  -For *j=1:N*
    - $x_k^j = f(x_k^i, x_{k-1}^o, x_k^o, r_{k+1}) + G v_k^j$
    - $w_k^j \propto \dfrac{p(z_k \mid x_k^j)}{p(z_k \mid \mu_k^{i^j}(r_k^j))}$

  - End For
- Normalize the weights
- Calculate the effective sample size: $\hat{N}_{eff} = \dfrac{1}{\sum_{i=1}^{N}(w_{k+1}^j)^2}$
- If $\hat{N}_{eff} < N_{Threshold}$

  -Resample
- End If
- $y_{k+1} = [x_{k+1}^T \quad \mathrm{r}_{k+1}]$

CHAPTER 4

SIMULATIONS and DISCUSSION

In this chapter, the algorithms and system models outlined in the previous chapters are applied to representative target trajectories and results are discussed.

4.1 Performance Evaluation

A tracking filter's performance depends on a variety of things, such as the trajectories of the target and the platform, measurement of noise and initialization of the filter and the process. How accurate the estimate will be in bearings-only target tracking depends, to a large extent, on the geometry of the scenario in question. Many simulations are needed to test the filter for all possible trajectories and filter initializations. Even if all scenarios could be tested, the result set will be too large to interpret . We have the risk that the filter shows too good performance in simulations while it may not be the case in practice because the scenarios diverts too much from the reality. There is also another risk which is the opposite of the above mentioned, i.e., the risk that the filter showing a bad performance in simulations is rejected; however, simulation scenarios may have low probability to occur in reality. In the case that the common scenarios are known, they can be directly used in the simulations. Another alternative can be testing the filter for scenarios in which performance of the filter is the most critical. Therefore, if the filter shows a good performance in critical situations then probably a performance which is worse can be accepted for the other scenarios. Furthermore, filters which show bad performance in critical situations can also be rejected without conducting some other tests.

Monte Carlo simulation is a method that is widely used to test the performance of the filters. In this method, all the filter inputs are considered to be variables with known distributions. Inputs that are obtained from these distributions can be used in many simulations. If the number of simulations is big then the filter performance is considered to reflect the true behavior of the filter. Monte Carlo Stimulations can be used to test the filter for several scenarios and the mean estimation error can be investigated in a many simulations, which helps to measure how well the filter performs in general.

In our simulations, process and measurement noises are selected as Gaussian. In order to investigate the influence of a parameter, it is varied in a deterministic manner while the others are held constant. To evaluate the performance a set of simulations are performed and the mean value of estimation error is investigated. The details about the selection of the parameters or noises are described in detail in the Simulations section. The tested parameters include the target initial position and velocity.

The main parameter used for performance comparison in this study is the Root-Mean-Square Error (RMSE) of the range. It can be computed in 2D (x, y) or in 3D (x,y,z). RMSE is calculated at each time step as given in Equation 4.1 to be able to observe the maneuver dependent characteristics of the algorithms. If there are (N_{mc}) number of Monte Carlo runs for a specified scenario RMSE is averaged.

$$RMSE_k = \sqrt{\frac{1}{N_{MC}} \sum_{i=1}^{N_{mc}} \left(\left(x_{1k} - x_{1k}^{\ true} \right)^2 + \left(x_{2k} - x_{2k}^{\ true} \right)^2 + \left(x_{3k} - x_{3k}^{\ true} \right)^2 \right)} \quad (4.1)$$

As an additional parameter, the time mean of root means square error RMSE is also calculated for each simulation. Here L is the number of steps in the simulation.

$$RMSE = \frac{1}{L} \sum_{k=1}^{L} \sqrt{\frac{1}{N_{MC}} \sum_{i=1}^{N_{mc}} \left(\left(x_{1k} - x_{1k}^{\ true} \right)^2 + \left(x_{2k} - x_{2k}^{\ true} \right)^2 + \left(x_{3k} - x_{3k}^{\ true} \right)^2 \right)} \quad (4.2)$$

Another parameter evaluated is range root mean square after observer maneuver. Since the system may be unobservable before the maneuver, this parameter is evaluated for the performance of the filters when the system becomes observable. Also, the final range error is calculated which is the error after obtaining the last measurement.

Lastly, confidence ellipses are shown for some simulation results. The uncertainty of an estimate can be expressed in terms of confidence ellipse [34]. For Kalman Filter solutions confidence ellipses can be calculated using covariance matrix values, and for particle filter solutions it can be evaluated using the particles at one time instant. The ellipses are drawn using the function in [37].

4.2 Filter Initialization

Initialization of filters is an important issue in estimation theory. Since the problem with bearings only tracking is unobservable for some sceanarios, this problem becomes crucial. In the literature initialization problem for bearings-only tracking is rarely mentioned.

4.2.1 Cartesian Coordinates Kalman Filter Initialization

Suppose that the initial range error prior is normally distributed ($r \sim N(\bar{r}, \sigma_r^2)$. The bearing measurement error is zero mean Gaussian, which can be shown as $\varphi \sim N(\bar{\varphi}, \sigma_\varphi^2)$ where $\bar{\varphi}$ is the first bearing measurement [9]. Let x_1 and x_2 components of the Cartesian state vector be written in terms of r and φ as follows:

$$x_1 = r\cos\varphi \qquad \mathrm{x}_2 = r\sin\varphi \tag{4. 3}$$

From the distributions of $(r-\bar{r})$ and $(\varphi-\bar{\varphi})$ and the transformations above, it can be shown that an approximate mean and covariance of (x_1 ,x_2) are:

$$P_{x_1x_2} = E[(x_1 - \overline{x}_1)(x_2 - \overline{x}_2)^T] = \begin{bmatrix} P_{x_1x_1} & P_{x_1x_2} \\ P_{x_2x_1} & P_{x_2x_2} \end{bmatrix} \quad (4.4)$$

where

$$\begin{aligned} P_{x_1x_1} &= \overline{r}\sigma_\varphi^2 \sin^2 \overline{\varphi} + \sigma_r^2 \cos^2 \overline{\varphi} \\ P_{x_2x_2} &= \overline{r}\sigma_\varphi^2 \cos^2 \overline{\varphi} + \sigma_r^2 \sin^2 \overline{\varphi} \\ P_{x_1x_2} &= (\sigma_r^2 - \overline{r}^2\sigma_\varphi^2) \sin \overline{\varphi} \cos \overline{\varphi} \\ P_{x_2x_1} &= P_{x_1x_2} \end{aligned} \quad (4.5)$$

Similarly, suppose that the target speed *s* and the course *c* is given by $s \sim N(\overline{s}, \sigma_r^2)$ and $c \sim N(\overline{c}, \sigma_r^2)$. The covariance values of velocity components $\overline{\dot{x}}_1$ and $\overline{\dot{x}}_2$ can be calculated as:

$$\begin{aligned} P_{\dot{x}_1\dot{x}_1} &= \overline{s}^2\sigma_c^2 \sin^2 \overline{c} + \sigma_s^2 \cos^2 \overline{c} \\ P_{\dot{x}_2\dot{x}_2} &= \overline{s}^2\sigma_c^2 \cos^2 \overline{c} + \sigma_s^2 \sin^2 \overline{c} \\ P_{\dot{x}_1\dot{x}_2} &= (\sigma_s^2 - \overline{s}^2\sigma_c^2) \sin \overline{c} \cos \overline{c} \\ P_{\dot{x}_2\dot{x}_1} &= P_{\dot{x}_1\dot{x}_2} \end{aligned} \quad (4.6)$$

Then the initial covariance matrix for the 2-D case can be defined as:

$$P_{x0} = \begin{bmatrix} P_{x_1x_1} & P_{x_1x_2} & 0 & 0 \\ P_{x_2x_1} & P_{x_2x_2} & 0 & 0 \\ 0 & 0 & P_{\dot{x}_1\dot{x}_1} & P_{\dot{x}_1\dot{x}_2} \\ 0 & 0 & P_{\dot{x}_2\dot{x}_1} & P_{\dot{x}_2\dot{x}_2} \end{bmatrix} \quad (4.7)$$

4.2.2 Modified Spherical Coordinates Kalman Filter Initialization

In the literature Modified Spherical Coordinates (MSC) Kalman Filter covariance initialization is done by using heuristic values, and taking covariance matrix as diagonal [3,4,8,32,].

One of the proposed solutions is transforming the covariance found in Cartesian coordinates into MSC using Jacobian matrix [11].

$$\begin{aligned} y &= g(x) \\ \overline{y} &\approx g(E(x)) = g(\overline{x}_1) \\ P_{y0} &= E[(y - \overline{y})(y - \overline{y})^T] = GP_{x0}G^T \end{aligned} \tag{4. 8}$$

where *G* is the Jacobian matrix of the transformation function from Cartesian to MSC.. However, since the function g is highly nonlinear, this approximation is not satisfactory for some cases. Therefore, a different initialization procedure can be defined using Monte Carlo Simulation Techniques as:

$$\begin{aligned} P_{y0} &= E[(y-\overline{y})(y-\overline{y})^T] = E(g(x)g(x)^T) - E(g(x))E(g(x))^T \\ P_{y0} &= \frac{1}{pn^2}\sum_{i=1}^{pn} g(x_i)g(x_i)^T - \left(\frac{1}{pn}\sum_{i=1}^{pn} g(x_i)\right)\left(\frac{1}{pn}\sum_{i=1}^{pn} g(x_i)\right)^T \end{aligned} \tag{4. 9}$$

where pn represents the number of particles, x_i is a sample drawn from the Cartesian coordinates initial estimate and covariance value. Here the samples are assumed to have multivariate normal distribution. In our simulations, this simple trick improves the accuracy of the algorithm.

4.2.3 Range Parameterized Kalman Filter Initialization

Range-Parameterized Extended Kalman Filter (RP-EKF) divides the range interval into 6 subintervals ($r_{\min} = \hat{r}_o - 2\sigma_r$, $r_{\max} = \hat{r}_o + 2\sigma_r$). The speed interval is also divided into 6 intervals ($s_{\min} = \hat{s}_o - 2\sigma_s, s_{\min} = \hat{s}_0 + 2\sigma_s$). Each filter is associated within a range estimate and speed estimate which gives *6x6=36* filters in total. Covariance matrix is initialized for each filter by using the algorithms described for MSC.

4.2.4 Particle Filter Initialization

For a particle filter application, samples are drawn from the multivariate Gaussian distribution using the covariance evaluated for MSC or Cartesian coordinates.

Also, accept-reject algorithm is used in order to not to have negative range or negative velocity magnitude.

4.3 Non-Maneuvering Target Simulations

For non-maneuvering target problem Cartesian coordinates-Extended Kalman Filter (CAR-EKF), modified spherical coordinates-Extended Kalman Filter (MSC-EKF), regularized particle filter (RPF) algorithms are implemented. The theory behind these algorithms are described in Section 3. Regularized particle filter is selected, since the process noise is small compared to the state values. The particle diversity is achieved by the regularization step. Resampling is carried when $N_{eff} < 9xN/10$. Here *N* denotes the total number of particles. At the regularization step a Gaussian kernel is used. h_{opt} is calculated using Equation 3.36, where the c_{n_x} is equal to $\pi^2/2$ (n_x=4).

4.3.1 Scenario 1

In the first scenario, 2D non-maneuvering target case is investigated. Target and observer trajectories are shown in Figure 4-1. Also, using Scenario 1 the proposed MSC filter covariance initialization procedure is investigated.

Observer has an initial course of 140° and has a speed of 155 m/s. It turns between 13-17 seconds at constant speed and has a new course of +20°. Target moves on a straight line with course of -160° at a initial speed of 125 m/s. Target velocity is propagated using Singer acceleration model. Total time of the simulation is 30 seconds.

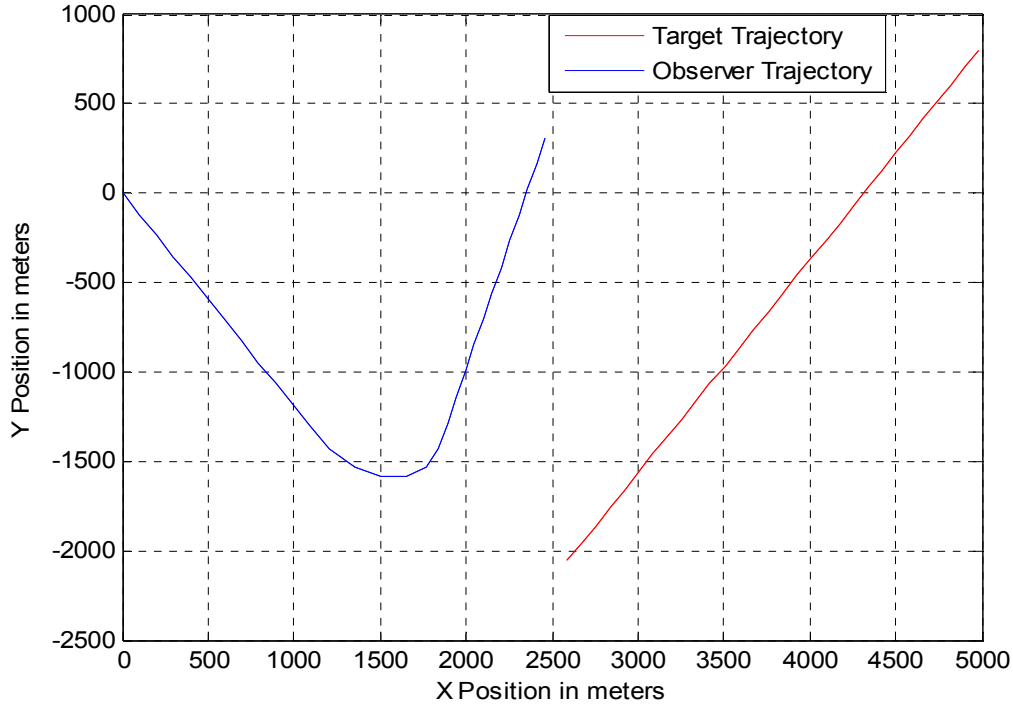

Figure 4-1 Target and Observer Trajectories for Scenario 1

In the first scenario target moves with nearly CV on a straight line, with the noise added on the magnitude of the velocity. An example of velocity magnitude is shown in Figure 4-2.

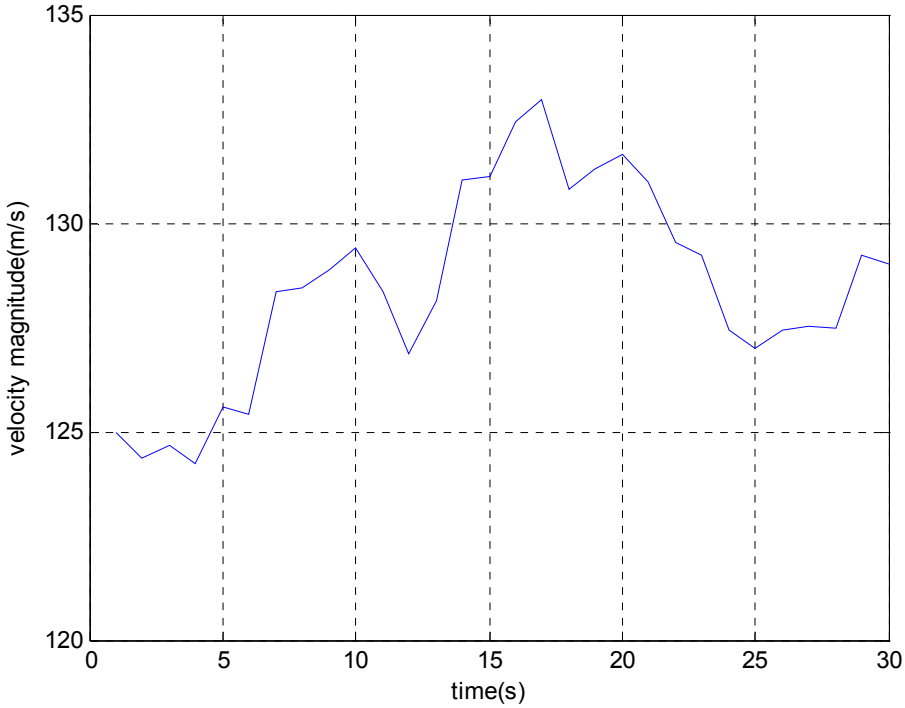

Figure 4-2 Velocity Magnitude of Target

The initial error and simulation parameters can be found in Table 4-1.

Table 4-1 Scenario 1 Simulation Parameters

Measurement Noise	Normally Distributed with μ = 0 and σ = 1.5°
Initial Range Error	Normally Distributed with μ = 0 and σ = 1.5e3 m
Initial Speed Error	Normally Distributed with μ = 0 and σ = 60 m/s
Initial Course Error	Normally Distributed with μ = 0 and σ = 30°
Process Noise	Singer Model with $w_k(0)$ = 1 m/s and β =0.5m/s
Particle Number For SIR	5000

100 MC runs are carried out for Scenario 1. In the first result set, the MSC covariance is initialized using linearization technique. The range error is shown in Figure 4-3 and a detailed comparison is given in Table 4-2.

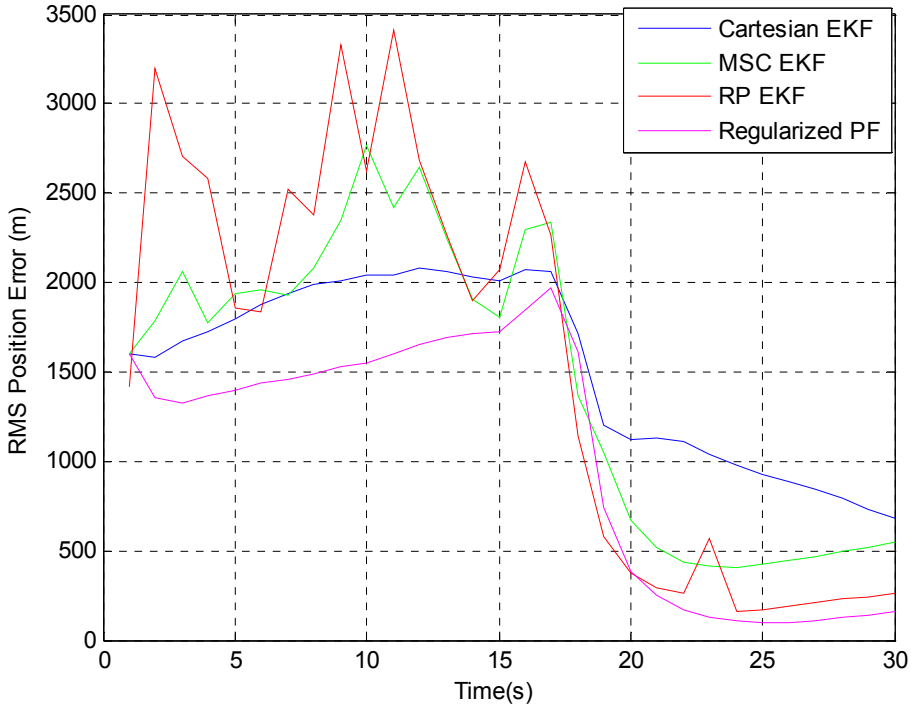

Figure 4-3: Scenario 1 Simulation Results for Initialization Procedure 1

The comparison criteria used in Table 4-2 is described in chapter 4. Here RTAMS stands for root time average mean square error, which is the all time averaged error for 100 MC runs. RTAMS after the maneuver *(t=17s)* is the root time average mean square error after the observer platform maneuver. And the final error is the final range error value that it is the error at final time (*t*=30 s).

Table 4-2 Scenario 1 Performance Comparison for Initialization Procedure 1

Algorithm	RTAMS(m)	RTAMS(m) After the Manueveur	Final Error(m)
Cartesian EKF	1525	1086	676
MSC EKF	1455	721	549
RP EKF	1546	494	262
Regularized PF	1027	435	159

The same simulation is carried out for MSC initialization based on MC estimation. The range error for MSC and RP-EKF is shown in Figure 4-4 and a detailed comparison is given in Table 4-3.

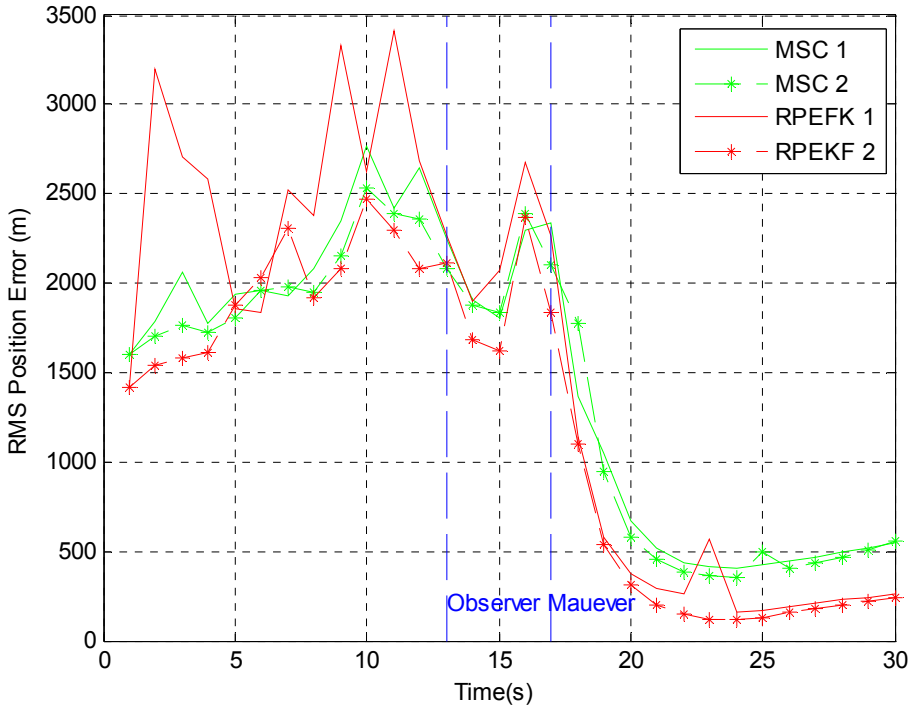

Figure 4-4: Simulation Results for Initialization Procedure 1 and 2 for MSC

Table 4-3 Performance Comparison for Different Initialization Procedures

Algorithm	RTAMS(m)	RTAMS(m) After the Maneuver	Final Error(m)
MSC EKF with initial Covariance 1	1455	721	549
MSC EKF with initial Covariance 2	1398	703	548
RP EKF with initial Covariance 1	1546	494	262
RP EPF with initial covariance 2	1216	393	240

Table 4-3, it can be seen the range estimate accuracy is increased for Monte Carlo initialization solution. Actually, range root time averaged mean square error

(RTAMS) for all observation period has an improvement of 5% in MSC EKF solution and 30% in RP-EKF solution.

In Figure 4-4, MSC and RP-EKF range error have some jumps before the maneuver of the observer. This is because, in some of the MC runs filter become unstable for a period of time, which is named here as partially divergent track. Partially divergent track means that the filter become unstable for a period of time, but eventually it converges to the true solution. This situation is illustrated in Figure 4-5. The solution is unstable in 11-15 seconds interval, and after this interval, it converges to the true estimate. All the states except the range state are observable, therefore they are not shown in the figure.

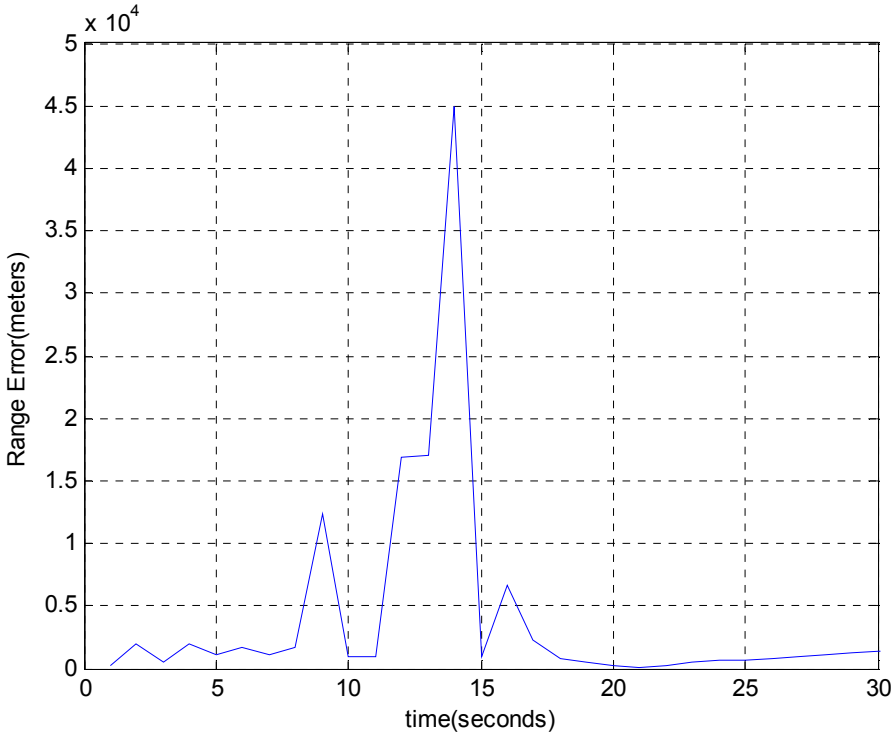

Figure 4-5 An Example of Partially Divergent Track Range Error

Partially divergent tracks are result of the covariance matrix initialization errors. This problem is addressed in [3]. Here, we proposed a different initialization

procedure to solve this problem. Even though, the same problem is encountered, error values are improved substantially.

RP-EKF improvement is much more than MSC-EKF. The reason is there are 36 filters running in parallel in RP-EKF, and the probability of producing partially divergent tracks is larger (the weighting of the divergent filters become zero after some time).

Error ellipses at time 10, 20 and 30 seconds are shown in Figure 4-7, Figure 4-8, Figure 4-9 and Figure 4-10. Here the MSC EKF and RP-EKF are initialized using Monte Carlo Estimation technique. RP-EKF and Regularized PF confidence (error) ellipse encapsulate the true position while MSC EKF and Cartesian EKF do not.

For Scenario 1 regularized particle filter (RPF) is implemented in modified spherical coordinates (MSC), too. Remember from Section 3 that a regularization step is included in RPF filter in order to prevent sample depletion. In MSC for 2 dimensional problem (2D), states can be written as: $y^{MSC} = [1/r \ \varphi \ \dot{\varphi} \ \dot{r}/r]^T$ .Since the last three states are observable, regularization step is only implemented for first state, which is the state that gives range information. By this way, increase in the variance of the estimates is kept bounded. The range error for RPF MSC filter and RPF Cartesian coordinates filter is shown in Figure 4-6.

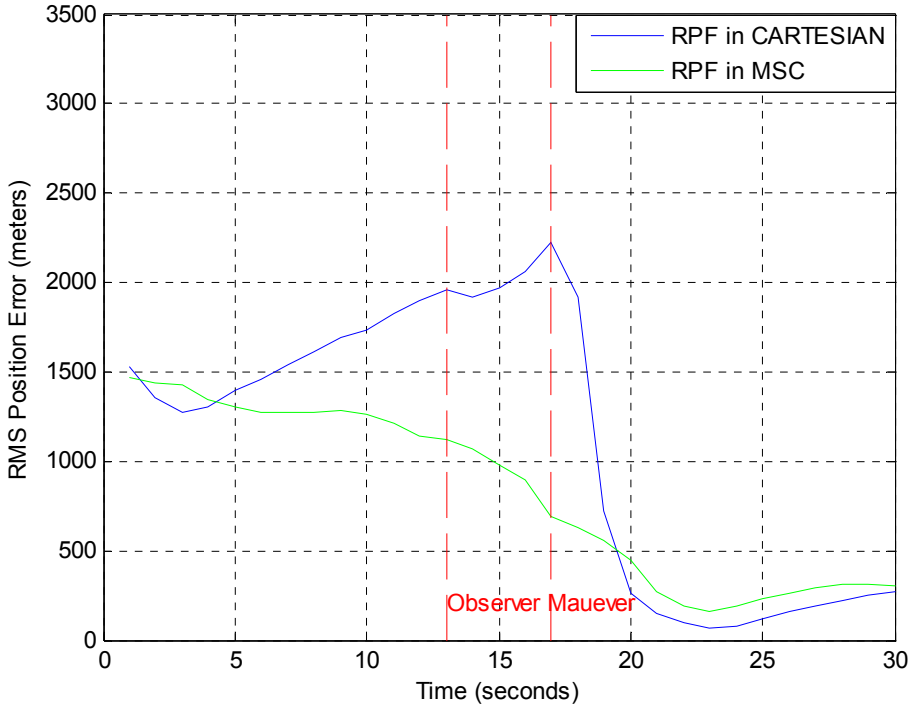

Figure 4-6 Comparison of Particle Filter Solutions with Different Coordinate Systems

As shown in Figure 4-6 overall performance RPF filter in MSC is better than RPF filter in Cartesian Coordinates. We would like to emphasize that regularization is applied partially. As shown in Figure 4-10 and Figure 4-11 the variance of the MSC regularized PF solution variance is small compared to Cartesian one. It remains as a future work to be studied in detail.

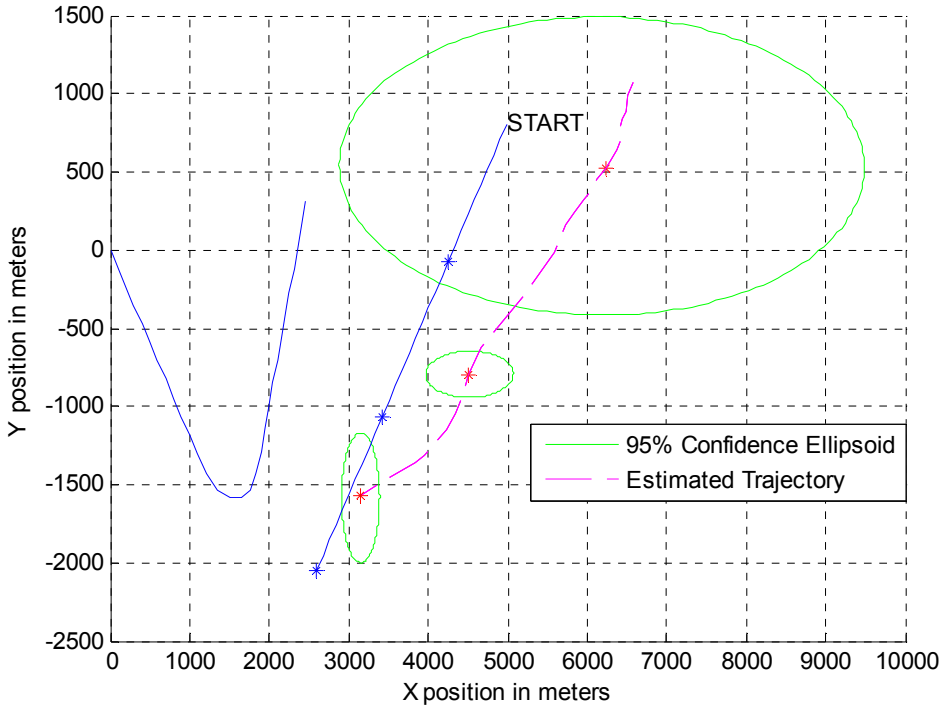

Figure 4-7 Cartesian Kalman Filter Confidence Ellipse Results for Scenario 1

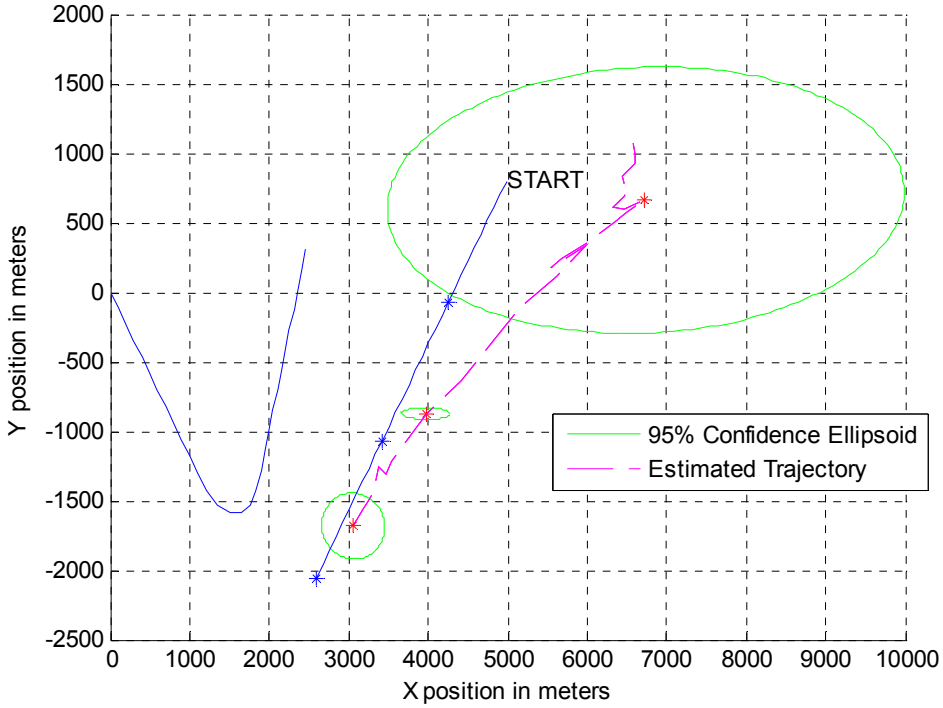

Figure 4-8 MSC Filter Confidence Ellipse Results for Scenario 1

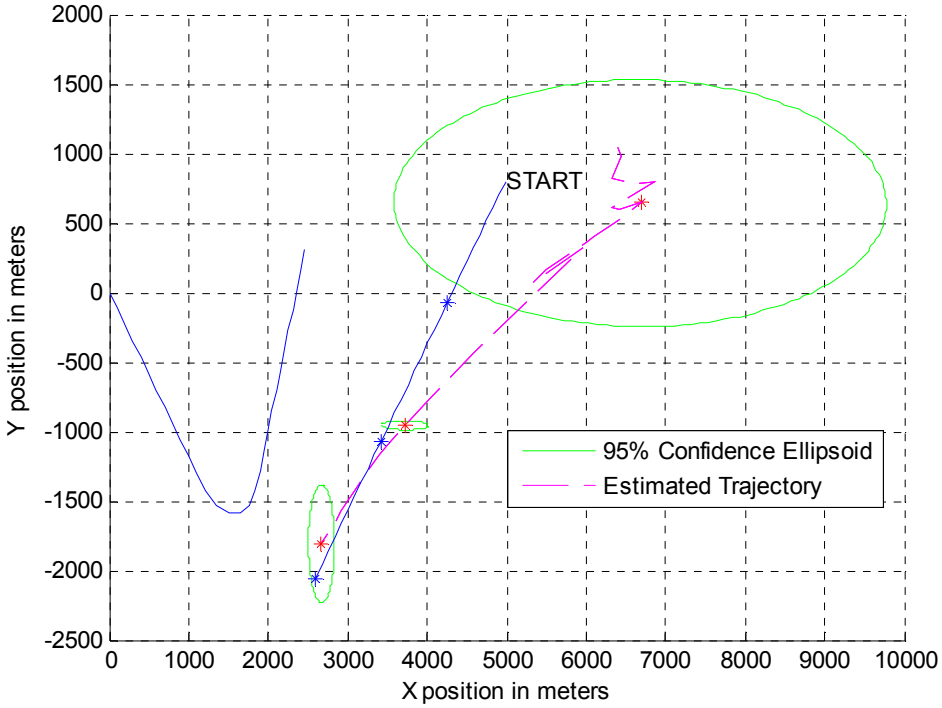

Figure 4-9 RP EKF Confidence Ellipse Results for Scenario 1

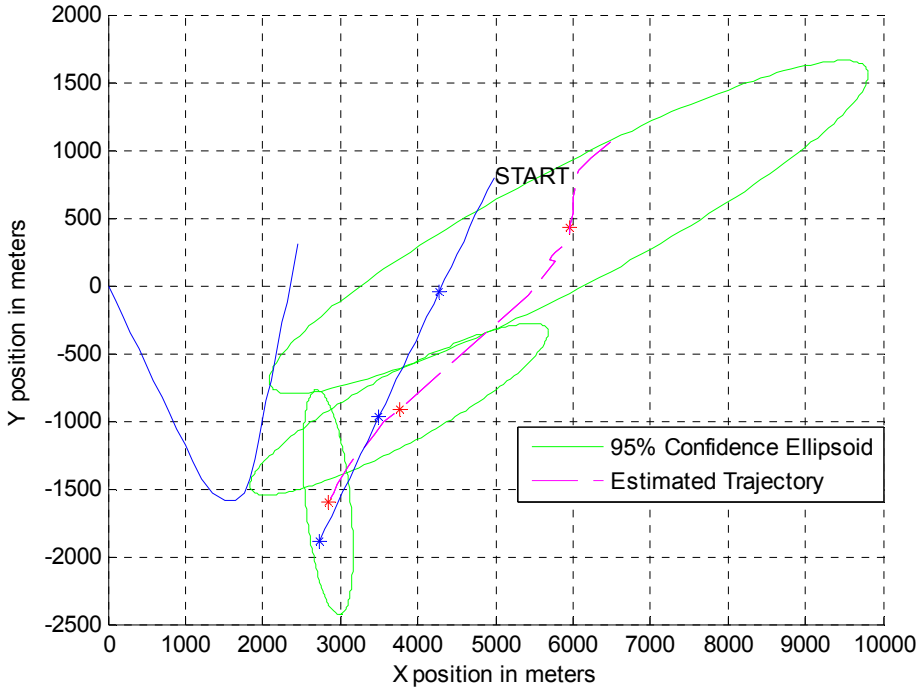

Figure 4-10 Cartesian Regularized PF Confidence Ellipse Results for Scenario 1

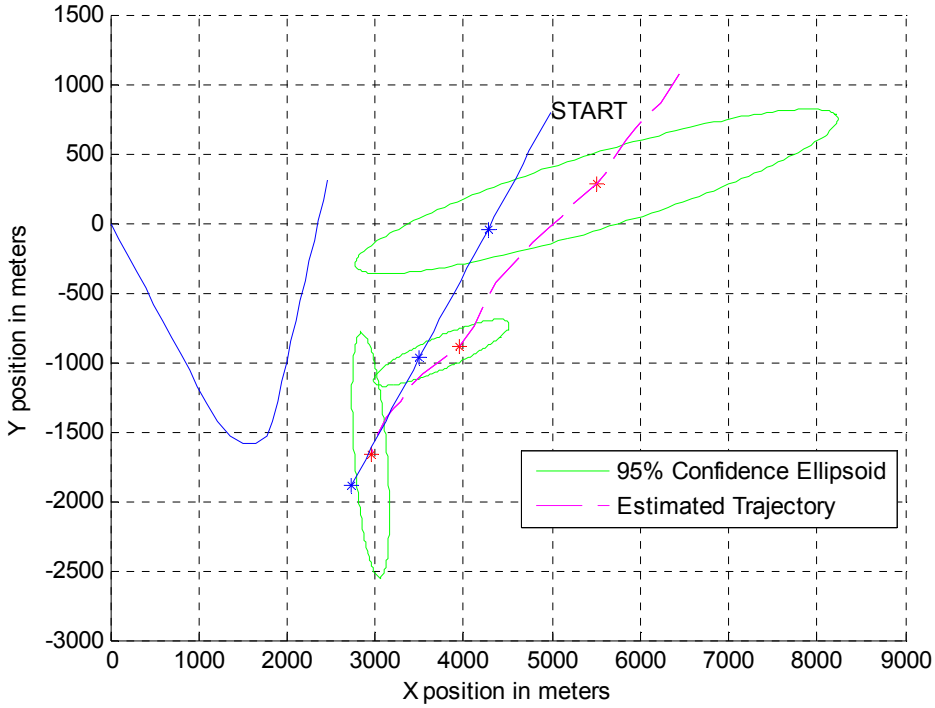

Figure 4-11 MSC Regularized PF Confidence Ellipse Results for Scenario 1

4.3.2 Scenario 2

In scenario 2 observer constantly performs a coordinated turn maneuver with a speed of 155 m/s. Target moves on a straight line with -160° bearing at an initial speed of 125 m/s. Target and observer trajectories are shown in Figure 4-12. Target velocity is propagated using Singer acceleration model. Total time of the simulation is 50 seconds.

100 MC runs are carried out for Scenario 2. The range error is shown in Figure 4-13.

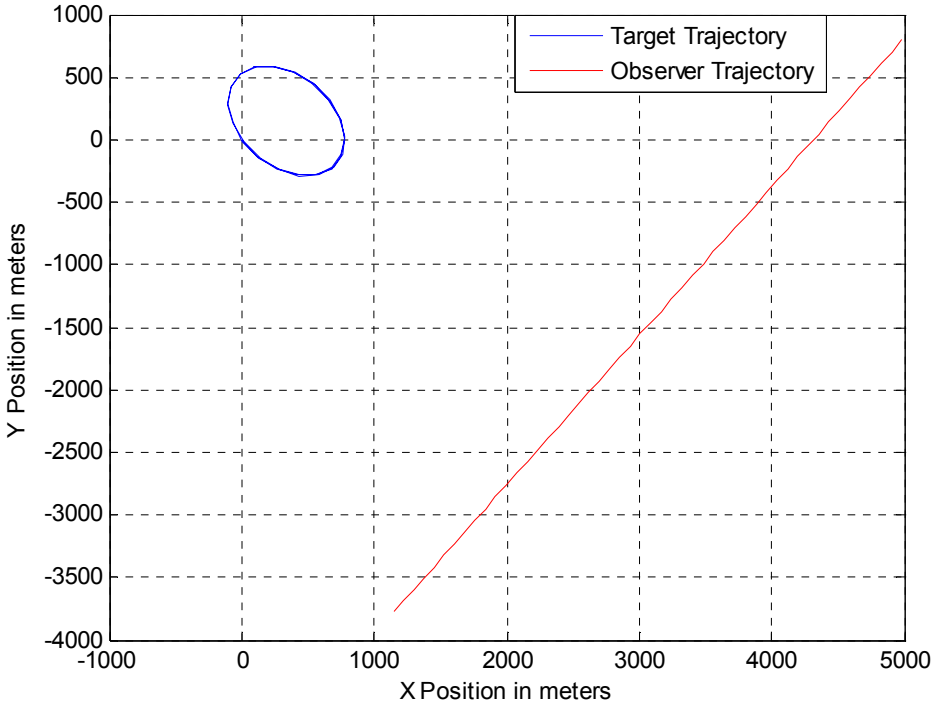

Figure 4-12 Target and Observer Trajectories for Scenario 2

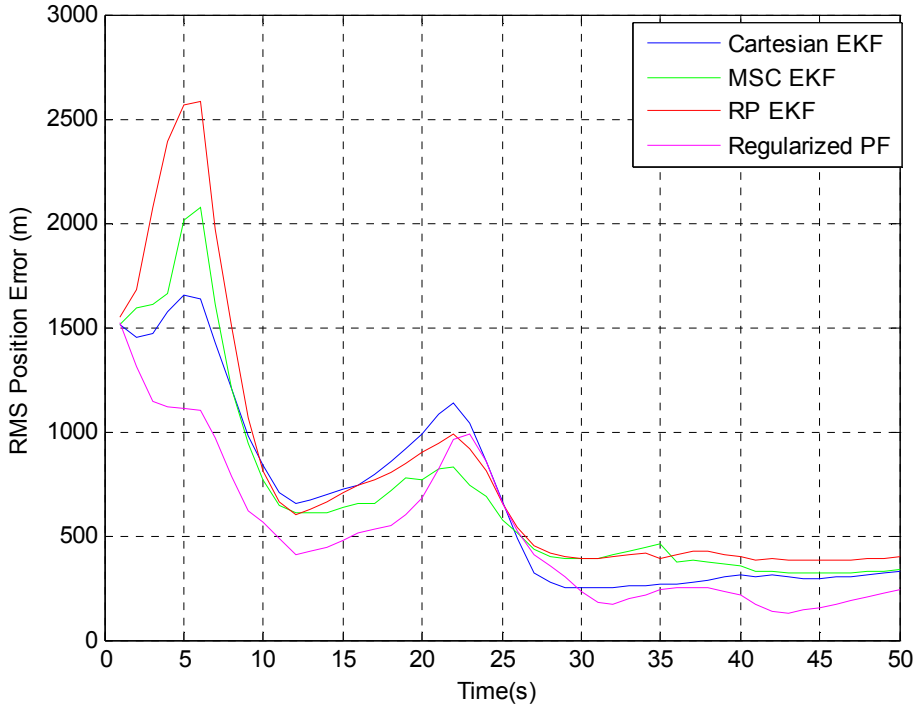

Figure 4-13 Scenario 2 Simulation Results

In the second scenario since observer constantly performs a maneuver, the performances of the studied filters do not differ. After *t=25 s* all the filters converged to the true result.

4.3.3 Scenario 3

In the third scenario, again 2D non-maneuvering target case is investigated. In this scenario the observer follows a leg-by-leg trajectory. Target and observer trajectories are shown in Figure 4-14. Measurement period is taken as 3 seconds which is denoted her as *T*.

The target maintains a constant speed and bearing of 120 m/s and 45° respectively. Observer also maintains a constant speed of 150 m/s, but periodically performs 90°course changes as follows:

from 90° to 0° at $t = (10 + 40k)T$ sec, k = [0; 1; 2]
from 0° to 90° at $t = (30 + 40k)T$ sec, k = [0; 1]

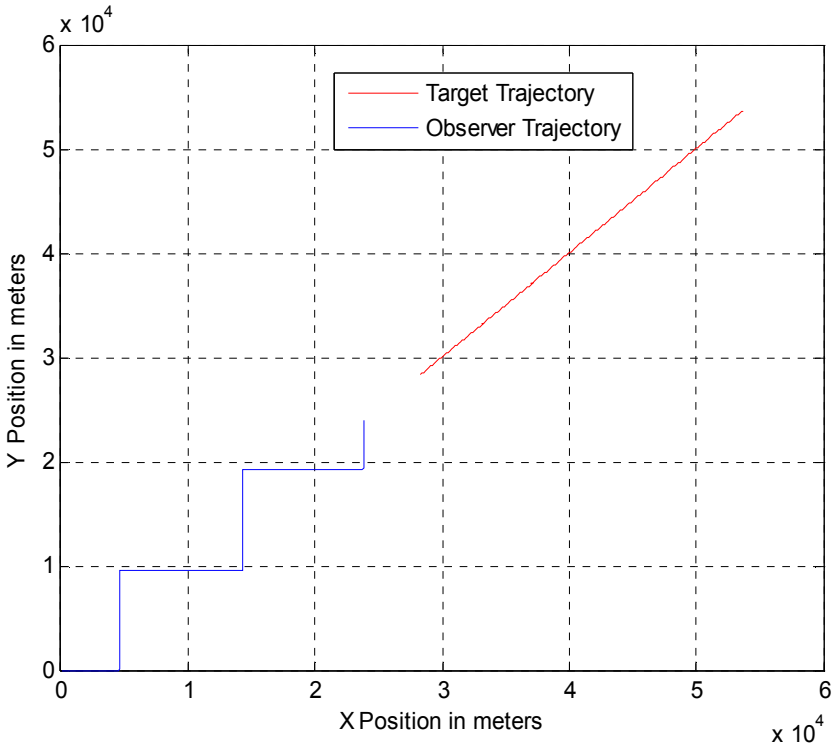

Figure 4-14: Target and Observer Trajectories for Scenario 3

Initial error and simulation parameters can be found in Table 4-4.

100 MC runs are carried out for comparison. The MSC covariance is initialized using Monte Carlo simulation based technique. The range error is shown in Figure 4-15 and a detailed comparison is given in Table 4-5.

Table 4-4 Simulation Parameters for Scenario 2

Measurement Noise	Normally Distributed with μ = 0 and σ = 1.5°
Initial Range Error	Normally Distributed with μ = 0 and σ = 10e3 m
Initial Speed Error	Normally Distributed with μ = 0 and σ = 100 m/s
Initial Course Error	Normally Distributed with μ = 0 and σ = 30°
Process Noise	Singer Model with $w_k(0)$ = 1 m/s and β =0.5°
Particle Number For SIR	5000

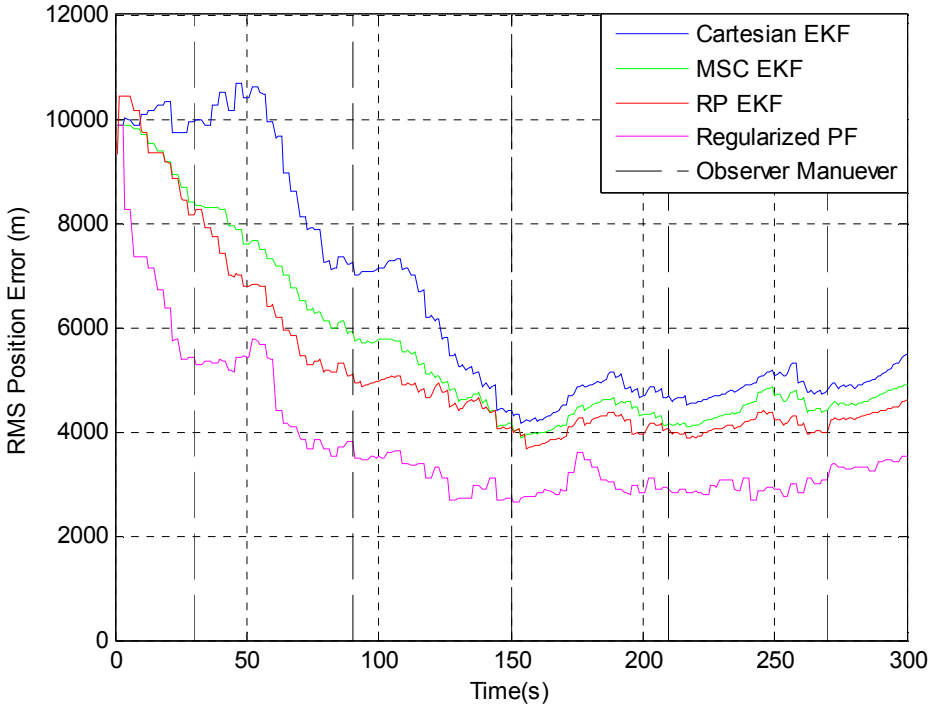

Figure 4-15: Scenario 3 Simulation Results

Table 4-5 Scenario 3 Performance Comparison

Algorithm	RTAMS(m)	RTAMS(m) After the Maneuver	Final Error(m)
Cartesian EKF	6172	4382	46231
MSC EKF	5308	3995	4315
RP EKF	5001	3779	3942
SIR PF	3642	2865	3054

As shown in Table 5-5 RPF particle filter performance is superior than other filters. In addition, since the initial range is large, the root time averaged mean square (RTAMS) error increased compared with Scenario 1.

In Scenario 3 system can be called “observable”. Observer maneuvers are very sharp and this movement increases the range observability for the problem.

4.3.4 Scenario 4

In Scenario 4, 3 dimensional (3D) bearings-only non-maneuvering target tracking is investigated. It is assumed that two measurements are taken from the sensor, one is the angle between x and y coordinates (azimuth angle), and the other is the angle between z and x-y plane (elevation angle). Observer and target have the same trajectory on the *x-y plane* with Scenario 1. They move on the plane z=0 meters ,and z=1000 meters respectively. The trajectory that is used in the simulation is given in Figure 4-16. We imposed a constraint on the z velocity v_z. This constraint is implemented by initializing *z* velocity by zero, and assigning a relatively small positive value for P_{vz} (Equation 5.8).

$$v_z = 0 \text{ m/s}$$
$$\mathrm{P}_{v_z} = 10^{-8} \text{ m}^2/s^2 \qquad (4.10)$$

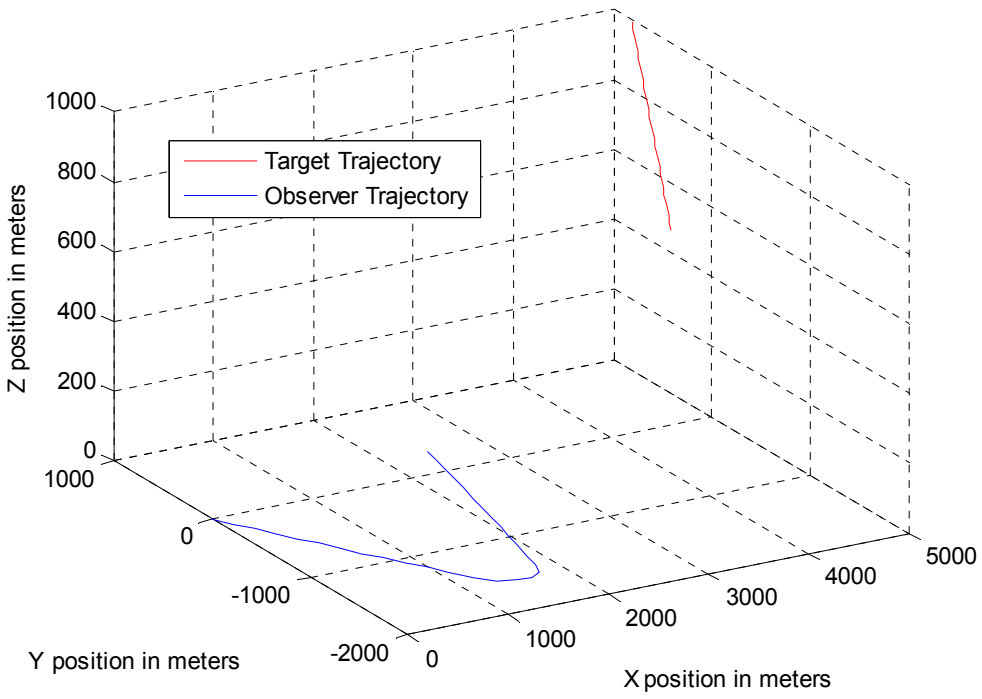

Figure 4-16 Target and Observer Trajectories for Scenario

Simulation parameters for Scenario is given in Table 4-6.

Table 4-6 Simulation Parameters for Scenario 4

Measurement Noise	Normally Distributed with $\mu=[0\,0]^T$,R=$\begin{bmatrix}\sigma^2 & 0\\ 0 & \sigma^2\end{bmatrix}$,$\sigma=1.5°$
Initial Range Error	Normally Distributed with μ = 0 and σ = 2e3 m
Initial Speed Error For v_x and v_y	Normally Distributed with μ = 0 and σ = 100 m/s
Initial Course Error For v_x and v_y	Normally Distributed with μ = 0 and σ = 30°
Process Noise	Singer Model with $w_k(0)$ = 1 m/s and β =0.5°
Particle Count For SIR	5000

100 MC runs are carried out for comparison. The range error is shown in Figure 4-17 and a detailed comparison is given in Table 4-7.

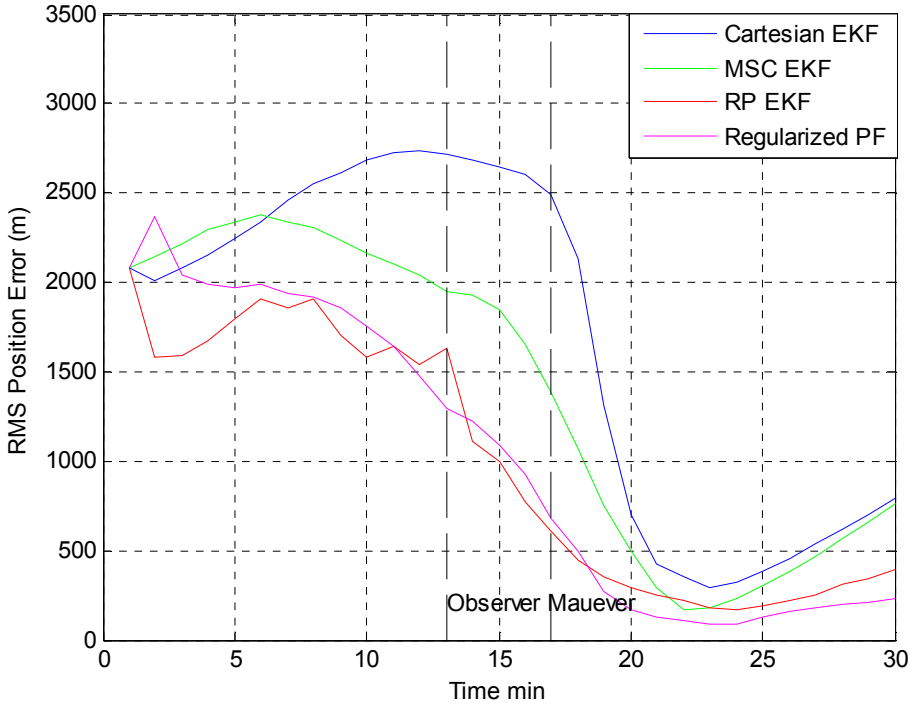

Figure 4-17 Scenario 4 Simulation Results

Table 4-7 Scenario 4 Performance Comparison

Algorithm	RTAMS(m)	RTAMS(m) After the Maneuver	Final Error(m)
Cartesian EKF	1695	882	797
MSC EKF	1309	551	761
RP EKF	987	302	391
Regularized PF	960	223	229

From Figure 4-17 it is seen that the regularized particle filter (Regularized PF) and the range-parameterized extended Kalman Filter (RP-EKF) error values are decreased even before the observer maneuver (*t=15* s). The constraint put on z velocity v_z keeps one state error bounded. Also, two measurements are taken in at a time, so H matrix can bring more improvement to the solution.

4.4 Maneuvering Target Simulations

For maneuvering target problem interacting multiple model filter (IMM), multiple-model particle filter (MM-PF), marginalized particle filter (MPF), and auxiliary particle filter (APF) algorithms are implemented. The theories behind these algorithms were described in Section 3. Initialization procedures are the same as in the non-maneuvering case.

4.4.1 Scenario 5

The target-observer geometry for the maneuvering target simulations are shown in Figure 4-18. Target executes a coordinated turn maneuver between 20 and 25 seconds with a turn rate, $\Omega = 20\,^\circ/s$. After that maneuver, it travels with nearly constant velocity for the rest of the observation period, which lasts 40 seconds. Observer has the same trajectory for the first 30 seconds with Scenario 1. Between 32-35 seconds it executes a second maneuver. After the second maneuver, it travels on a straight line.

We choose the transitional probability matrix as:

$$\Pi = \begin{bmatrix} 0.9 & 0.05 & 0.05 \\ 0.4 & 0.5 & 0.1 \\ 0.4 & 0.1 & 0.5 \end{bmatrix} \quad (4.\ 11)$$

Column 1, 2 and 3 represent the constant velocity model, clockwise coordinated turn model, and counter-clockwise coordinated turn model, respectively. A high probability value is assigned for transition to CV model. Also, it is assumed that passing from one CT model to another has a low probability.

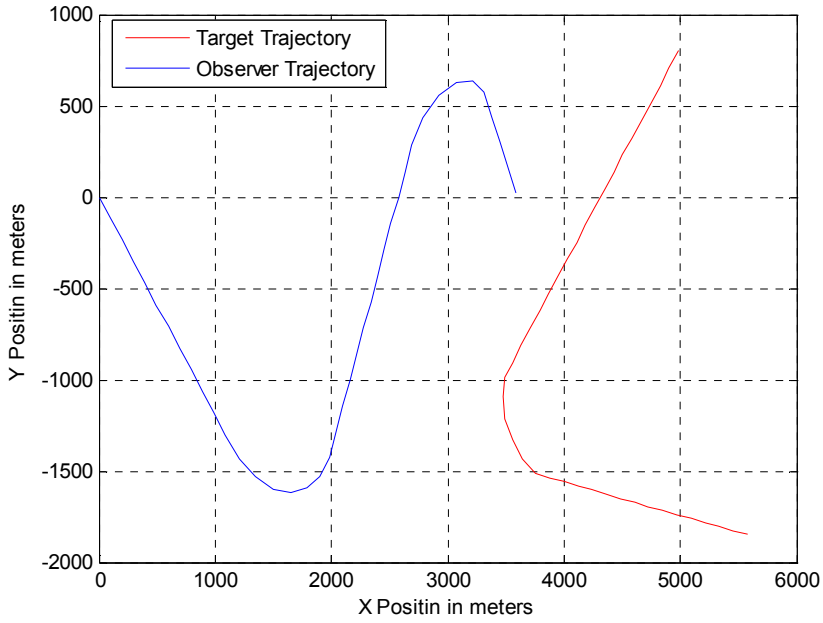

Figure 4-18 Target and Observer Trajectories for Scenario 5

Initial error, noise and simulation parameters are given in Table 4-8.

Table 4-8 Simulation Parameters for Scenario 5

Measurement Noise	Normally Distributed with μ = 0 and σ = 1.5°
Initial Range Error	Normally Distributed with μ = 0 and σ = 20e3 m
Initial Speed Error	Normally Distributed with μ = 0 and σ = 100 m/s
Initial Course Error	Normally Distributed with μ = 0 and σ = 30°
Process Noise	Singer Model with $w_k(0)$ = 1 m/s and β =0.5°

Particle Number For SIR	5000

Two different cases are investigated for maneuvering target tracking.

Firstly, it is assumed that the turn rate is known exactly. The range error for mentioned case is shown in Figure 4-19. The error ellipses for IMM EKF and multiple model filter at time 10,20,30,40 are shown in Figure 4-20 and Figure 4-21

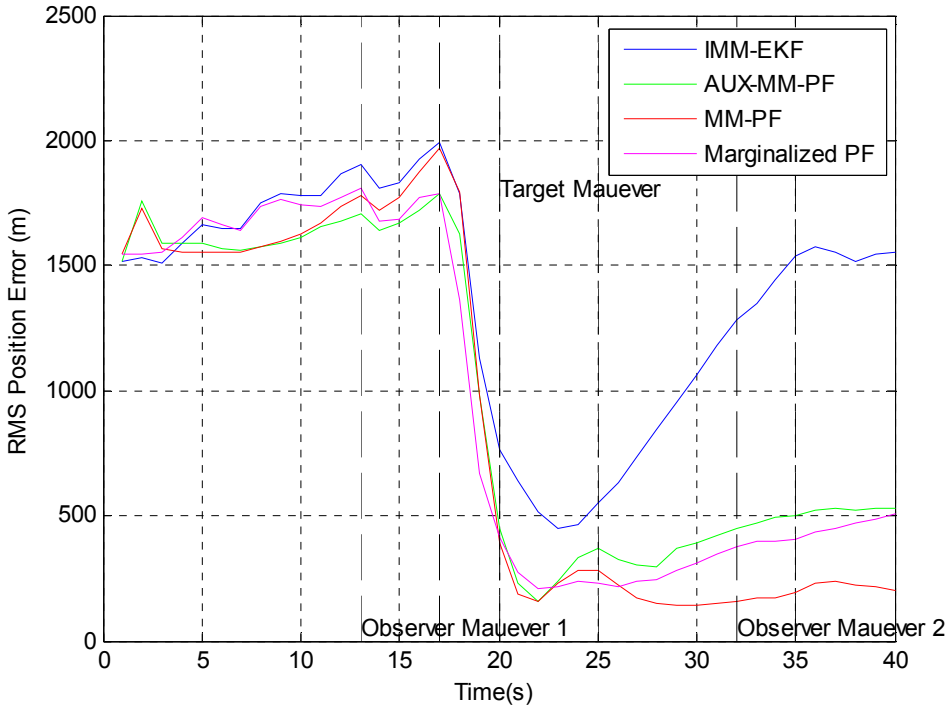

Figure 4-19 Maneuvering Target with Known Turn Rate Simulation Results

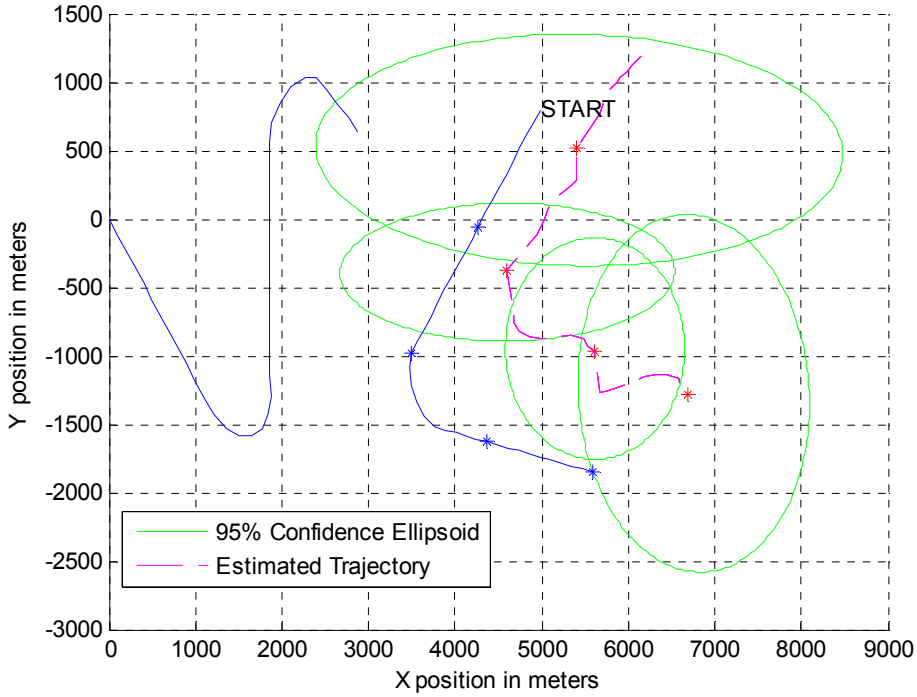

Figure 4-20 IMM EKF Confidence Ellipse Results for Scenario 5

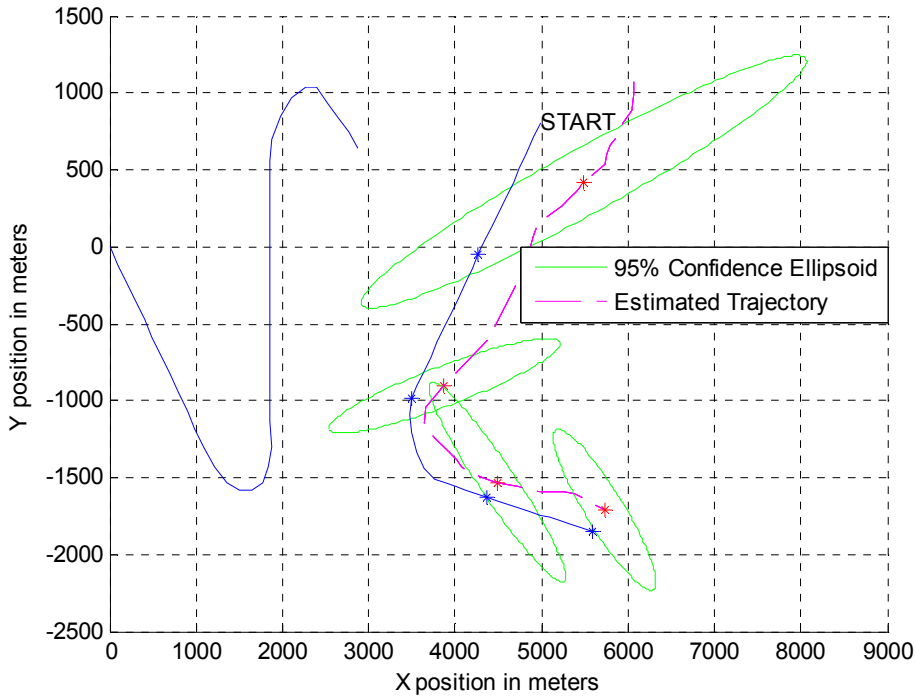

Figure 4-21 Multiple Model PF Confidence Ellipse Results for Scenario 5

From Figure 4-19, it is clear that the performance of the IMMEKF is poor compared to the other three filters. Also, IMM-EKF has one divergent track which is not taken into consideration in the calculations. A detailed comparison is given in Table 4-9. MMPF has the best performance, which has a mean range error of 377 meters after the observer maneuvers. The performance differences can be explained as follows: There are two sources of approximations in IMM-EKF. Firstly, IMM-EKF approximates the non-Gaussian density by a Gaussian and use linearization approximation for the Kalman Filter. Secondly, the mode for the target is estimated using the mode probabilities with merging of histories.

For particle solutions the non-Gaussian posterior density is approximated in a near optimal manner and there is no linearization error. In addition, the mode is estimating by the mode probabilities with no merging of histories.

It is interesting that the Multiple Model Particle Filter (MM-PF) solution is better than Marginalized Particle Filter solution. Since in Marginalized PF solution the optimal importance density is used, it is expected to have better performance. However, it uses EKF for (a local linearisation approximation) to compute the mode-conditioned filtered estimates. It can be concluded that linearization error is dominant over the importance density error.

Table 4-9 Scenario 5 with Known Turn Rate Performance Comparison

Algorithm	RTAMS(m)	RTAMS(m) After the Maneuver	Final Error(m)
IMM	1386	1157	1722
AUX-MM-PF	970	534	529
MM-PF	886	377	199
Marginalized PF	948	457	506

Mode probabilities for IMM-EKF and MM-PF are shown in Figure 4-22 and Figure 4-23, respectively.

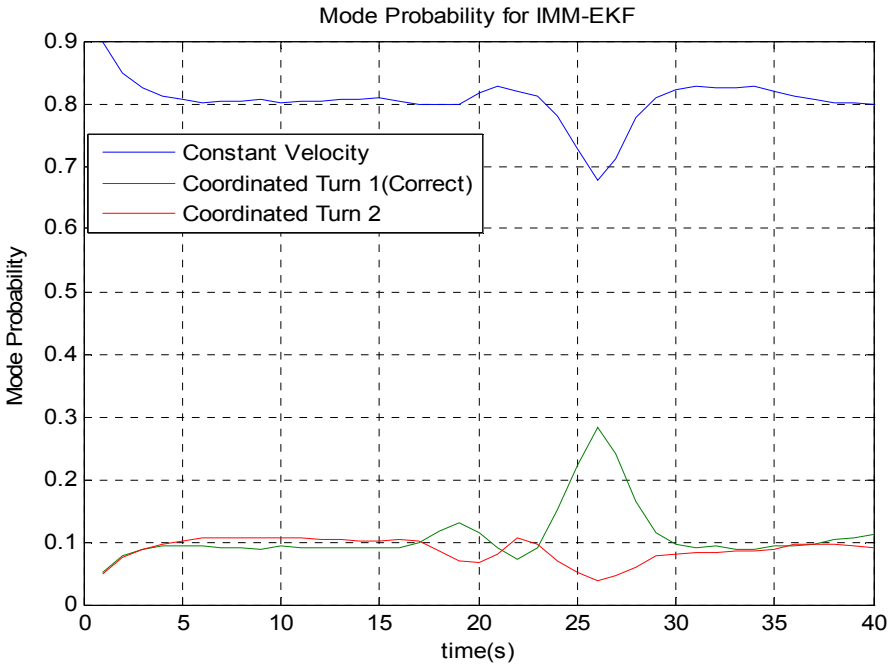

Figure 4-22 Mode Probabilities of IMM-EKF

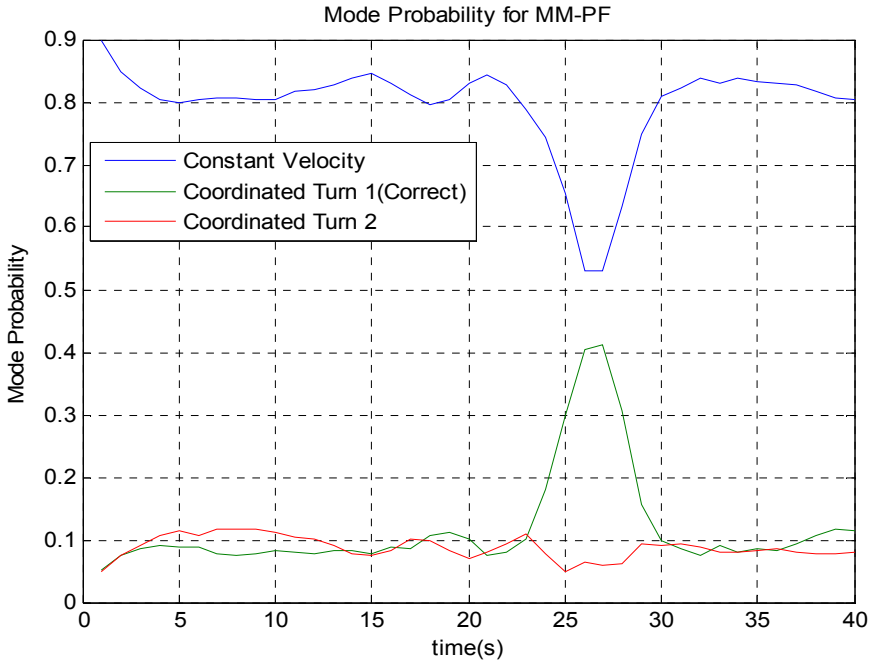

Figure 4-23 Mode Probabilities of MM-PF

Next, maneuvering a target with unknown coordinate turn is investigated. Modification is done only for the interacting multiple model extended Kalman Filter.

In the first simulation set with unknown turn rate, turn rate estimate is sampled from a Gaussian density with $\mu = 20°/\mathrm{s}$ and $\sigma = 5°/\mathrm{s}$,where the actual turn rate is $\Omega = 20\ °/s$. For this set 200 Monte Carlo runs are carried .The other parameters are the same as the case with known turn rate, which are given in Table 4-8. The simulation results are shown in Figure 4-24 and are summarized in Table 4-10.

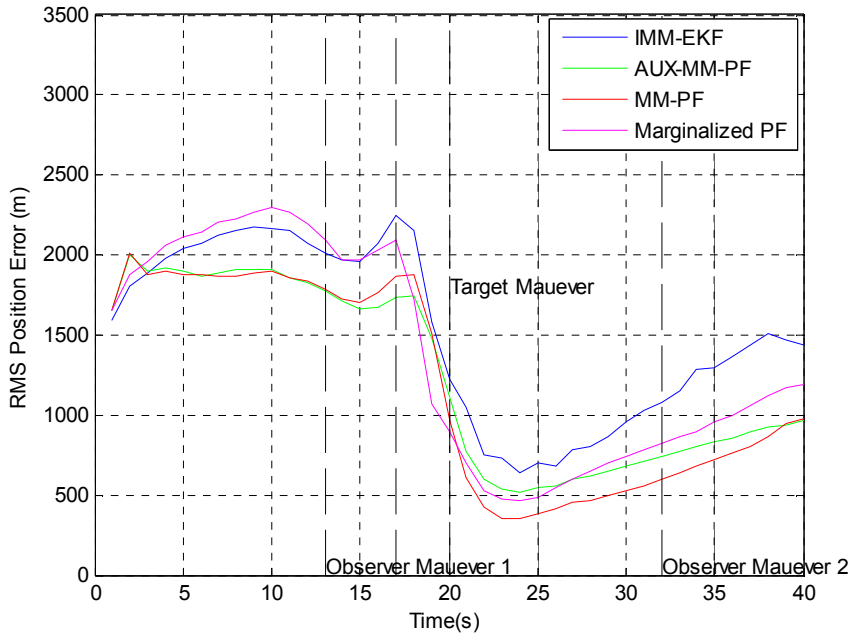

Figure 4-24 Maneuvering Target with unknown Turn Rate Case 1 Simulation Results

Table 4-10 Scenario with unknown Turn Rate Performance Comparison

Algorithm	RTAMS(m)	RTAMS(m) After the Maneuver	Final Error(m)
IMM-EKF	1290	929	1116
AUX-MM-PF	986	540	642
MM-PF	905	478	662
Marginalized PF	933	571	883

In Table 4-10 IMM-EKF seems to have better results than the case with known turn rate. However, IMM-EKF produced 6 divergent tracks for this case. Therefore, it can be said that overall performance is degraded. Divergent tracks are discarded in evaluating the values of Table 4-10. Despite the decline in performance of particle filters, it can be considered as successful.

For the second simulation set with unknown turn rate, turn rate estimate is sampled from a Gaussian density with $\mu = 20°/s$ and $\sigma = 10°/s$. For this set 200 Monte Carlo runs are carried, too. The simulation results are shown in Figure 4-25 and a detailed comparison is given in Table 4-11.

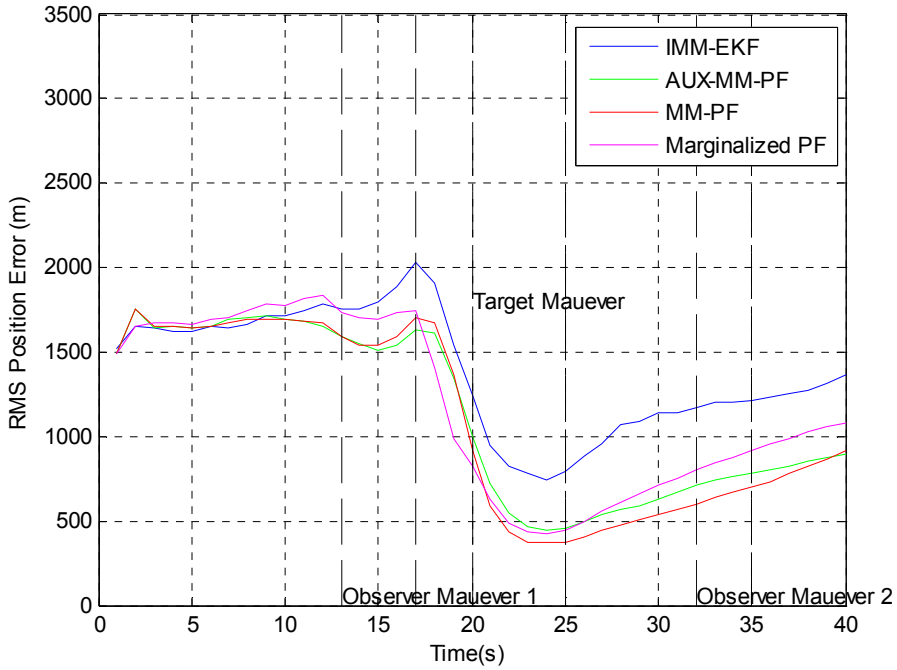

Figure 4-25 Maneuvering Target with unknown Turn Rate Case 2 Simulation Results

Table 4-11 Scenario with unknown Turn Rate Performance Comparison

Algorithm	RTAMS(m)	RTAMS(m) After the Maneuver	Final Error(m)
IMM-EKF	1267	926	1191
AUX-MM-PF	966	607	718
MM-PF	947	561	722
Marginalized PF	1028	647	1050

IMM-EKF resulted in 8 divergent tracks for this simulation set. If one compare Table 4-10 and Table 4-11, it can be seen that the performance of the particle filter solutions are degraded, too. Nevertheless, it is acceptable in terms of divergent tracks. Since the number of divergent tracks for IMM-EKF solution increases, it can be said its sensitivity for turn rate error is much more than the particle filter solutions.

It can be concluded that even the IMM-EKF is modified in order to overcome the unknown rate error, its performance degraded substantially. On the other hand, the particle filter solutions performance is acceptable in terms of convergence.

CHAPTER 5

CONCLUSIONS

In this thesis, the problem of bearings-only target tracking is examined using recursive estimation algorithms. Particle filter algorithms are compared with classical Kalman filter based algorithms for different system models and scenarios. Main focus is obtaining a robust estimation algorithm for bearings-only target tracking problem.

Mainly two different problems are investigated, non-maneuvering and maneuvering target tracking. The system models for mentioned problems are described in detail. It is well known that the estimation filters' performances depend on the characteristics of the model. Expressing the problem in a suitable way increases the resultant performance for any filtering method. In the tracking problem considered here different coordinate axes choices are compared.

After defining the state space model, related filtering techniques are described. Firstly, extended Kalman Filter based solutions are introduced. Secondly, the sequential Monte Carlo methods, i.e. particle filters, and the theory behind them are introduced. Based on particle filter solutions, the key points like the degeneracy phenomenon and the choice of the importance density are underlined. In this work regularization is applied to only the unobservable state variable namely the inverse of the range. We believe that this choice will improve the filer performance by keeping the variance of the particles smaller compared to application of the regularization to full state. In the context of tracking maneuvering target tracking, a different importance density, auxiliary multiple

model particle filter and marginalized particle filter are studied in order to provide improvements.

In order to track non-maneuvering target extended Kalman Filter and regularized particle filter are used both in Cartesian coordinates and modified spherical coordinates (MSC). The problem of the covariance matrix initialization in MSC is addressed. A set of weighted EKFs each with a different initial range estimate, range-parameterized extended Kalman Filter is examined. The prior distribution is used as the importance density for the sake of simplicity in the regularized particle filter. The drawback of the regularization step is the increase in the variance of the estimates. In order to overcome this problem system model is expressed in MSC, and regularization step is only applied for inverse of the range state.

Maneuvering target tracking problem is investigated using multiple model filters. Modeling is done both in Cartesian coordinates and in spherical coordinates. In MSC we have used EKF based IMM filter. The effect of the error on the turn rate is investigated for maneuvering target-tracking problem. An extra state is added for turn rate in IMM filter, and the system model is modified. Even after this modification, it produces divergent tracks. The number of divergent tracks increases with the turn rate error.

For Cartesian coordinate representation we have used particle based approaches that uses multiple model idea. The basic method is the multiple model particle filter (MMPF). Two modifications of the basic method are also investigated namely, auxiliary multiple model particle filter (AUX-MMPF) and marginalized particle filter (MPF). One noticeable result is that MMPF performance is superior to the MPF. It is also observed that particle filter errors increase proportionally with the turn rate error. Nevertheless, the particle filter sate estimates converged to the true estimate.

The results confirm the superior performance of particle filter based algorithms against the conventional KF-based IMM algorithms. It is an expected result since even the system model is simple; the model is non-linear including trigonometric functions.

As a future work, different scenarios should be implemented to investigate the studied filters performance. To improve the performance of the tracking filters the initialization of covariance matrix need to be investigated more. The particle filter expressed in modified spherical coordinates should be studied in detail. Applying regularization step only for unobservable state increased the performance of the estimation algorithm. This problem can be used to understand the advantages and disadvantages of regularization step used in particle filter applications. 3 dimensional maneuvering target tracking problem can be extended by defining new models in order to track the turn maneuvers on the y-z plane. According to the specific tracking problem here, further modifications can be done in particle filter algorithm.

REFERENCES

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

[2] Aidala, V. J. and S. E. Hammel*, "Utilization of Modified Polar Coordinates for Bearings-only Tracking,"* IEEE Trans. on Automatic control, Vol. AC-28, No. 3, pp. 283-294, March 1983.

[3] Robinson, P. and Yin, M. "*Modified spherical coordinates for radar"* Proc. AIAA Guidance, Navigation and Control Conference, pp. 55-64, March 1994

[4] J. M. C. Clark, R. B. Vinter, and M. M. Yaqoob, "*Shifted Rayleigh filter: a new algorithm for bearings only tracking*," IEEE Trans. Aerospace and Electronic Systems, vol. 43, no. 4, pp. 1373–1384, Oct. 2007.

[5] T. L. Song and J. L. Speyer, "*A stochastic analysis of a modified gain extended Kalman filter with applications to estimation with bearings only measurements,*" IEEE Trans. Automat. Contr., vol. 30, no. 10, pp. 940–949, Oct. 1985.

[6] S. Julier, J. Uhlmann, and H. F. Durrant-Whyte, "*A new method for the nonlinear transformation of means and covariances in filters and estimators*," IEEE Trans. Automat. Contr., vol. 45, no. 3, pp. 477–482, Mar. 2000.

[7] S. SadhuS. Mondal, M. Srinivasan and T.K. Ghosha, "*Sigma point Kalman filter for bearing only tracking"* Journal Signal Processing, vol 86 Issue 12, December 2006

[8] N. Peach, MA, C. Eng, *"Bearings-only tracking using a set of range-parameterised extended Kalman filters"* Control Theory and Applications, IEE Proceedings, vol 142, Issue 1, pp.73 - 80, Jan 1995

[9] Fredrik Gustafsson, "*Particle Filter Theory and Practice with Positioning Applications*", IEEE A&E Systems Magazine, vol 25,No 74,July 2010.

[10] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, *"A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking*," IEEE Trans. on Signal Processing, vol. 50, no. 2, pp. 174–188, Feb. 2002.

[11] Gordon N. ea,*." Beyond The Kalman Filter*", Artech House, 2004.

[12] T. Kirubarajan, Y. Bar-Shalom, and D. Lerro, "*Bearings-only tracking of maneuvering targets using a batch-recursive estimator,"* IEEE Transactions on Aerospace and Electronic Systems, vol. 37, no. 3, pp. 770–780, 2001.

[13] J.-P. Le Cadre and O. Tremois, "*Bearings-only tracking for maneuvering sources*" IEEE Transactions on Aerospace and Electronic Systems, vol. 34, no. 1, pp. 179–193, 1998.

[14] A. Doucet, N. J. Gordon, and V. Krishnamurthy, *"Particle filters for state estimation of jumpMarkov linear systems*" IEEE Trans. Signal Processing, vol. 49, no. 3, pp. 613–624, 2001.

[15] R. Karlsson and N. Bergman, "*Auxiliary particle filters for tracking a maneuvering target*" Proc. 39th IEEE Conference on Decision and Control, vol. 4, pp. 3891–3895, Sydney, Australia, December 2000.

[16] C. Andrieu and A. Doucet. *"Particle Filtering for partially observed Gaussian state space models"* Journal of the Royal Statistical Society, vol. 64, no. 4, pp. 827-836, 2002

[17] C. Jauffret and D. Pillon. "*Observability in Passive TargetMotion Analysis*" In IEEE Transactions on Aerospace and Electronic Systems, volume 32, October 1996.

[18] E. Fogel and M. Gavish. "*Nth-Order Dynamic Target Observability From Angle Measurements*" In IEEE Transactions on Aerospace and Electronic Systems, volume 24, May 1988.

[19] Passerieux, J.M.; Van Cappel, D., *"Optimal observer maneuver for bearings-only tracking*" IEEE Transactions on Aerospace and Electronic Systems, volume 24, July 1988.

[20] Rickard Karlsson "*Simulation Based Methods for Target Tracking*", PHD Thesis Division of Automatic Control & Communication systems Department of Electrical Engineering Linkopings University 2002.

[21] S. Blackman and R. Popoli. *"Design and Analysis of Modern Tracking Systems"* Artech House, 1999.

[22] X. RONG LI e.a,"*Survey of Maneuvering Target Tracking Part I: Dynamic Models*" IEEE Transactıons On Aerospace And Electronıc Systems Vol. 39, No. 4 pp 1333-1360

[23] Singer, R. A. "*Estimating optimal tracking filter performance for manned aneuvering targets"* Transactions on Aerospace and Electronic Systems, Transactions on Aerospace and Electronic Systems, AES-6, pp. 473–483, July 1970

[24] J. Le Cadre and O. Trémois. "O*ptimization of the Observer Motion Using Dynamic Programming"* International Conference on Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 I

[25] Y.Bar.Shalom,X.R.Li, and T. Kirubajan "*Estimation with Applications To Tracking and Navigation"*, John Wiley&Sons NY,2001.

[26] Y. Bar-Shalom and X. Li. Estimation and Tracking: "*Principles, Techniques, and Software*". Artech House, 1993.

[27] Fredrik Gustafsson, "*Particle Filter Theory and Practice with Positioning Applications*", IEEE A&E Systems Magazine, Vol 25,No 74,July 2010..

[28] Bobrovsky, B.; Zakai, M., "*A lower bound on the estimation error for Markov processes"*, IEEE Trans.Automatic Control" ,vol 24 April 1979

[29] Kirubarajan, T., Bar-Shalom, Y., Blair, W. D., and Watson, G. A., *"IMMPDA Solution to Benchmark for Radar Resource Allocation and Tracking in the Presence of ECM"*, IEEE Trans. Aerospace and Electronic Systems, Vol. 34, No. 3, pp. 1023-1036, October 1998.

[30] G. Liu, E.Gao, C Fan, "*Multirate Interacting Multiple Model Algorithm Combined with Particle Filter for Nonlinear/Non-Gaussian Target Tracking*" 16th International Conference on Artificial Reality and Telexistence--Workshops (ICAT'06)

[31] A.Athalye, S.Hong,P.M. Djuric "Distrubuted *"Architecture and Interconnection Scheme for Multiple Model Particle Filters",* ICASSP, IEEE, 2006

[32] Walter Grossman *"Bearings-Only Tracking: A Hybrid Coordinate System Approach"* Journal Of Guidance, Control, And Dynamics Vol. 17, No. 3, May-June 1994

[33] J.C.Hassab., "*Underwater Signal Data processing"* .Boca Raton FL:CRC Press 1989

[34] Wilfred Schofield,Mark Breach *"Engineering surveying*". Butterworth-Heinemann 2007.

[35] T. Bailey, "*Constrained initialisation for bearing-only SLAM*" in Proc. IEEE Int. Conf. Robotics Automation, 2003, pp. 1966–1971.

[36] M. Deans and M. Hebert, "*Experimental comparison of techniques for localization and mapping using a bearing-only sensor*" in Experimental Robotics VII, D. Rus and S. Singh, Eds. New York: Springler Verlag, 2001.

[37] Schwar D.M. "ftp://ftp.mathworks.com/pub/contrib/v4/stats/ftest", 1998.

APPENDIX A

MSC PLANT EQUATIONS

The derivation is the same as in [4] with obvious notation changes.
The derivation for the first three equations is straightforward as shown below:

$$\frac{d}{dt}\left(\frac{1}{r}\right) = -\frac{\dot{r}}{r^2} = -\frac{1}{r}\frac{\dot{r}}{r} = -y_1 y_4 \tag{A.1}$$

$$\frac{d}{dt}(\varphi) = \frac{\vartheta}{\cos(\theta)} = \frac{y_5}{\cos(y_3)} \tag{A.2}$$

$$\frac{d}{dt}(\theta) = \dot{\theta} = y_6 \tag{A.3}$$

To find the derivatives of $\dot{r}/r$, Ω and $\dot{\theta}$, first express the relative coordinates X, Y, and Z in terms of r, φ, and θ and then calculate their first and second derivatives. The coordinates are:

$$\begin{aligned} X &= r\cos(\theta)\cos(\varphi) \\ Y &= r\cos(\theta)\sin(\varphi) \\ Z &= -r\sin(\theta) \end{aligned} \tag{A.4}$$

The Cartesian rates are:

$$\begin{aligned}
\dot{X} &= \dot{r}\cos(\theta)\cos(\varphi) - r\dot{\theta}\sin(\theta)\cos(\varphi) - r\dot{\varphi}\cos(\theta)\sin(\varphi) \\
\dot{Y} &= \dot{r}\cos(\theta)\sin(\varphi) - r\dot{\theta}\sin(\theta)\sin(\varphi) + r\dot{\varphi}\cos(\theta)\cos(\varphi) \\
\dot{Z} &= -\dot{r}\sin(\theta) - r\dot{\theta}\cos(\theta)
\end{aligned} \tag{A.5}$$

The Cartesian accelerations are:

$$\begin{aligned}
\ddot{X} &= \ddot{r}\cos(\theta)\cos(\varphi) - 2\dot{r}\dot{\theta}\sin(\theta)\cos(\varphi) - 2\dot{r}\dot{\varphi}\cos(\theta)\sin(\varphi) \\
&\quad + 2r\dot{\theta}\dot{\varphi}\sin(\theta)\sin(\varphi) - r\dot{\theta}^2\cos(\theta)\cos(\varphi) - r\dot{\varphi}^2\cos(\theta)\cos(\varphi) \\
&\quad - r\ddot{\theta}\sin(\theta)\cos(\varphi) - r\ddot{\varphi}^2\cos(\theta)\sin(\varphi) \\
\ddot{Y} &= \ddot{r}\cos(\theta)\sin(\varphi) - 2\dot{r}\dot{\theta}\sin(\theta)\sin(\varphi) + 2\dot{r}\dot{\varphi}\cos(\theta)\cos(\varphi) \\
&\quad - 2r\dot{\theta}\dot{\varphi}\sin(\theta)\cos(\varphi) - r\dot{\theta}^2\cos(\theta)\sin(\varphi) - r\dot{\varphi}^2\cos(\theta)\sin(\varphi) \\
&\quad - r\ddot{\theta}\sin(\theta)\sin(\varphi) + r\ddot{\varphi}^2\cos(\theta)\cos(\varphi) \\
\ddot{Z} &= -\ddot{r}\sin(\theta) - 2\dot{r}\dot{\theta}\cos(\theta) + \dot{r}\dot{\theta}^2\sin(\theta) - r\ddot{\theta}\cos(\theta)
\end{aligned} \tag{A.6}$$

Next the Cartesian accelerations are expressed in observer; X^o is along the radial and Y^o is tangential vectors observer coordinates. The following rotation matrix transforms a vector from inertial (I) into antenna (A) coordinates.

$$C_R^o = \begin{bmatrix} \cos(\theta)\cos(\varphi) & \cos(\theta)\sin(\varphi) & -\sin(\theta) \\ -\sin(\varphi) & \cos(\varphi) & 0 \\ \sin(\theta)\cos(\varphi) & \sin(\theta)\sin(\varphi) & \cos(\theta) \end{bmatrix} \tag{A.7}$$

The third row of equation A.6 gives:

$$\begin{aligned}
\ddot{Y} &= 2\dot{r}\omega - r\dot{\theta}\omega\tan(\theta) + r\dot{\omega} \\
&= \mathrm{a}_{Ht}^a - \mathrm{a}_{Ho}^a
\end{aligned} \tag{A.8}$$

This can be solved for $\ddot{\theta}$

$$\begin{aligned}\frac{d\dot{\theta}}{dt} &= -2\left(\frac{\dot{r}}{r}\right)\dot{\theta} + \Omega^2\tan(\theta) - \frac{a_{vt}}{r} + \frac{a_{ot}}{r} \\ &= -2y_4y_6 - y_5^2\tan(y_3) + y_1(a_{Vt} - a_{Vo})\end{aligned} \tag{A.9}$$

This is the sixth row of state equation.

The second row of A.6 gives:

$$\begin{aligned}\ddot{Y} &= 2\dot{r}\vartheta - r\dot{\theta}\vartheta\tan(\theta) + r\dot{\Omega} \\ &= \mathrm{a}_{Ht}^{a} - \mathrm{a}_{Ho}^{a}\end{aligned} \tag{A.10}$$

Using A.10 and A.9 $\dot{\Omega}$ can be shown as:

$$\begin{aligned}\frac{d\dot{\vartheta}}{dt} &= \left(-2\frac{\dot{r}}{r} + \dot{\theta}\tan(\theta)\right)\Omega + \frac{a_{Ht}}{r} - \frac{a_{Ht}}{r} \\ &= y_5(-2y_4 + y_6\tan(y_3)) - y_1(a_{Ht} - a_{Ho})\end{aligned} \tag{A.11}$$

Finally if we combine the first row of equation A.6 with A.7 $\ddot{X}$ can be given as:

$$\begin{aligned}\ddot{X} &= \ddot{r} - r\dot{\theta}^2 + r\vartheta^2 \\ &= \mathrm{a}_{Rt}^{a} - \mathrm{a}_{Ro}^{a}\end{aligned} \tag{A.12}$$

Using this equation final plant equation can be calculated as:

$$\begin{aligned}\frac{d}{dt}\left(\frac{\dot{r}}{r}\right) &= \dot{\theta}^2 + \dot{\vartheta}^2 - \left(\frac{\dot{r}}{r}\right)^2 + \frac{a_{Rt}}{r} - \frac{a_{Rt}}{r} \\ &= y_6^2 + y_5^2 - y_4^2 - y_1(a_{Rt} - a_{Ro})\end{aligned} \tag{A.13}$$

APPENDIX B

RELATION BETWEEN CARTESIAN COORDINATES and MODIFIED SPHERICAL COORDINATES

This appendix describes the relations between the Cartesian coordinates and the modified spherical coordinates (MSC). The Jacobians for the transformation equations are also given. The Cartesian coordinates are denoted by x^{car} and the MSC by y^{msc}. The state vector are given below:

$$x^{car} = (x_1\ x_2\ x_3\ x_4\ x_5\ x_6)^T = (x\ y\ z\ \dot{x}\ \dot{y}\ \dot{z})^T$$
$$y^{msc} = (z_1\ z_2\ z_3\ z_4\ z_5\ z_6)^T = (\frac{1}{r}\ \varphi\ \theta\ \frac{\dot{r}}{r}\ \Omega\ \dot{\theta})^T$$
$$\Omega = \dot{\varphi}\cos\theta$$

B.1 Transformations from y^{msc} to x^{car}

$$y_4 = \frac{\dot{r}}{r} = \frac{x_1x_4 + x_2x_5 + x_3x_6}{x_1^{\ 2} + x_2^{\ 2} + x_3^{\ 2}}$$
$$y_5 = \Omega = \dot{\varphi}\cos\theta = \frac{x_1x_5 - x_2x_4}{x_1^{\ 2} + x_2^{\ 2}}\cos(\arctan(\frac{-x_3}{\sqrt{x_1^{\ 2} + x_2^{\ 2}}}))$$
$$y_6 = \dot{\theta} = \frac{-x_6(x_1^{\ 2} + x_2^{\ 2}) + x_3(x_1x_4 + x_2x_5)}{(x_1^{\ 2} + x_2^{\ 2} + x_3^{\ 2})\sqrt{x_1^{\ 2} + x_2^{\ 2}}}$$

$$y_1 = \frac{1}{r} = \frac{1}{\sqrt{x_1^{\,2} + x_2^{\,2} + x_3^{\,2}}}$$

$$y_2 = \varphi = \arctan(\frac{x_2}{x_1})$$

$$y_3 = \theta = \arctan(\frac{-x_3}{\sqrt{x_1^{\,2} + x_2^{\,2}}})$$

The Jacobian is defined by:

$$J_y(x_1, x_1, ..., x_6) = \begin{pmatrix} \frac{\partial y_1}{\partial x_1} & \cdots & \frac{\partial y_1}{\partial x_6} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_6}{\partial x_1} & \cdots & \frac{\partial y_6}{\partial x_6} \end{pmatrix} = \begin{pmatrix} J_{y11} & \cdots & J_{y16} \\ \vdots & \ddots & \vdots \\ J_{y61} & \cdots & J_{y66} \end{pmatrix}$$

where

$$J_{y11} = -\frac{x_1}{(x_1^{\,2} + x_2^{\,2} + x_3^{\,2})^{3/2}}$$

$$J_{y12} = -\frac{x_2}{(x_1^{\,2} + x_2^{\,2} + x_3^{\,2})^{3/2}}$$

$$J_{y13} = -\frac{x_3}{(x_1^{\,2} + x_2^{\,2} + x_3^{\,2})^{3/2}}$$

$$J_{y14} = J_{y15} = J_{y16} = 0$$

$$J_{y21} = -\frac{x_2}{x_1^{\,2} + x_2^{\,2}}$$

$$J_{y22} = \frac{x_1}{x_1^{\,2} + x_2^{\,2}}$$

$$J_{y23} = J_{y24} = J_{y25} = J_{y26} = 0$$

$$J_{y31} = \frac{x_3 x_1}{(x_1^{\,2} + x_2^{\,2} + x_3^{\,2})\sqrt{x_1^{\,2} + x_2^{\,2}}}$$

$$J_{y32} = \frac{x_3 x_2}{(x_1^{\,2} + x_2^{\,2} + x_3^{\,2})\sqrt{x_1^{\,2} + x_2^{\,2}}}$$

$$J_{y31} = -\frac{\sqrt{x_1^{\,2} + x_2^{\,2}}}{(x_1^{\,2} + x_2^{\,2} + x_3^{\,2})}$$

$$J_{y34} = J_{y35} = J_{y36} = 0$$

$$J_{y41}=\frac{x_4(x_1^{\,2}+x_2^{\,2}+x_3^{\,2})-2x_1(x_1x_4+x_2x_5+x_3x_6)}{(x_1^{\,2}+x_2^{\,2}+x_3^{\,2})^2}$$

$$J_{y42}=\frac{x_5(x_1^{\,2}+x_2^{\,2}+x_3^{\,2})-2x_2(x_1x_4+x_2x_5+x_3x_6)}{(x_1^{\,2}+x_2^{\,2}+x_3^{\,2})^2}$$

$$J_{y43}=\frac{x_6(x_1^{\,2}+x_2^{\,2}+x_3^{\,2})-2x_3(x_1x_4+x_2x_5+x_3x_6)}{(x_1^{\,2}+x_2^{\,2}+x_3^{\,2})^2}$$

$$J_{y44}=\frac{x_1}{x_1^{\,2}+x_2^{\,2}+x_3^{\,2}}$$

$$J_{y45}=\frac{x_2}{x_1^{\,2}+x_2^{\,2}+x_3^{\,2}}$$

$$J_{y46}=\frac{x_3}{x_1^{\,2}+x_2^{\,2}+x_3^{\,2}}$$

$$_{y51}=\frac{x_5(x_1^{\,2}+x_2^{\,2})-2x_1(x_1x_5-x_2x_4)}{(x_1^{\,2}+x_2^{\,2})^2}\cos(\arctan(\frac{-x_3}{\sqrt{x_1^{\,2}+x_2^{\,2}}}))-$$

$$\frac{x_1(x_1x_5-x_2x_4)}{(x_1^{\,2}+x_2^{\,2}+x_3^{\,2})(x_1^{\,2}+x_2^{\,2})^{3/2}}\sin(\arctan(\frac{-x_3}{\sqrt{x_1^{\,2}+x_2^{\,2}}}))$$

$$J_{y52}=\frac{-x_4(x_1^{\,2}+x_2^{\,2})+2x_2(x_1x_5-x_2x_4)}{(x_1^{\,2}+x_2^{\,2})^2}\cos(\arctan(\frac{-x_3}{\sqrt{x_1^{\,2}+x_2^{\,2}}}))-$$

$$\frac{x_2(x_1x_5-x_2x_4)}{(x_1^{\,2}+x_2^{\,2}+x_3^{\,2})(x_1^{\,2}+x_2^{\,2})^{3/2}}\sin(\arctan(\frac{-x_3}{\sqrt{x_1^{\,2}+x_2^{\,2}}}))$$

$$J_{y53}=\frac{(x_1x_5-x_2x_4)}{(x_1^{\,2}+x_2^{\,2}+x_3^{\,2})(x_1^{\,2}+x_2^{\,2})^{3/2}}\sin(\arctan(\frac{-x_3}{\sqrt{x_1^{\,2}+x_2^{\,2}}}))$$

$$J_{y54}=\frac{-x_2}{(x_1^{\,2}+x_2^{\,2})}\cos(\arctan(\frac{-x_3}{\sqrt{x_1^{\,2}+x_2^{\,2}}}))$$

$$J_{y55}=\frac{x_1}{(x_1^{\,2}+x_2^{\,2})}\cos(\arctan(\frac{-x_3}{\sqrt{x_1^{\,2}+x_2^{\,2}}}))$$

$$J_{y56}=0$$

$$J_{y61} = \frac{-2x_1x_6 + x_3x_4}{(x_1^2 + x_2^2 + x_3^2)\sqrt{x_1^2 + x_2^2}} - 2x_1 \frac{-x_6(x_1^2 + x_2^2) + x_3(x_1x_4 + x_2x_5)}{(x_1^2 + x_2^2 + x_3^2)^2\sqrt{x_1^2 + x_2^2}} -$$
$$x_1 \frac{-x_6(x_1^2 + x_2^2) + x_3(x_1x_4 + x_2x_5)}{(x_1^2 + x_2^2 + x_3^2)(x_1^2 + x_2^2)^{3/2}}$$

$$J_{y62} = \frac{-2x_2x_6 + x_3x_5}{(x_1^2 + x_2^2 + x_3^2)\sqrt{x_1^2 + x_2^2}} - 2x_2 \frac{-x_6(x_1^2 + x_2^2) + x_3(x_1x_4 + x_2x_5)}{(x_1^2 + x_2^2 + x_3^2)^2\sqrt{x_1^2 + x_2^2}} -$$
$$x_2 \frac{-x_6(x_1^2 + x_2^2) + x_3(x_1x_4 + x_2x_5)}{(x_1^2 + x_2^2 + x_3^2)(x_1^2 + x_2^2)^{3/2}}$$

$$J_{y63} = \frac{x_1x_4 + x_2x_5}{(x_1^2 + x_2^2 + x_3^2)\sqrt{x_1^2 + x_2^2}} - 2x_3 \frac{-x_6(x_1^2 + x_2^2) + x_3(x_1x_4 + x_2x_5)}{(x_1^2 + x_2^2 + x_3^2)^2\sqrt{x_1^2 + x_2^2}}$$

$$J_{y64} = \frac{x_1x_3}{(x_1^2 + x_2^2 + x_3^2)\sqrt{x_1^2 + x_2^2}}$$

$$J_{y65} = \frac{x_2x_3}{(x_1^2 + x_2^2 + x_3^2)\sqrt{x_1^2 + x_2^2}}$$

$$J_{y66} = \frac{\sqrt{x_1^2 + x_2^2}}{x_1^2 + x_2^2 + x_3^2}$$

B.2 Transformations from x^{car} to y^{msc}

$$x_1 = x = \frac{\cos(y_2)\cos(y_3)}{y_1}$$

$$x_2 = y = \frac{\sin(y_2)\cos(y_3)}{y_1}$$

$$x_3 = z = -\frac{\sin(y_3)}{y_1}$$

$$x_4 = \dot{x} = \frac{y_4\cos(y_2)\cos(y_3) - y_5\sin(y_2) - y_6\cos(y_2)\sin(y_3)}{y_1}$$

$$x_5 = \dot{y} = \frac{y_4\sin(y_2)\cos(y_3) - y_5\cos(y_2) - y_6\sin(y_2)\sin(y_3)}{y_1}$$

$$x_6 = \dot{z} = \frac{-y_4\sin(y_3) - y_6\cos(y_3)}{y_1}$$

The Jacobian is defined by

$$J_x(y_1, y_1, ..., y_6) = \begin{pmatrix} \frac{\partial x_1}{\partial y_1} & \cdots & \frac{\partial x_1}{\partial y_6} \\ \vdots & \ddots & \vdots \\ \frac{\partial x_6}{\partial y_1} & \cdots & \frac{\partial x_6}{\partial y_6} \end{pmatrix} = \begin{pmatrix} J_{x11} & \cdots & J_{x16} \\ \vdots & \ddots & \vdots \\ J_{x61} & \cdots & J_{x66} \end{pmatrix}$$

where

$$J_{x11} = -\frac{\cos(y_2)\cos(y_3)}{{y_1}^2}$$

$$J_{x12} = -\frac{\sin(y_2)\cos(y_3)}{y_1}$$

$$J_{x13} = -\frac{\cos(y_2)\sin(y_3)}{y_1}$$

$$J_{x14} = J_{x15} = J_{x16} = 0$$

$$J_{x21} = -\frac{\sin(y_2)\cos(y_3)}{{y_1}^2}$$

$$J_{x22} = \frac{\cos(y_2)\cos(y_3)}{y_1}$$

$$J_{x23} = -\frac{\sin(y_2)\sin(y_3)}{y_1}$$

$$J_{x24} = J_{x25} = J_{x26} = 0$$

$$J_{x31} = \frac{\sin(y_3)}{{y_1}^2}$$

$$J_{x32} = 0$$

$$J_{x33} = -\frac{\cos(y_3)}{y_1}$$

$$J_{x34} = J_{x35} = J_{x36} = 0$$

$$J_{x41} = \frac{-y_4 \cos(y_2)\cos(y_3) + y_5 \sin(y_2) + y_6 \cos(y_2)\sin(y_3)}{y_1^2}$$

$$J_{x42} = -\frac{y_4 \sin(y_2)\cos(y_3) + y_5 \cos(y_2) - y_6 \sin(y_2)\sin(y_3)}{y_1}$$

$$J_{x43} = -\frac{y_4 \cos(y_2)\sin(y_3) + y_6 \cos(y_2)\cos(y_3)}{y_1}$$

$$J_{x44} = \frac{\cos(y_2)\cos(y_3)}{y_1}$$

$$J_{x45} = -\frac{\sin(y_2)}{y_1}$$

$$J_{x46} = -\frac{\cos(y_2)\sin(y_3)}{y_1}$$

$$J_{x51} = \frac{-y_4 \sin(y_2)\cos(y_3) - y_5 \cos(y_2) + y_6 \sin(y_2)\sin(y_3)}{y_1^2}$$

$$J_{x52} = \frac{y_4 \cos(y_2)\cos(y_3) - y_5 \sin(y_2) - y_6 \cos(y_2)\sin(y_3)}{y_1}$$

$$J_{x53} = -\frac{y_4 \sin(y_2)\sin(y_3) + y_6 \sin(y_2)\cos(y_3)}{y_1}$$

$$J_{x54} = \frac{\sin(y_2)\cos(y_3)}{y_1}$$

$$J_{x55} = \frac{\cos(y_2)}{y_1}$$

$$J_{x56} = -\frac{\sin(y_2)\sin(y_3)}{y_1}$$

$$J_{x61} = \frac{y_4 \sin(y_3) + y_6 \cos(y_3)}{y_1^2}$$

$$J_{y62} = 0$$

$$J_{x63} = \frac{-y_4 \cos(y_3) + y_6 \sin(y_3)}{y_1}$$

$$J_{y64} = -\frac{\sin(y_3)}{y_1}$$

$$J_{x65} = 0$$

$$J_{y66} = -\frac{\cos(y_3)}{y_1}$$