SHORT RANGE THRUSTING PROJECTILE TRACKING

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

BY

OZAN ÖZGÜN BİLGİN

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

SEPTEMBER 2012

Approval of the thesis:

SHORT RANGE THRUSTING PROJECTILE TRACKING

submitted by **OZAN ÖZGÜN BİLGİN** 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. Mustafa Kuzuoğlu
Electrical and Electronics Engineering Dept., METU __________________

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

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

Asst. Prof. Dr. Umut Orguner
Electrical and Electronics Engineering Dept., METU __________________

Dr. Erdem Ertan
ASELSAN Inc. __________________

Date: 03.09.2012

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 : Ozan Özgün BİLGİN

Signature :

ABSTRACT

SHORT RANGE THRUSTING PROJECTILE TRACKING

Bilgin, Ozan Özgün

M. S., Department of Electrical and Electronics Engineering
Supervisor: Prof. Dr. Mübeccel Demirekler

September 2012, 74 pages

Short range thrusting projectiles are one of the various threats against armored vehicles and helicopters on the battlefield. Developing a countermeasure for this kind of projectiles is very crucial since they are vast in number and easy to operate on the battlefield. A countermeasure may consist of fire point prediction of the projectile and attack the launcher of it, or it may be the impact point prediction of the projectile and apply a hard-kill counter measure on its way to the ally target. For both of the countermeasure concepts, dynamics and parameters of the projectile must be estimated precisely. In this thesis, dynamic models for thrusting and ballistic flight modes of thrusting projectile are obtained. Three different tracking filters are suggested for precise tracking of the projectiles and their estimation performances are compared. These filters are the Extended Kalman Filter (EKF), the Particle Filter (PF) and the Marginalized Particle Filter (MPF).

Keywords: Ballistic, Target Tracking, Thrusting Projectile, Parameter Estimation, Marginalization, Kalman Filter, Particle Filter

ÖZ

KISA MENZİLLİ İTKİYE SAHİP MERMİLERİN TAKİBİ

Bilgin, Ozan Özgün

Yüksek Lisans, Elektrik Elektronik Mühendisliği Bölümü
Tez Yöneticisi : Prof. Dr. Mübeccel Demirekler

Eylül 2012, 74 sayfa

Kısa menzilli itkiye sahip mermiler zırhlı araçlara ve helikopterlere karşı tehdit oluşturan ve savaş alanında oldukça fazla bulunan tehditlerin başında gelmektedir. Savaş alanında çok fazla bulunması ve kolaylıkla dost unsurlara karşı kullanılabilir olması nedeniyle bu tip tehditlere karşı tedbir uygulanması oldukça önemlidir. Karşı tedbir yöntemleri olarak atım noktasının tahmini ile birlikte atıcıya karşı önlem alınması veya çarpışma noktasının hesaplanarak merminin havada imha edilmesi yöntemleri sıralanabilir. Bu tip karşı tedbir yöntemleri için merminin dinamiğinin iyi şekilde kestirilmesi gerekmektedir. Bu çalışmada, itkiye sahip ve balistik uçuş durumlarını içeren hedef dinamik modelleri elde edilmiştir. Daha sonra elde edilen hedef modelleri kullanılarak hedef üç farklı izleme süzgeci ile izlenmiş ve bu süzgeçlerin başarımları karşılaştırılmıştır. Gerçeklenen süzgeçler genişletilmiş Kalman süzgeci (EKF), parçacık süzgeci (PF) ve ayrıştırılmış parçacık süzgeci (MPF)'dir.

Anahtar Kelimeler: Balistik, Hedef İzleme, İtkiye Sahip Mermi, Parametre Kestirimi, Kalman Süzgeci, Parçacık Süzgeci

To My Family

ACKNOWLEDGEMENTS

I would like to express my sincere thanks and gratitude to my supervisor Prof. Dr. Mübeccel Demirekler for her complete guidance, advice and criticism throughout this study.

I am also thankful to all lecturers at the Department of Electrical and Electronics Engineering, who helped me to acquire the basic knowledge onto which I have built my thesis.

I would like to express my appreciation to ASELSAN Inc. for providing me a peaceful working environment and my colleagues in ASELSAN Inc. for their support.

Finally, I would like to express my special thanks to my family for their permanent support and sincere love.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Short range thrusting projectiles are one of the various threats against armored vehicles and helicopters on the battlefield. Their dynamics are highly nonlinear and they change modes between thrusting and ballistic through their flight. Therefore, the dynamics of the projectile cannot be defined by classical linear models such as, constant velocity, acceleration etc. The focus of this thesis is to derive dynamic model of the thrusting projectile and develop a precise tracking algorithm for the estimation of position and velocity of the projectile. In this chapter, first, information about short range thrusting projectiles is introduced. Then, the radar concept about the ballistic target tracking with radar concept will be described. These tracking algorithms are the Extended Kalman Filter (EKF), the Particle Filter (PF) and the Marginalized Particle Filter (MPF). In addition, survey of the literature on the tracking algorithms that are used for thrusting projectiles is reviewed. This chapter is completed with the outline of the thesis.

1.1. Short Range Thrusting Projectiles

Short range thrusting projectiles are generally man launched short range missiles. Their external flight can be considered in two phases as thrust and ballistic. They are fin-stabilized projectiles [1]. The most famous type of the man launched projectile is Rocket Propelled Grenade (RPG). The most famous RPG of all is the RPG-7 (Figure 1).

The rocket motor of the projectile is ignited after a short period it leaves the launcher. At the beginning of the flight it has an initial velocity which is referred as the nozzle velocity [2]. Before the ignition of rocket motor, the projectile is in the ballistic phase. Once the rocket motor is ignited, projectile enters the thrust phase. In this phase, the rocket motor applies thrust to the projectile generally for a brief period. When the rocket motor is depleted, in other words, when the thrust applied to the projectile is almost zero, the projectile enters ballistic phase again and flies in this phase until it impacts. The target dynamics and the dominant forces acting on the projectile are different in each phase. The launch sequence of the RPG-7 can be seen in Figure 2.

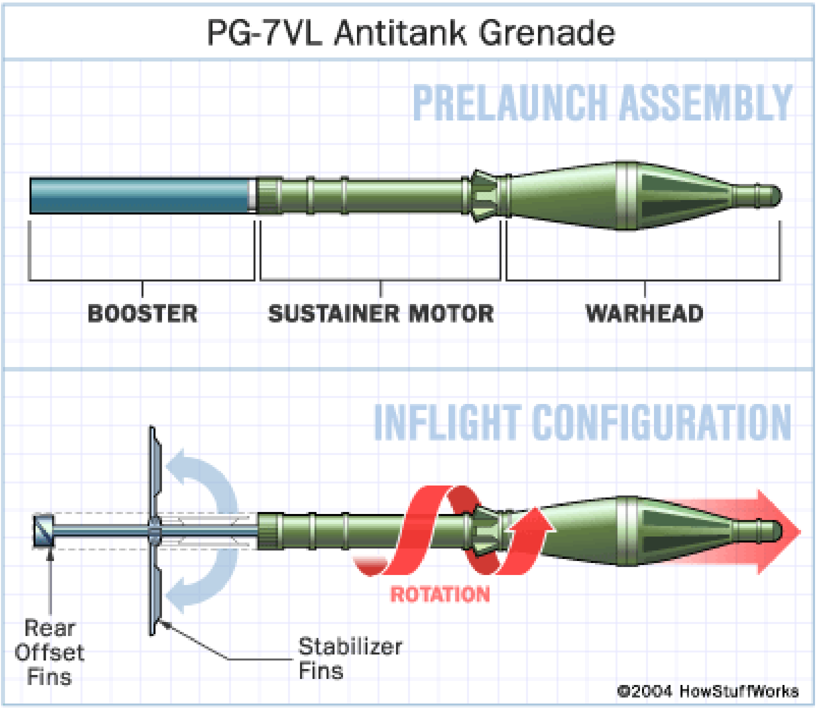

Figure 1: the RPG-7 Projectile [2]

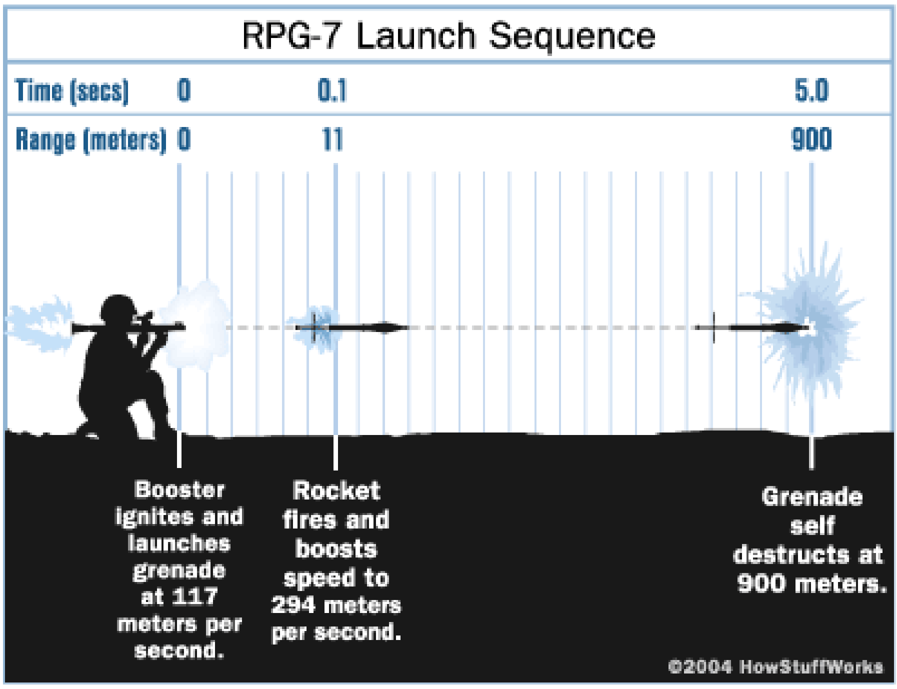
RPG-7 Launch Sequence
Time (secs)
0
0.1
5.0
Range (meters)
0
11
900
Booster ignites and launches grenade at 117 meters per second.
Rocket fires and boosts speed to 294 meters per second.
Grenade self destructs at 900 meters.
©2004 HowStuffWorks

Figure 2: the RPG-7 Launch Sequence [2]

Unfortunately, there is no detailed information about these kinds of projectiles in the literature and therefore, the ballistic parameters and the thrust profile are not known. For this reason, the projectile simulation in this thesis is performed for a hypothetical model with some dynamic model simplifications. Details are given in Chapter 5.

1.2. Ballistic Target Tracking with Radar

There is great interest in the literature for the tracking of the ballistic targets. The focus of these studies is mainly on the tracking of the long range and high altitude missile systems. The motivation of these studies is to enhance practical tracking applications for homeland security and missile defense systems. There are a vast number of studies that focused on the tracking of the missiles on their reentry phase. Reentry phase is the phase where missile is under endo-atmospheric effects like air lift and resistance. Two of the major papers on this subject surveys about the ballistic target tracking filters that are developed so far [3], [4]. In [3], the Extended Kalman Filter (EKF), the Unscented Kalman Filter (UKF) and the particle filter (PF) performances are compared. The focus of these studies are better modeling of the ballistic target and implementing tracking with the classical suboptimal Bayesian estimation algorithms.

There are also other studies on tracking ballistic missiles on their thrust phases[5],[6]. However, the first tracking algorithm for thrusting and ballistic mode switching target is developed by Yuan [7], [8]. In his study, Yuan focused on multiple model tracking of a short range thrusting projectile by Interacting Multiple Model EKF and predicting its impact point.

Bruno and Pavlov are focused on state estimation ballistic targets by the PF in their several studies [9], [10] and [11]. They assume a model with four states which favors the particle filter because it is well known that the particle filter performance decreases as the state dimension increases. More precisely, to achieve the same performance, the number of particles should be increased exponentially with the number of states.

There is an algorithm developed to improve estimation performances of particle filters for high state dimension models. This algorithm is called Marginalization (or sometimes it is called Rao-Blackwellization). The idea is to marginalize out the linear states from the system model, and estimate them with classical Kalman filter. The study on marginalization for different types of systems can be found in [12] and [13].

Our motivation is to develop a precise target tracking filter for thrusting short range projectiles. The filter should be a multiple model filter because of the thrusting and the ballistic modes of the projectile. With this motivation, a multiple model marginalized particle filter algorithm is developed and its estimation performance for different scenarios is compared with EKF and PF.

1.3. Outline of the Thesis

This thesis is composed of six chapters. The first chapter is introduction. In the second and third chapters, theoretical background required for this study is given. Fourth chapter provides detailed information about the tracking algorithms used in this study. The fifth chapter includes simulation description done in this study and the results of these simulations. Thesis is summarized in the conclusion chapter.

CHAPTER 2

RADAR THEORY

Radar stands for the Radio Detection and Ranging. Radars transmit electromagnetic radiation into the environment and receive returned reflections from objects in the environment. Radar radiation propagates at the speed of light. In this chapter, introductory information about radar fundamentals is given. More detailed information for interested readers can be found in [14] and [15].

2.1. Detection

The main purpose of the radar is to determine whether there is a target in the field of the view of the radar or not. Detection is based on transmitting electromagnetic waves and receiving its reflection from targets and clutter (Clutter is all other undesired objects other than the target.). In addition to the electromagnetic power received from the target and the clutter, there is also an additive white noise in the received signal which is caused by the thermal noise of the radar receiver.

In radar detection, there is a tradeoff between probability of detection (P_d) and probability of false alarm (P_{fa}). Typically, the P_{fa} is used to establish a detection threshold on the received signal. This threshold sets the P_d for a target based on its signal to interference ratio.

The detection process also depends on the radar cross section fluctuations and the statistical model employed to describe the fluctuating model. The fluctuating models are called Swerling models. There are four different Swerling models which are proposed for different distributions of the scatter points of the target and level of the correlation between measurements. A widely used model is Swerling – 1, in which radar cross section (RCS) value of the target is assumed to be correlated between the measurements for coherently integrated pulses. The relation between P$_{fa}$ and P$_d$ is as follows:

$$P_d \cong P_{fa}^{\left(\frac{1}{1+SNR}\right)} \tag{2-1}$$

The SNR value in (2-1) is in linear scale.

The detection threshold is calculated for the desired P$_{fa}$ by using the noise level. The most famous and optimal algorithm for the detection threshold calculation is Constant False Alarm Rate (CFAR). CA-CFAR (Cell Averaging CFAR) uses adaptive measurements of the noise in the neighboring cells of the Range – Doppler matrix and adaptively adjusts the detection threshold. CA-CFAR maintains the desired P$_{fa}$ in a noise varying environment. Range – Doppler matrix is obtained by applying matched filter to the complex envelope of the received filter and processing a bank of received pulses together [14].

2.2. Radar Range Equation

Radar range performance is measured by the signal to noise ratio of the echo received from the target. The signal to interference ratio on the receiver depends on the transmitter, receiver, and antenna characteristics of the radar as well as the target and the environment. Received power from the target is calculated as:

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

where,

P_t: Transmitted signal peak power [W]

G_t: Transmit antenna gain [Unitless]

G_r: Receive antenna gain [Unitless]

λ: Wavelength of the transmitted signal [m]

σ: RCS of the target [m^2]

L_s: System losses [Unitless]

L_a: Atmospheric Loss [Unitless]

R: Range of the target [m]

The received signal noise power is calculated as follows:

$$P_n = kTBN \tag{2-3}$$

where,

k: Boltzman Constant = $1.38x10^{-23}$ [J/K]

T: Standart Temperature [K]

B: Noise equivalent bandwidth of the receiver [Hz]

N: Noise figure of the receiver [Unitless]

The signal to noise ratio S/N is calculated using the ratio between the received signal power and the received noise power.

The maximum radar range can be then calculated with the minimum detectible signal $P_{r_{min}}$ and (2-2) as:

$$R_{max} = \left[\frac{P_t G_t G_r \lambda^2 \sigma}{(4\pi)^3 L_s L_a P_{r_{min}}}\right]^{1/4} \tag{2-4}$$

2.3. Measurements

In most applications, it is not enough to know the target is present or not. It is also necessary to know the target's location. Radar can measure target's distance (range) and direction (angle) with respect to the radar as follows:

2.3.1. Range Measurement

Range is determined by the measurement of the time that the radio waves take to reach the target and return. Radio waves travel at constant speed, the speed of light. Therefore, target's range is the speed of the light times half of the round trip time. Different kinds of waveforms can be used to apply different kind of ranging techniques. The most common radar waveform is a train of narrow rectangular-shaped pulses modulating the sine wave carrier.

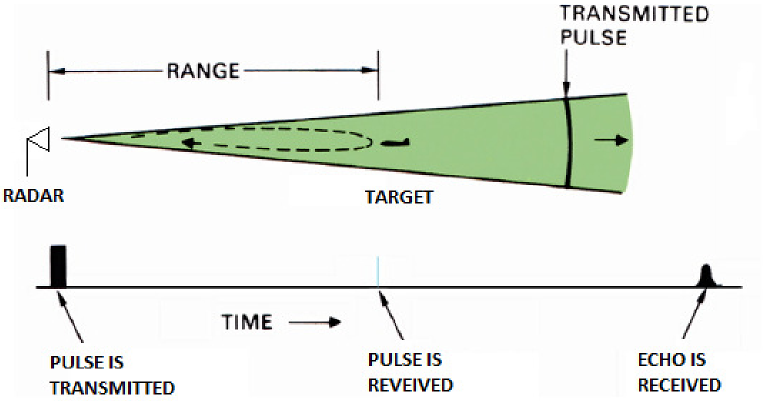

Figure 3: Pulsed Ranging [15]

The transit time is measured by observing the time delay T_R between the transmitted and received pulse.

$$R = \frac{c.T_R}{2} \tag{2-5}$$

For correct range measurement, echoes of the closely spaced targets should not be overlapped (after the pulse compression if a pulse compression technique is used). The bandwidth (BW) of the transmitted pulse determines the range resolution.

$$R_{res} = \frac{c}{2.BW} \tag{2-6}$$

Once radar transmits a pulse, it has to wait a sufficient time to transmit the next pulse before the echo of the pervious pulse returns to guarantee that there is no ambiguity occurs. Therefore, the maximum rate at which pulses may be transmitted is determined by the maximum unambiguous range desired.

$$R_{unamb} = \frac{c}{2.PRF} \tag{2-7}$$

PRF stands for the pulse repetition frequency.

The range measurement error probability density function directly depends on the measurement resolution and can be assumed uniform within the range resolution cell. There are some complicated techniques to determine the range within the range resolution cell by using the signal power levels from the cell under investigation and neighboring cells. After applying such techniques range error tends to the Gaussian distribution. The standard deviation of this error depends on both bandwidth and SNR [14].

$$\sigma_{Range\ Error} \cong \frac{c}{2B\sqrt{2.SNR}} \tag{2-8}$$

2.3.2. Doppler Measurement

The Doppler effect is a shift in the frequency of a wave reflected or received by an object in motion [15]. The wave radiated from a point source is compressed in the direction of motion and is spread out in the opposite direction. In other words, Doppler shift is the frequency difference between the transmitted and the received waves.

For every half wavelength per second that a target’s range decreases, the radial frequency phase of the received echo advances by the equivalent one whole cycle per second.

$$f_{Doppler} = \frac{-\dot{R}}{\lambda/2} \tag{2-9}$$

where, $\dot{R}$ is the radial component of the relative velocity between target and radar. The Doppler resolution frequency depends on the PRF and number of coherently processed pulses N. The Doppler frequency is obtained by slow time FFT. In another words, the FFT is taken between pulses on the same range bin. Therefore, the sampling frequency is PRF and FFT size is the number of samples. The radial velocity resolution of the radar is:

$$v_{res} = \frac{PRF}{N} \cdot \frac{\lambda}{2} \tag{2-10}$$

From the sampling theory, the maximum unambiguous Doppler frequency is determined by the sampling PRF. The maximum unambiguous radial velocity of the radar is:

$$v_{unamb} = \frac{PRF}{2} \cdot \frac{\lambda}{2} \tag{2-11}$$

The Doppler measurement error probability function directly depends on the Doppler resolution of the radar and the SNR.

$$\sigma_{Doppler\ Error} \cong \frac{PRF.\lambda}{2N\sqrt{2.SNR}} \tag{2-12}$$

By sensing Doppler frequency Radar does not only measures range rate directly but also differentiates the moving targets from stationary clutter [15]. This feature is called Moving Target Identification (MTI). MTI eliminates clutter by passing multiple coherent pulses through an MTI filter. MTI filter eliminates spectral regions with heavy clutter. For stationary radar, clutter is spread around zero Doppler frequency.

2.3.3. Direction Finding

Direction finding (DF) is a process to find the angular position of the target in the receive beam of the radar. In other words, DF is the angular error between boresight of the receive beam and the target [14]. There are different DF techniques. These techniques can be categorized in two as sequential lobing and monopulse techniques. In sequential lobing, two beams with different angle offsets are directed on the target sequentially in time (Figure 4). These beams are constructed with the same and only antenna aperture. The angle error is calculated by difference of the signal amplitude acquired by sequential beams. The drawback of the sequential lobing is that, it is affected by the signal amplitude fluctuations.

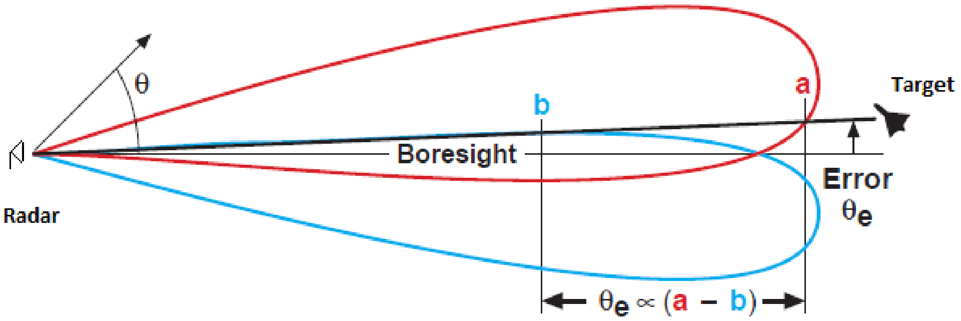

Figure 4: Amplitude Comparison Direction Finding [15]

The other method is the monopulse technique. In this technique, two simultaneous beams are produced by different halves of the antenna aparture to avoid inaccuracies due to pulse-to-pulse fluctuations in the received signal. By using simultaneous beams angle error is sensed in a single pulse. In amplitude comparison monopulse, overlappling beams are directed to the target and amplitude difference is used to calculate angle error (Figure 4). In another technique, both halves of the antenna produce beams pointing in the boresight direction and phase difference of the received signal is used to calculate angle difference.

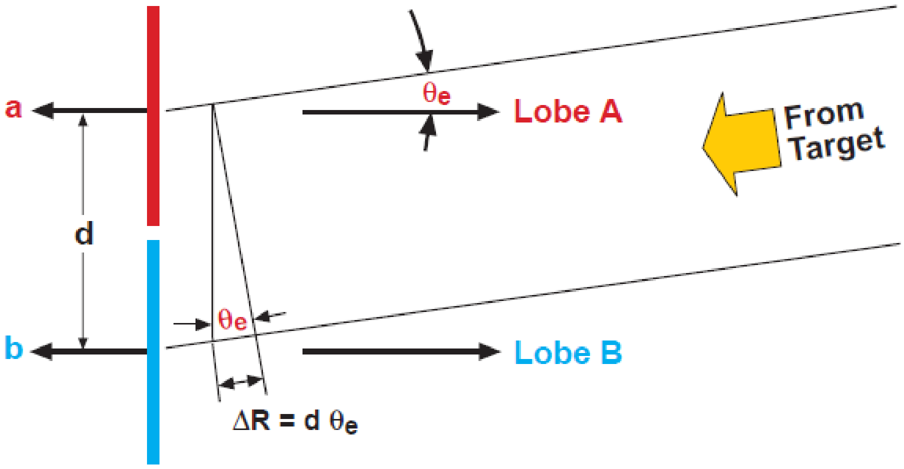

Figure 5: Phase Comparison Monopulse Direction Finding [15]

In phase comparison monopulse, the angle error θ_e is proportional to the phase difference between the signals acquired by the two different halves of the antenna.

In monopulse, the DF angle error standard deviation depends on the beam width of the receive beam and the SNR.

$$\sigma_{DF\ Error} \cong \frac{\theta_{BW}}{k_m\sqrt{2.SNR}} \tag{2-13}$$

where,

θ_{BW}: 3 dB beam width of the receive beam

k_m: slope of the amplitude ratio of sum and diff antenna patterns. It depends on the illumination tapering of the antenna.

CHAPTER 3

DYNAMICS OF BALLISTIC PROJECTILES

Our purpose in this thesis is to develop dynamic models for thrusting and ballistic modes of thrusting projectiles and develop a precise tracking algorithm. Unfortunately, there is not much detailed information about thrusting ballistic projectiles in the literature. Especially for the thrust profiles, there is no information at all. In this study, the modeling of a ballistic target few sources that exist in the literature are used. In this chapter, dynamics of the thrusting projectile for ballistic and thrust phases are explained. The equations in this chapter are mainly adopted from [16] and [17].

It is known from [1] that thrusting ballistic projectile are in the class of fin stabilized projectiles. This chapter constitutes mathematical models representing the flight of fin-stabilized projectiles which are dynamically stable and at least trigonometrically symmetric. The essential forces acting on ballistic projectiles are investigated.

3.1. Forces Acting on Fin Stabilized Projectiles

In this section dominant forces acting on the projectile are investigated. The forces are composed of, the Drag Force $(\overrightarrow{DF})$, the Lift Force $(\overrightarrow{LF})$, the Magnus Force $(\overrightarrow{MF})$, the Gravitational Force $(\vec{G})$ and the Thrust Force $(\vec{T})$. Newton's law of motion of the center of mass of the projectile is:

$$\vec{F} = m\vec{a} = \overrightarrow{DF} + \overrightarrow{LF} + \overrightarrow{MF} + \vec{G} + \vec{T} \tag{3-1}$$

These forces are explained in the following subsections. Since short range ballistic trajectories are our interest, effects of the Earth rotation are neglected (i.e. Coriolis force) when constructing equations of motion. Based on the same reasoning Earth is assumed to be flat.

3.1.1. Drag Force

The aerodynamic drag force opposes the forward velocity of the projectile, as illustrated in Figure 6. Drag is the first force to remember when the exterior ballistics is studied and generally referred as “air resistance” [16].

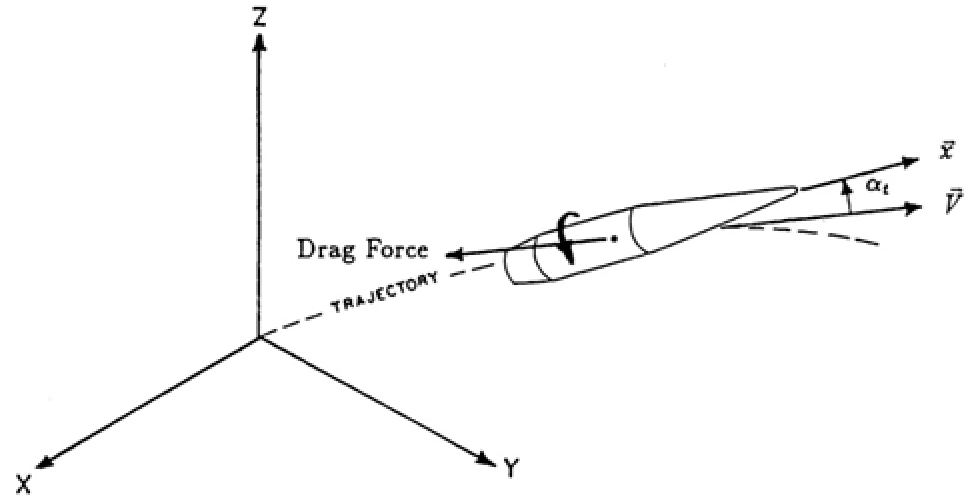

Figure 6: Drag Force Acting on the Projectile [16]

The vector representation of the drag force is as follows:

$$\overrightarrow{DF} = -\frac{1}{2}\,\rho S C_D \vec{V} V \tag{3-2}$$

where,

ρ = air density

$\vec{V}$ = velocity vector

V= the scalar magnitude of the velocity vector

S = projectile reference area

C_D= drag coefficient.

In practice the reference area S is defined as:

$$S = \frac{\pi d^2}{4} \tag{3-3}$$

where,

d = projectile diameter.

The diameter is usually taken as the diameter after the o-give nose section of the projectile. The effect of the yawing (or pitching since the projectile is assumed rotationally symmetric) is accounted in the drag coefficient C_D. This can be thought as if the effective reference area of the projectile is changed when the nose of the projectile is deflected with respect to the velocity vector $\vec{V}$. The drag coefficient C_D is well approximated by the following equation.

$$C_D = C_{D_0} + C_{D_{\delta^2}}\delta^2 \tag{3-4}$$

where,

C_{D_0}= the zeros-yaw drag coefficient

$C_{D_{\delta^2}}$= yaw drag coefficient

$\delta = sin\,(\alpha_t)$

α_t = the angle between the projectile nose direction and velocity vector

Note that as the yaw angle increases, drag coefficient and drag force are increasing in a quadratic manner. Drag force renders the projectile dynamics as highly nonlinear. In addition, as speed increases, drag force increases also in a quadratic manner and tends to slow down the projectile.

3.1.2. Lift Force

Aerodynamic lift is the force perpendicular to the projectile trajectory as it can be seen in Figure 7. If the nose of the projectile is above its trajectory, the projectile tends to climb because of the lift force. The lift force is the force which causes airplanes to fly.

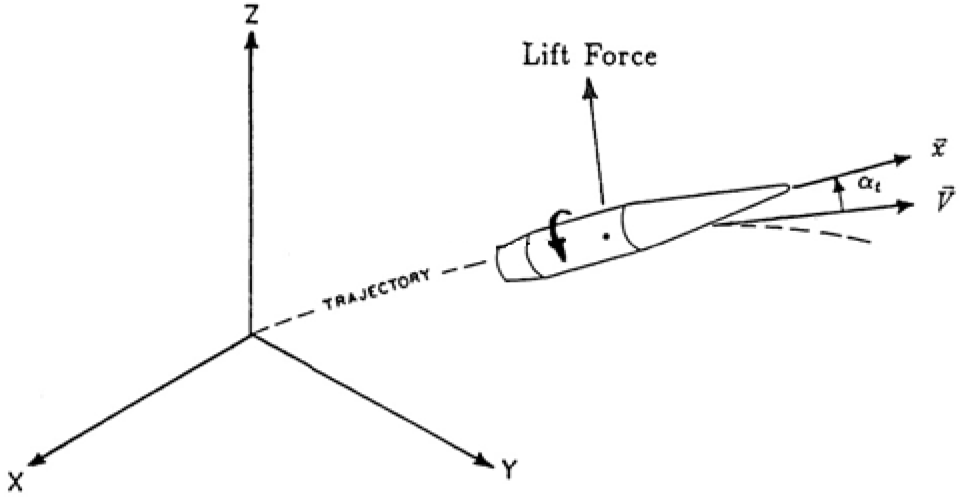

Figure 7: Lift Force Acting on the Projectile [16]

The vector form of the lift force is as follows:

$$\overrightarrow{LF} = -\frac{1}{2}\,\rho S C_{L_\alpha} \vec{V} \times (\vec{x} \times \vec{V}) \tag{3-5}$$

where,

$\vec{x}$= projectile nose direction

C_{L_α}= lift coefficient.

and other parameters are same as defined before.

The lift force coefficient exhibits nonlinear behavior because the lift coefficient varies with the yawing of the projectile.

$$C_{L_\alpha} = C_{L_{\alpha_0}} + \; C_{L_{\alpha_2}} \delta^2 \tag{3-6}$$

where,

$C_{L_{\alpha_0}}$= linear lift force coefficient

$C_{L_{\alpha_2}}$= cubic lift force coefficient

Note that as the yaw angle increases lift coefficient and the lift force increases in a cubic manner because of the cross product in (3-5). This is why the second coefficient in (3-6) is referred as cubic lift coefficient. Lift force is another factor which renders the projectile dynamics as nonlinear

3.1.3. Magnus Force

Magnus force is the phenomenon which occurred when a spinning object flying in a fluid creates a whirlpool around itself and experiences a force perpendicular to the line of motion. This force is caused because of the unequal pressures on the each side of the projectile which is caused by the viscous interaction between the fluid (air) and the spinning surface. Curved flight of the football is a demonstration of the Magnus force.

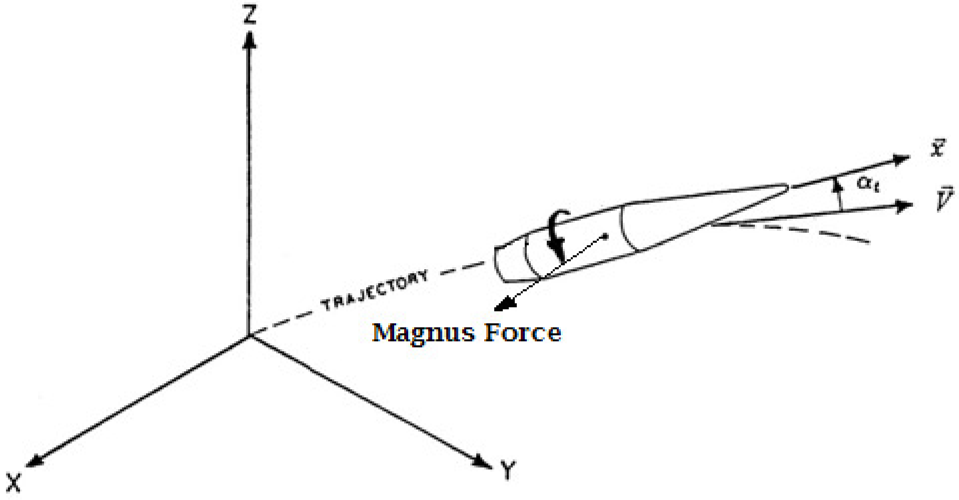

Figure 8: Magnus Force Acting on the Projectile [16]

The vector equation defining the Magnus force is as follows:

$$\overrightarrow{MF} = -\frac{1}{2}\,\rho V^2 S\left(\frac{pd}{V}\right) C_{N_{P\alpha}}\left(\frac{\vec{V}}{V} \times \vec{x}\right) \qquad (3\text{-}7)$$

where,

$C_{N_{P\alpha}}$= Magnus force coefficient

d=diameter of the projectile

p= axial spin (in radians per second, positive for right hand spin)

and other parameters are as defined before.

The Magnus force is proportional to the product of the spin and the sine of the yawing angle. The Magnus force coefficient is nonlinear in the yawing angle. The characteristics of this force are very similar with the lift force.

$$C_{N_{P\alpha}} = C_{N_{P\alpha_0}} + C_{N_{P\alpha_2}}\delta^2 \tag{3-8}$$

where,

$C_{N_{P\alpha_0}}$= linear lift force coefficient

$C_{N_{P\alpha_2}}$= cubic lift force coefficient

3.1.4. Gravitational Force

This force is nothing but the well known gravitational force caused by the gravitational acceleration. Acceleration due to gravity is calculated by [17]:

$$\vec{g} = g_0\left(\frac{Re}{Re + h}\right)^2 \tag{3-9}$$

$$g_0 = 9.80665[1 - 0.0026 * cos\ (2lat)] \tag{3-10}$$

where,

Re= mean radius of the Earth

h= height from sea level

lat= latitude in radians

For our applications height and latitude change through the flight of the projectile is negligible so that, the gravitational acceleration is assumed to be constant. Furthermore, Earth is assumed to be flat in this study.

3.1.5. **Thrust**

Thrust is the force induced by the rocket motor of the projectile. The thrust vector is on the direction of the nose of the projectile.

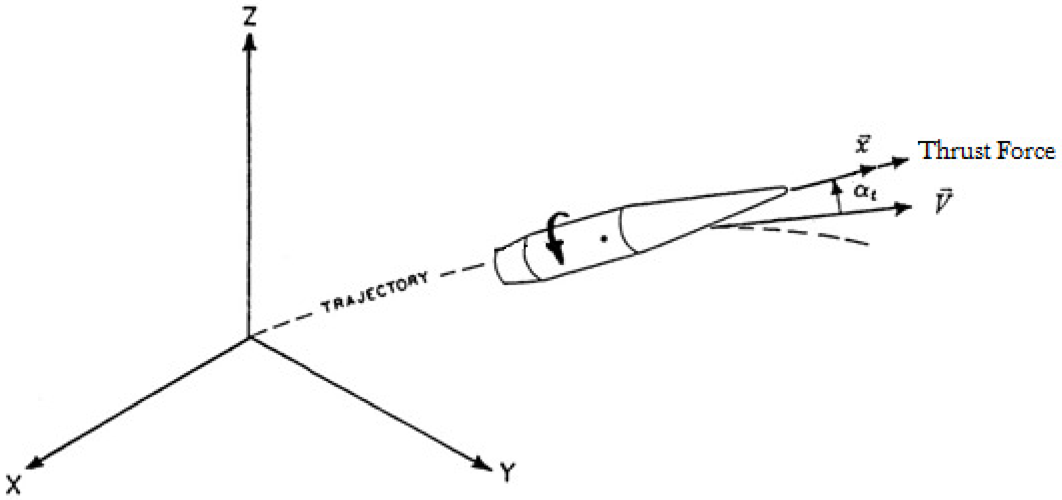

Figure 9: Thrust Acting on the Projectile [16]

The vector equation defining the Thrust is as follows:

$$\vec{T} = T(t)\vec{x} \tag{3-11}$$

Thrust is a time varying force because of the nature of the rocket motor of the projectile.

Thrust profile of a projectile is directly depended to the physical structure of the rocket motor. It is known from [1] that the type of projectile in our concern has solid rocket motors. The thrust profiles of these kinds of motors depend on the grain shape of the motor. The grain is the shape of the burning surface of the solid motor fuel [18], [19].

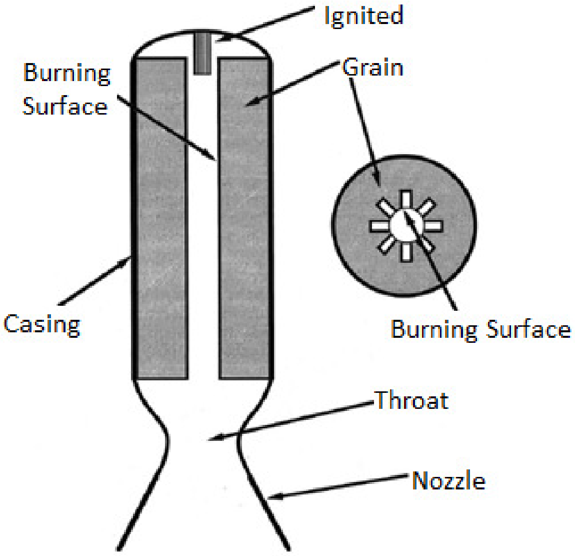

Figure 10: Schematic of a solid fuelled rocket motor [18]

The resultant force occurs on the rocket is directly depends to the rate of burn of the solid rocket fuel. From the Newton's second law of motion: "the resultant of the forces acting on an object is equal to the rate of change of the linear momentum of the object". Therefore the shape of the burning surface is the factor which designates the thrust profile.

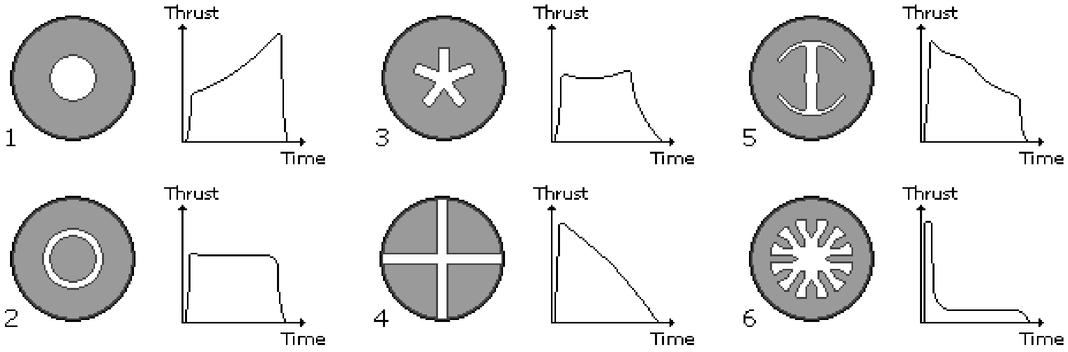

Figure 11: Thrust profiles for different kinds of grain shapes [19]

CHAPTER 4

TRACKING METHODOLOGY

In order to predict the trajectory of the thrusting projectiles, motion models for thrusting and ballistic phases should be constructed. After that, these models are used in multiple model Bayesian estimation algorithms in order to accomplish the tracking of the target. In this chapter, dynamic model of the thrusting ballistic projectile is constructed and the tracking algorithms that are used in simulations are defined.

4.1. Dynamic Model

The significant forces acting on a thrusting projectile are gravity, drag and thrust [7]. By ignoring other insignificant forces, dynamic models for both thrusting and ballistic modes of a ballistic target is obtained according to point mass motion equations. General expression of the dynamic model is as follows:

$$\dot{\boldsymbol{x}}(t) = f\big(\boldsymbol{x}(t)\big) + \tilde{w}(t) \tag{4-1}$$

Motion model is best defined in Cartesian coordinates. Therefore, in this study, we used Cartesian coordinates for the motion model of the target. The state consists of the positions and the velocities of the target in each axis as well as the drag ($\alpha(t)$) and thrust ($\tau(t)$) parameters [7].

$$\boldsymbol{x}(t) = [x(t)\ \dot{x}(t)\ y(t)\ \dot{y}(t)\ z(t)\ \dot{z}(t)\ \alpha(t)\ \tau(t)]^T \tag{4-2}$$

The dynamic model for thrusting ballistic projectile can be written as follows.

$$\begin{bmatrix}\ddot{x}(t)\\ \ddot{y}(t)\\ \ddot{z}(t)\end{bmatrix} = \frac{\tau(t)}{V(t)}\begin{bmatrix}\dot{x}(t)\\ \dot{y}(t)\\ \dot{z}(t)\end{bmatrix} + \alpha(t)D(t)\begin{bmatrix}\dot{x}(t)\\ \dot{y}(t)\\ \dot{z}(t)\end{bmatrix} + g\begin{bmatrix}0\\ 0\\ -1\end{bmatrix} + \widetilde{w}_d \qquad (4\text{-}3)$$

In (4-3), the acceleration for each axis is modeled by considering the effect of the dominant forces. In addition to that there is additive gravitational acceleration in the z axis.

The first term on the right side of the equation is the specific thrust in each of the three axes. $\tau(t)$ is the acceleration in m/s^2, $V(t)$ is the magnitude of the velocity of the projectile in m/s. For the free-flight mode effect of this acceleration is zero.

The second term is the acceleration caused by the drag force that affects the projectile in each three axes. This force is related to the velocity and the altitude of the projectile. $\alpha(t)$ is the drag coefficient in m^2/kg and D(t) is a parameter consists of velocity and air density.

$$D(t) = \frac{-\rho(z)V(t)}{2}$$
$$\rho(z) = \rho(0)e^{-cz} \qquad (4\text{-}4)$$

In (4-4), z is the altitude in meters and c is the air density constant in m^{-1}.

In addition to the deterministic coefficients, the dynamic uncertainties of the motion are modeled by a zero mean white Gaussian noise $\widetilde{w}_d$. Thrust and drag parameters are modeled as follows:

$$\alpha(t) = \alpha_0 + \widetilde{w}_\alpha(t) \qquad (4\text{-}5)$$

$$\tau(t) = \tau_0 + \widetilde{w}_\tau(t) \qquad (4\text{-}6)$$

$\widetilde{w}_\alpha$ and $\widetilde{w}_\tau$ are zero mean white Gaussian noises. Dynamic model process noise is as follows:

$$\widetilde{w} = \begin{bmatrix} \widetilde{w}_d(t) \\ \widetilde{w}_\alpha(t) \\ \widetilde{w}_\tau(t) \end{bmatrix} \tag{4-7}$$

Details and the model of the process noise is given in part 4.3.

4.2. Discrete Time Motion Model

The motion model given above is discretized since the Bayesian estimation is done in discrete time. One reason for it is that the measurement flow to the tracker system is discrete because of the nature of the measurement device (i.e. radar). Our goal is to obtain the following equation from (4-1).

$$\boldsymbol{x}[k+1] = f_d(\boldsymbol{x}[k]) + w_k \tag{4-8}$$

In the discretization procedure we first assume that the noise is zero and write the solution of the nonlinear differential equation recursively then add the corresponding noise term as a Gaussian noise with appropriate mean and covariance matrix.

There are several discretization methods like Euler method, Runge - Kutta methods etc. [20]. More complex methods are also developed all aim to improve the discretization performance, in other words to decrease the discretization error. More complex methods use higher order moments of the state. Performance of the discretization method depends on the discretization time period. Discretization performance increases as the time period decreases [20]. In this work, dynamic model is discretized by the Euler's forward method with a relatively short time period (1 ms).

Combining (4-3) through (4-7) the continuous time deterministic motion model (differential equation) is obtained as follows.

$$\dot{x} = f\big(\boldsymbol{x}(t)\big) = \begin{bmatrix} \dot{x}(t) \\ \frac{\tau(t)}{V(t)}\dot{x}(t) + \alpha(t)D(t)\dot{x}(t) \\ \dot{y}(t) \\ \frac{\tau(t)}{V(t)}\dot{y}(t) + \alpha(t)D(t)\dot{y}(t) \\ \dot{z}(t) \\ \frac{\tau(t)}{V(t)}\dot{z}(t) + \alpha(t)D(t)\dot{z}(t) - g \\ 0 \\ 0 \end{bmatrix} \tag{4-9}$$

In (4-9) continuous time state transition function can be seen. Discrete time state transition function is calculated by Euler's forward method [20] as follows.

$$\boldsymbol{x}[k+1] = f_d(\boldsymbol{x}[k]) \cong \boldsymbol{x}[k] + hf(\boldsymbol{x}[k]) \tag{4-10}$$

Where, $f(.)$ is defined in (4-9) and h is the discretization step size which is defined as track update period T from now on.

$$f_d(\boldsymbol{x}[k]) = \begin{bmatrix} x[k] + T\dot{x}[k] \\ \dot{x}[k]\left(1 + T\left(\frac{\tau[k]}{V[k]} + \alpha[k]D[k]\right)\right) \\ y(k) + T\dot{y}(k) \\ \dot{y}[k]\left(1 + T\left(\frac{\tau[k]}{V[k]} + \alpha[k]D[k]\right)\right) \\ z(k) + T\dot{z}(k) \\ \dot{z}[k]\left(1 + T\left(\frac{\tau[k]}{V[k]} + \alpha[k]D[k]\right)\right) - gT \\ \alpha[k] \\ \tau[k] \end{bmatrix} \tag{4-11}$$

More accurate discretization can be done by taking higher order moments of $f(.)$ into consideration. But, this level is sufficient for this work, since the quantization period is taken fairly small.

4.3. Process Noise Model

Process noise in (4-8) is acting on the states that represent motion and ballistic parameters. The process noise of the motion states are modeled in continuous time (CWNA) [7]. Then it is discretized. The advantage of this model is that it is independent to the discretization period compared to the direct discrete time modeling (DWNA) [7]. The covariance matrix for process noise Q for the states defined in (4-2) is as follows.

$$Q = \begin{bmatrix} \bar{Q}_v & 0 & 0 \\ 0 & Tq_\alpha & 0 \\ 0 & 0 & Tq_\tau \end{bmatrix} \tag{4-12}$$

$$\bar{Q}_v = \begin{bmatrix} Q_v & 0 & 0 \\ 0 & Q_v & 0 \\ 0 & 0 & Q_v \end{bmatrix} \tag{4-13}$$

$$Q_v = \begin{bmatrix} T^3/3 & T^2/2 \\ T^2/2 & T \end{bmatrix} q_v \tag{4-14}$$

q_v, q_α and q_τ are the corresponding power spectral densities (PSD) of process noises.

$$q_v = {d_v}^2 T \tag{4-15}$$

$$q_\alpha = {d_\alpha}^2 T \tag{4-16}$$

$$q_\tau = {d_\tau}^2 T \tag{4-17}$$

Where, d_v (m/s^2), $d_\alpha(m^2/(kg.s))$ and $d_\tau(m/s^3)$ are the root mean square (rms) rate of change in velocity, drag coefficient rate of change and thrust coefficient rate of change correspondingly[7].

4.4. Separation of the Linear and Nonlinear Parts

Performances of the three filters (EKF, PF and MPF) are compared in this study. All of them use an IMM approach with two models which correspond to the thrust and ballistic parts of the trajectory. In section 4.6, details of the implementation are given. The application of the marginalized particle filter requires the separation of the linear and the nonlinear parts of the model. At this point, we give an approximate equivalent of the state space equations that is separated into the linear and the nonlinear subsystems. Marginalization reduces the dimension of the nonlinear system so that the model can be utilized to obtain better estimates of the state using the same number of particles as that of standard particle filter. [12]- [13] The idea in the marginalized particle filter is to marginalize out the linear states using Bayes' theorem and filter them with standard Kalman filter which is optimal for linear and Gaussian estimation problems. This technique is sometimes referred as Rao-Blackwellization. Below, a brief explanation for the marginalized particle filter is given.

Let the state vector be partitioned as follows:

$$\boldsymbol{x}_k = \begin{bmatrix} \boldsymbol{x}_k^l \\ \boldsymbol{x}_k^n \end{bmatrix} \tag{4-18}$$

Where $\boldsymbol{x}_k^l$ denotes the states with conditionally linear dynamics and $\boldsymbol{x}_k^n$ denotes the states with nonlinear dynamics. The goal is to obtain a system model in the following form with the selected linear and nonlinear states.

$$\boldsymbol{x}_{k+1}^l = A_1^l(\boldsymbol{x}_k^n) + A_2^l(\boldsymbol{x}_k^n)\boldsymbol{x}_k^l + w_k^l \tag{4-19}$$

$$\boldsymbol{x}_{k+1}^n = A_1^n(\boldsymbol{x}_k^n) + A_2^n(\boldsymbol{x}_k^n)\boldsymbol{x}_k^l + w_k^n \tag{4-20}$$

$$y_k = C(\boldsymbol{x}_k^n)\boldsymbol{x}_k^l + v_k \tag{4-21}$$

System model is given in (4-11). We select the velocities as the nonlinear part of the state. A close investigation of (4-11) shows that this choice seems to be inappropriate because of the altitude $z(k)$ term in $D(k)$. However, thrusting projectiles can be assumed flying less than 100 m altitude always. This is especially true for man-launched projectiles.

As it can be seen from Figure 12, air density changes slightly for the altitude range from 0 m to 100. This means Drag force would change slightly in this altitude range. To illustrate, let a projectile has S x $C_d = 10^{-4}$ m^2, m = 2 kg and V = 150 m/s, the difference in the drag force that affects the projectile between 0 m and 100 m altitude is 0.01 N. This shows that, it is possible to assume that the air density is constant during the flight. Hence, the system model can be marginalized. This assumption is reasonable because such an uncertainty in air density always exists due to the changes in the humidity and the temperature of the air density.

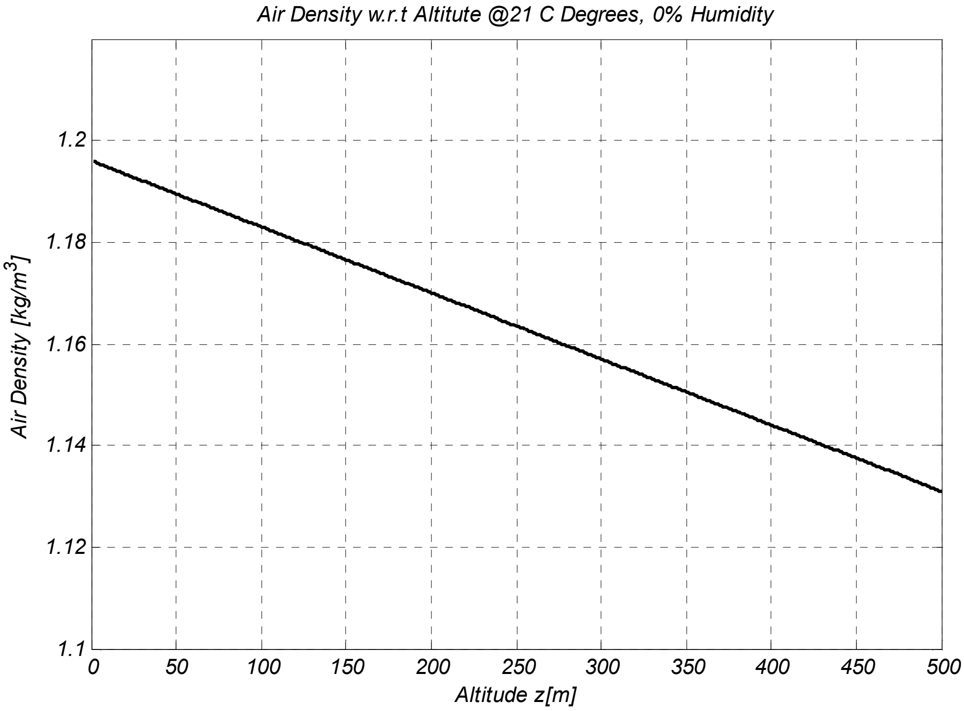

Figure 12: Air Density w.r.t. Altitude

A close look at (4-11) shows that the drag and the thrust parameter as well as position states can be taken as linear states.

$$\boldsymbol{x}_k^l = \begin{bmatrix} x[k] \\ y[k] \\ z[k] \\ \alpha[k] \\ \tau[k] \end{bmatrix} \tag{4-22}$$

$$\boldsymbol{x}_k^n = \begin{bmatrix} \dot{x}[k] \\ \dot{y}[k] \\ \dot{z}[k] \end{bmatrix} \tag{4-23}$$

From (4-11), (4-22) and (4-23) the linear state transition model can be written in the following form.

$$\boldsymbol{x}_{k+1|k}^l = \begin{bmatrix} T & 0 & 0 \\ 0 & T & 0 \\ 0 & 0 & T \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \boldsymbol{x}_k^n + \begin{bmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{bmatrix}^{5x5} \boldsymbol{x}_k^l + w_k^l \tag{4-24}$$

Hence, the linear state transition model can be expressed as in (4-19). $A_1^l(\boldsymbol{x}_k^n)$ and $A_2^l(\boldsymbol{x}_k^n)$ are the first and second terms on the right hand side of (4-24) respectively. More than that, these functions are in special forms. A_1^l is a linear function of the nonlinear states so it can be expressed as a matrix coefficient and A_2^l is independent of $\boldsymbol{x}_k^n$ and its 5x5 identity matrix.

The nonlinear state transition model can be put in the following form.

$$\begin{gathered} \boldsymbol{x}_{k+1|k}^n = \\ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \boldsymbol{x}_k^n - \begin{bmatrix} 0 \\ 0 \\ -gT \end{bmatrix} + \\ \begin{bmatrix} 0 & 0 & 0 & \frac{T}{V[k]}\dot{x}[k] & D[k]\dot{x}[k] \\ 0 & 0 & 0 & \frac{T}{V[k]}\dot{y}[k] & D[k]\dot{y}[k] \\ 0 & 0 & 0 & \frac{T}{V[k]}\dot{z}[k] & D[k]\dot{z}[k] \end{bmatrix}^{3x5} \boldsymbol{x}_k^l + w_k^n \end{gathered} \tag{4-25}$$

The nonlinear states can also be expressed in the desired form in (4-20). $A_1^n(\boldsymbol{x}_k^n)$ is the sum of the first two expressions on the right hand side of (4-25). It's a nonlinear (affine) function of $\boldsymbol{x}_k^n$. $A_2^n(\boldsymbol{x}_k^n)$ is the third expression on the right hand side.

As it can be seen from (4-24) and (4-25), the state noise is also partitioned into linear and nonlinear structures. The state noise is assumed white and Gaussian where density is defined as follows.

$$w_k = \begin{bmatrix} w_k^l \\ w_k^n \end{bmatrix} \sim N(0, Q_k) \tag{4-26}$$

$$Q_k = \begin{bmatrix} Q_k^l & Q_k^{ln} \\ \left(Q_k^{ln}\right)^T & Q_k^n \end{bmatrix} \tag{4-27}$$

After these steps, our model is considered as marginalizable. The expressions are used in the marginalized particle filter algorithm are given in Section 4.8.3.

4.5. Measurement Model

The measurement unit is assumed to be a three dimensional radar that produces measurements in the three axes in spherical coordinates so the measurements are nonlinear functions of the state variables. Because of this the measurement model of the radar is nonlinear.

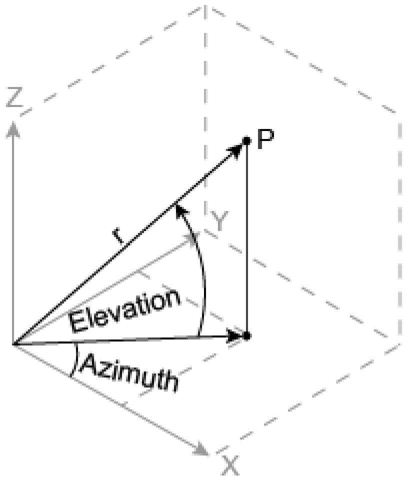

Figure 13: Spherical Coordinate System Definition

The measurement vector consists of azimuth angle (θ), elevation angle (φ) and range (R) respectively. v_k is the measurement noise and is independent of the process noise and white Gaussian.

$$y_k = \begin{bmatrix} \theta_m \\ \varphi_m \\ r_m \end{bmatrix} = \begin{bmatrix} tan^{-1}(y/x) \\ tan^{-1}(z/\sqrt{x^2+y^2}) \\ \sqrt{x^2+y^2+z^2} \end{bmatrix} + \begin{bmatrix} v_\theta \\ v_\varphi \\ v_r \end{bmatrix} \tag{4-28}$$

$$v_\theta \sim N(0, \sigma_\theta^2)$$

$$v_\varphi \sim N\left(0, \sigma_\varphi^2\right) \tag{4-29}$$

$$v_r \sim N(0, \sigma_r^2)$$

4.6. Converted Measurements

In this work, the goal is to track a ballistic target with radar. This means both the process and measurement models are nonlinear. For measurement systems of high accuracy, nonlinearity in measurement model can be handled by unbiased conversion [8]. In this study, converted measurements are used for measurement update in Tracking Filters [21].

Assuming the spherical measurement noises are Gaussian, the unbiased converted measurements are as follows:

$$
\begin{aligned}
x_m^u &= b_1^{-1} b_2^{-1} r_m cos\,(\theta_m) cos\,(\varphi_m) \\
y_m^u &= b_1^{-1} b_2^{-1} r_m sin\,(\theta_m) cos\,(\varphi_m) \\
z_m^u &= b_1^{-1} r_m sin\,(\varphi_m)
\end{aligned}
\qquad (4\text{-}30)
$$

where,

$$
\begin{aligned}
b_1 &= e^{-\sigma_\theta^2/2} \\
b_2 &= e^{-\sigma_\varphi^2/2}
\end{aligned}
\qquad (4\text{-}31)
$$

Note that the spherical measurements are defined in (4-28) and (4-29). Since the measurements are assumed as Gaussians, their covariance matrix should be transformed to the Cartesian coordinate system too. Let R_m^u be a 3x3 covariance matrix of the unbiased converted measurements. Elements of this matrix are as follows:

$$
\begin{aligned}
R_m^u(1{,}1) &= [(b_1 b_2)^{-2} - 2] {r_m}^2 cos^2\theta_m cos^2\varphi_m \\
&+ \tfrac{1}{4}(r_m^2 + \sigma_r^2)(1 - b_1^4 cos2\theta_m)(1 + b_2^4 cos2\varphi_m)
\end{aligned}
\qquad (4\text{-}32)
$$

$$
\begin{aligned}
R_m^u(2{,}2) &= [(b_1 b_2)^{-2} - 2] {r_m}^2 sin^2\theta_m cos^2\varphi_m \\
&+ \tfrac{1}{4}(r_m^2 + \sigma_r^2)(1 - b_1^4 cos2\theta_m)(1 + b_2^4 cos2\varphi_m)
\end{aligned}
\qquad (4\text{-}33)
$$

$$
\begin{aligned}
R_m^u(3{,}3) &= [(b_2)^{-2} - 2] {r_m}^2 sin^2\varphi_m \\
&+ \tfrac{1}{2}(r_m^2 + \sigma_r^2)(1 + b_2^4 cos2\varphi_m)
\end{aligned}
\qquad (4\text{-}34)
$$

$$
\begin{aligned}
R_m^u(1{,}2) &= [(b_1 b_2)^{-2} - 2] {r_m}^2 sin\theta_m cos\theta_m cos^2\varphi_m \\
&+ \tfrac{1}{4}(r_m^2 + \sigma_r^2) b_1^4 cos2\theta_m (1 + b_2^4 cos2\varphi_m)
\end{aligned}
\qquad (4\text{-}35)
$$

$$R_m^u(1,3) = [b_1^{-1}b_2^{-1} - b_1^{-1} - b_1]{r_m}^2 cos\theta_m cos\varphi_m sin\varphi_m + \frac{1}{2}(r_m^2 + \sigma_r^2)b_1 b_2^4 cos\theta_m sin2\varphi_m \tag{4-36}$$

$$R_m^u(2,3) = [b_1^{-1}b_2^{-1} - b_1^{-1} - b_1]{r_m}^2 sin\theta_m cos\varphi_m sin\varphi_m + \frac{1}{2}(r_m^2 + \sigma_r^2)b_1 b_2^4 sin\theta_m sin2\varphi_m \tag{4-37}$$

Rest of the elements of the covariance matrix R_m^u is zero.

According to [21], unbiased conversion is valid when the following inequalities stand.

$$\frac{r_m\ \sigma_\theta^2}{\sigma_r} < 0.4 \tag{4-38}$$

$$\sigma_\theta < 0.4rad \approx 23^o \tag{4-39}$$

$$\frac{r_m\ \sigma_\varphi^2}{\sigma_r} < 0.4$$

$$\sigma_\varphi < 0.4rad \approx 23^o$$

All of these inequalities are satisfied in this study.

4.7. State Initialization

Since projectiles that are in concern have very short time of flights, their tracks should be in steady state in a short time. In other words, the transition times of tracks should be short. For this purpose initial state estimation should be done precisely. In the literature, velocity initialization of targets is generally done by converting two measurements in Cartesian coordinates and differentiating them [7]. But for our application this kind of initialization leads to poor general estimation performances. The solution that we used is given in [7] as N-point initialization. The initialization is done by polynomial fitting to N unbiased converted measurements. Second order polynomial fitting and line fitting is tested and it is observed that line fitting lead better results. Polynomial fitting is done as follows:

Let the kth measurement be modeled as follows:

$$y_p(k) = h(k)a_n + v_k \tag{4-40}$$

where,

$$h(k) = \left[1\; t_k \dots \frac{{t_k}^n}{n!}\right] \tag{4-41}$$

$$a_n = [a_0\; a_1 \dots\; a_n] \tag{4-42}$$

Position and velocities in each three axes are initialized by polynomial fitting. “n” denotes the order of the polynomial. Fitting is done by least squares method. For the linear case (i.e. n = 1), the solution is as follows:

$$a_1 = \left[\sum_{k=1}^{N} h(k)h(k)^T\right]^{-1} \sum_{k=1}^{N} h(k)y_p(k) \tag{4-43}$$

And the covariance of the polynomial coefficients are calculated as follows:

$$P_p = \left[\sum_{k=1}^{N} \frac{h(k)h(k)^T}{\sigma_p^2(k)}\right]^{-1} \tag{4-44}$$

From the calculated polynomial coefficients and their covariance matrix mean and covariance of the position and velocity of the target can be found. The drag coefficient and the thrust coefficient are initialized by certain values. The drag and thrust accelerations are coupled. In another words, they are affecting the projectile on the same axis. Therefore, it is not possible to estimate both of them correctly. In this study, drag coefficient is assumed to be known (initialized with real value and small variance) and thrust coefficient is estimated (initialized with a high random value and large variance).

4.8. Tracking Filters

4.8.1. Extended Kalman Filter

The extended Kalman filter is a very widely used state estimation algorithm for nonlinear but Gaussian systems. The filtering is done by the linearization (a series expansion) of the nonlinearities in the dynamic system. In the Extended Kalman Filter, the state transition and measurement models need not be linear functions of the state but must be differentiable. Assume that at time $k,$ the state of the system is estimated as $\boldsymbol{x}_{k|k}$ with the corresponding covariance matrix $P_{k|k}$. The prediction equations are simply obtained using the nonlinearities as follows.

$$\boldsymbol{x}_{k+1|k} = f_d(\boldsymbol{x}_{k|k}) \tag{4-45}$$

$$\boldsymbol{y}_{k+1|k} = h_d(\boldsymbol{x}_{k+1|k}) \tag{4-46}$$

The state covariance matrix prediction is done by means of Kalman filter equations using the Jacobian matrices of the nonlinearity of the state and the observation.

$$F_k = \left.\frac{\partial f_d}{\partial x}\right|_{x_{k|k}} \tag{4-47}$$

and

$$H_{k+1} = \left.\frac{\partial h_d}{\partial x}\right|_{x_{k+1|k}} \tag{4-48}$$

There is obligation to use unbiased converted measurements in marginalized particle filter. In order to make a fair comparison between these filters unbiased converted measurements are used in EKF too. As a result, nonlinear function in (4-46) becomes linear and the measurement update is done as in the classical Kalman filter.

The pseudo code of the extended Kalman filter algorithm is as follows:

Table 1: Extended Kalman Filter

```
1) Initialization:
   Set k →1
   Set the initial estimate: $\boldsymbol{x}_{k|k} = \boldsymbol{x}_0$, $\boldsymbol{P}_{k|k} = \boldsymbol{P}_0$
2) Prediction:
   Calculation of Process Model Jacobian:
   $F_k = \frac{\partial f_d}{\partial x}\Big|_{x_{k|k}}$
   State Mean Prediction: $\boldsymbol{x}_{k+1|k} = f_d(\boldsymbol{x}_{k|k})$
   State Covariance Prediction:$P_{k+1|k} = F_k P_{k|k} {F_k}^T + Q$
3) Measurement Update
   Calculation of Measurement Model Jacobian:
   $H_{k+1} = \frac{\partial f_d}{\partial x}\Big|_{x_{k+|k}}$
   Innovation Mean: $z_{k+1} = \boldsymbol{y}_{k+1} - h_d(\boldsymbol{x}_{k+1|k})$
   Innovation Cov:  $S_{k+1} = H_{k+1} P_{k+1|k} {H_{k+1}}^T + R$
   Kalman Gain: $K_{k+1} = P_{k+1|k} {H_{k+1}}^T {S_{k+1}}^{-1}$
   State Mean Estimate: $\boldsymbol{x}_{k+1|k+1} = \boldsymbol{x}_{k+1|k} + K_{k+1} z_{k+1}$
   State Cov. Estimate: $P_{k+1|k+1} = P_{k+1|k} - K_{k+1} H_{k+1} P_{k+1|k}$
4) k = k +1
5) Go to 2
```

4.8.2. Particle Filter

Particle filter (PF) is a Sequential Monte Carlo method to solve the Bayesian state estimation problem for the nonlinear and the non-Gaussian systems. Goal of the Bayesian approach is to construct the pdf of the state at the current time k by using all the information (measurements) available up to the current time. For our problem, the state pdf in (4-49) remains Gaussian in every time step.

$$\boldsymbol{x}_{k|k} \sim p(\boldsymbol{x}_k|\boldsymbol{Y}_k) \tag{4-49}$$

In (4-49), $\boldsymbol{x}_{k|k}$ denotes the posterior estimate of the state vector and $\boldsymbol{Y}_k = \{\boldsymbol{y}_i\}_{i=1}^{k}$ denotes all the available measurements up to the time k.

The idea of particle filtering is to represent the state pdf with random particles. As the number of particles approaches to infinity particle filter represents the pdf of the state perfectly and becomes optimal.

In this study, Sequential Importance Resampling (SIR) algorithm is used. This algorithm is selected because it is a solution to the degeneracy problem which will be explained shortly [22].

4.8.2.1. Sequential Importance Resampling

Sequential Importance Sampling (SIR) algorithm is one of the most famous PF algorithms and also called “bootstrap” filter [22]. The SIR algorithm defines the importance density (the density that the samples are drawn for the next time interval and also the one where weight update calculation is applied to) as the transitional prior density. Posterior densities acquired from other Bayesian estimation filters like (KF, EKF) can also be used as importance densities.

SIR is a simple and easy way to implement the PF. The pseudo code of the algorithm is given as follows.

Table 2: SIR Algorithm

```
1) Initialization:
   FOR i = 1:N
       Draw samples from initial importance density: $\boldsymbol{x}_{0|-1}^{i} \sim p_{x_0}(x_0)$
   END FOR
   Set k →0
2) Measurement Update:
   FOR i = 1:N
       Calculation of weights: $w_k^{(i)} = p\big(y_k \big| \boldsymbol{x}_{k|k-1}^{i}\big)$
   END FOR
   Normalization of weights: $w_k^{(i)} = \frac{w_k^{(i)}}{\sum_{j=1}^{N} w_k^{(j)}}$
   To conserve the notation: : $\boldsymbol{x}_{k|k} = \boldsymbol{x}_{k|k-1}$
3) Resampling
   [$\boldsymbol{x}_{k|k}, w_k$]=Resample($\boldsymbol{x}_{k|k}, w_k$)
4) PF: Time Update:
   FOR i = 1:N
       Random number draw: $\overline{w}_k^{(i)} \sim N(0, Q)$
       State Prediction: $\boldsymbol{x}_{k+1|k}^{(i)} = A_d(\boldsymbol{x}_{k|k}^{(i)}) + \overline{w}_k^{n,(i)}$
   END FOR
5) k = k +1
6) Go to 2
```

The details of the resample function on the resampling stage are given in 4.8.4.

4.8.3. Marginalized Particle Filter

Implementation of the particle filter for nonlinear (even for non-Gaussian) filtering problems is easy. But, the main drawback of particle filter is its exponentially increasing computational complexity with the state dimension. The tracking problem focused on this thesis has eight dimensional space which can be considered very high for particle filtering.

A solution to the complexity problem is to marginalize out the states appearing linearly in the dynamics of the system [12]- [13]. Marginalized particle filter solves the linear part of the system by using a Kalman filter for each particle of the nonlinear part. In another words, “The Marginalized Particle Filter (MPF) is a clever combination of the standard particle filter and the Kalman Filter” [13].

As in all Bayesian estimation algorithms, our aim is to estimate the posterior state density p($\boldsymbol{x}_k|\boldsymbol{Y}_k$) where, $\boldsymbol{Y}_k = \{\boldsymbol{y}_i\}_{i=1}^{k}$ is the measurements up to time k. By employing Bayesian Theory and partitioning the states as linear and nonlinear the estimation of the state variables can be obtained as in (4-50).

$$p\left(\boldsymbol{x}_k^l, \boldsymbol{X}_k^n | \boldsymbol{Y}_k\right) = \underbrace{p\left(\boldsymbol{x}_k^l | \boldsymbol{X}_k^n, \boldsymbol{Y}_k\right)}_{KF} \underbrace{p(\boldsymbol{X}_k^n | \boldsymbol{Y}_k)}_{PF} \qquad (4\text{-}50)$$

The first expression on the right hand side of (4-50) is analytically tractable since all nonlinear states $\boldsymbol{X}_k^n = \{\boldsymbol{x}_i^n\}_{i=1}^{k}$ up to time k are given. Since the underlying model for the linear states is linear and linear states are considered as Gaussian distributed the linear states estimation can be done with the classical Kalman filter which is the optimal solution for linear – Gaussian case. The estimate of the second expression on the right hand side of (4-50) is provided by the particle filter.

Inspecting (4-19) - (4-21), we see that the linear state variables are dependent to the nonlinear state variables and vice versa. There should be a nested estimation algorithm composed of Kalman filter and particle filter.

Let’s start the derivation of the MPF structure with (4-20). As it can be seen, the linear states are directly added to the nonlinear states with a coefficient.

$$\boldsymbol{x}_{k+1}^{n} = A_1^n(\boldsymbol{x}_k^n) + A_2^n(\boldsymbol{x}_k^n)\boldsymbol{x}_k^l + w_k^n \tag{4-51}$$

Rewriting (4-51) as,

$$\underbrace{\boldsymbol{x}_{k+1}^{n} - A_1^n(\boldsymbol{x}_k^n)}_{z_k} = A_2^n(\boldsymbol{x}_k^n)\boldsymbol{x}_k^l + w_k^n \tag{4-52}$$

the right hand side of the (4-52) looks like a measurement equation of linear states. We can define left hand side as pseudo measurement and use the right hand side of (4-52) as measurement model.

With the addition of pseudo measurement, state equations are evolved as follows:

$$\boldsymbol{x}_{k+1}^{l} = A_1^l(\boldsymbol{x}_k^n) + A_2^l(\boldsymbol{x}_k^n)\boldsymbol{x}_k^l + w_k^l \tag{4-53}$$

$$z_k = A_2^n(\boldsymbol{x}_k^n)\boldsymbol{x}_k^l + w_k^n \tag{4-54}$$

$$y_k = C(\boldsymbol{x}_k^n)\boldsymbol{x}_k^l + v_k \tag{4-55}$$

We can interpret this system as a system with two measurements. However, there is a correlation between pseudo measurement noise (w_k^n) and the process noise (w_k^l) which should be taken care of. The decorrelation of the noises can be accomplished by Gram – Schmidt procedure [23].

$$\bar{w}_k^l = w_k^l - E\{w_k^l(w_k^n)^T\}E\{w_k^n(w_k^n)^T\}^{-1}w_k^n \tag{4-56}$$

By using the definition in (4-27),

$$\bar{w}_k^l = w_k^l - Q_k^{ln}(Q_k^n)^{-1}w_k^n \tag{4-57}$$

The covariance of $\bar{w}_k^l$is as in (4-58).

$$\bar{Q}_k^l = Q_k^l - Q_k^{ln}(Q_k^n)^{-1}Q_k^{ln} \tag{4-58}$$

To obtain decorrelated system, (4-57) is substituted into (4-53),

$$\boldsymbol{x}_{k+1}^{l} = A_1^l(\boldsymbol{x}_k^n) + A_2^l(\boldsymbol{x}_k^n)\boldsymbol{x}_k^l + \bar{w}_k^l + Q_k^{ln}(Q_k^n)^{-1}w_k^n \tag{4-59}$$

And by combining (4-54) and (4-59),

$$\boldsymbol{x}_{k+1}^{l} = A_1^l(\boldsymbol{x}_k^n) + A_2^l(\boldsymbol{x}_k^n)\boldsymbol{x}_k^l + \bar{w}_k^l + Q_k^{ln}(Q_k^n)^{-1}\big[z_k - A_2^n(\boldsymbol{x}_k^n)\boldsymbol{x}_k^l\big] \tag{4-60}$$

Rewriting (4-60), the decorrelated state space model of the system is obtained as:

$$\boldsymbol{x}_{k+1}^{l} = A_1^l(\boldsymbol{x}_k^n) + A^l(\boldsymbol{x}_k^n)\boldsymbol{x}_k^l + Q_k^{ln}(Q_k^n)^{-1}z_k + \bar{w}_k^l \tag{4-61}$$

$$A^l(\boldsymbol{x}_k^n) = A_2^l(\boldsymbol{x}_k^n) + Q_k^{ln}(Q_k^n)^{-1}A_2^n(\boldsymbol{x}_k^n) \tag{4-62}$$

The decorrelated system is composed of (4-61) as its process model and, (4-54) and (4-55) as two uncorrelated measurements of this system. From this point forward, the algorithm is constructed step by step by using the decorrelated state space system model. The pseudo code of the algorithm is given in Table 3.

Table 3: Marginalized Particle Filter Algorithm

```
1) Initialization:
   FOR i = 1:N
       Linear particles: $\boldsymbol{x}_{0|-1}^{l,(i)} = \boldsymbol{x}_0^l,\ P_{0|-1}^{(i)} = P_0$
       Nonlinear particles: $\boldsymbol{x}_{0|-1}^{n,(i)} \sim p_{x_0^n}(\boldsymbol{x}_0^n)$
   END FOR
   Set k →0
2) PF Measurement Update:
   FOR i = 1:N
       Calculation of weights: $w_k^{(i)} = p\left(y_k \middle| X_k^{n,(i)}, Y_{k-1}\right)$
   END FOR
```

```
    Normalization of weights: $w_k^{(i)} = \frac{w_k^{(i)}}{\sum_{j=1}^{N} w_k^{(j)}}$
    To conserve the notation: $\boldsymbol{x}_{k|k}^{n} = \boldsymbol{x}_{k|k-1}^{n}$
3) Resampling
    $[\boldsymbol{x}_{k|k}^{n}, \boldsymbol{x}_{k|k-1}^{l}, \boldsymbol{P}_{k|k-1}, w_k] = Resample(\boldsymbol{x}_{k|k}^{n}, \boldsymbol{x}_{k|k-1}^{l}, \boldsymbol{P}_{k|k-1}, w_k)$
4) KF: Measurement Update:
    FOR i = 1:N
        Innovation Covariance Calc.: $M_k^{(i)} = C_k^{(i)} P_{k|k-1}^{(i)} \left(C_k^{(i)}\right)^T + R_k$
        Kalman Gain Calculation: $K_k^{(i)} = P_{k|k-1}^{(i)} \left(C_k^{(i)}\right)^T \left(M_k^{(i)}\right)^{-1}$
        State Mean Estimation: $\boldsymbol{x}_{k|k}^{l,(i)} = \boldsymbol{x}_{k|k-1}^{l,(i)} + K_k^{(i)} (y_k - C_k^{(i)} \boldsymbol{x}_{k|k-1}^{l,(i)})$
        State Covariance Estimation: $P_{k|k}^{(i)} = P_{k|k-1}^{(i)} - K_k^{(i)} M_k^{(i)} \left(K_k^{(i)}\right)^T$
    END FOR
Combine and output result for time k: $\boldsymbol{x}_{k|k} = \begin{bmatrix} \boldsymbol{x}_{k|k}^{l} \\ \boldsymbol{x}_{k|k}^{n} \end{bmatrix}$
5) PF: Time Update:
    FOR i = 1:N
        Random number draw: $\overline{w}_k^{n,(i)} \sim N(0, Q_n)$
        State Prediction: $\boldsymbol{x}_{k+1|k}^{n,(i)} = A_1^n(k,i) + A_2^n(k,i) \boldsymbol{x}_{k|k}^{l,(i)} + \overline{w}_k^{n,(i)}$
    END FOR
6) KF: Time Update:
    FOR i = 1:N
        Innovation Calculation: $N_k^{(i)} = A_2^n(k,i) P_{k|k}^{(i)} \left(A_2^n(k,i)\right)^T + Q_k^n$
        Kalman Gain Calculation: $\overline{L}_k^{(i)} = A^l(k,i) P_{k|k}^{(i)} \left(A_2^n(k,i)\right)^T \left(N_k^{(i)}\right)^{-1}$
```

```
            State Mean Estimation and One Step Ahead Prediction:
            $\boldsymbol{x}_{k+1|k}^{l,(i)} = A^{l}(k,i)\boldsymbol{x}_{k|k}^{l,(i)} + Q_{k}^{ln}(Q_{k}^{n})^{-1}z_{k}^{(i)} + A_{1}^{l}(k,i) + \bar{L}_{k}^{(i)}(z_{k} - A_{2}^{n}(k,i)\boldsymbol{x}_{k|k}^{l,(i)})$
            State Covariance Estimation and One Step Ahead Prediction:
            $P_{k+1|k}^{(i)} = A^{l}(k,i)P_{k|k}^{(i)}\left(A^{l}(k,i)\right)^{T} - \bar{L}_{k}^{(i)}N_{k}^{(i)}\left(\bar{L}_{k}^{(i)}\right)^{T} + \bar{Q}_{k}^{l}$
        END FOR
    7) k = k +1
    8) Go to 2
```

Details of the Resampling step are given in 4.8.4.

4.8.3.1. Initialization

The algorithm starts with Initialization. N particles are selected for the linear and the nonlinear states. Nonlinear particles are drawn from the initial pdf of the nonlinear states. Although, it is not necessary for our problem nonlinear states are assumed to be Gaussian in general. Linear states are assumed to be Gaussian and linear particles and their corresponding covariance matrices are initialized as mean and covariance matrix of the initial pdf of the linear states. In this algorithm, a Kalman filter is constructed for each particle.

$$\boldsymbol{x}_{0|-1}^{n,(i)} \sim p_{\boldsymbol{x}_{0}^{n}}(\boldsymbol{x}_{0}^{n}) \tag{4-63}$$

$$\begin{aligned} \boldsymbol{x}_{0|-1}^{l,(i)} &= \boldsymbol{x}_{0}^{l}, \forall i = 1..N \\ P_{0|-1}^{(i)} &= P_{0}, \forall i = 1..N \end{aligned} \tag{4-64}$$

4.8.3.2. Measurement Update of Nested Particle Filter

Measurement update of the particle filter is done via weight update in the classical particle filter. The density of the output prediction is given in (4-65).

$$p(y_k|X_k^n, Y_{k-1}) = N(C_k \boldsymbol{x}_{k|k-1}^l, C_k P_{k|k-1} {C_k}^T + R_k) \tag{4-65}$$

The predictions of nonlinear particles are used to calculate output prediction as in (4-65). After that, weight update is accomplished by the calculation of the measurement likelihood for each output prediction calculated from N particles.

$$w_k^{(i)} = p\left(y_k \middle| X_k^{n,(i)}, Y_{k-1}\right) \tag{4-66}$$

After the weight update for all the particles is completed, normalization should be done to make sure that the sum of weights is equal to one.

$$w_k^{(i)} = \frac{w_k^{(i)}}{\sum_{j=1}^N w_k^{(j)}} \tag{4-67}$$

4.8.3.3. Measurement Update of Nested Kalman Filter

The information available in the actual measurement y_k is used to update linear states x_k^l. The measurement y_k is assumed to be Gaussian; the linear state can be estimated by Kalman Filtering. Assuming that the estimates of the nonlinear states and the predictions of the linear states at time k are known, linear states estimation is done as follows (The '(i)' notation denotes the i$^{\text{th}}$ particle):

$$\boldsymbol{x}_{k|k}^{l,(i)} = \boldsymbol{x}_{k|k-1}^{l,(i)} + K_k^{(i)}(y_k - C_k^{(i)} \boldsymbol{x}_{k|k-1}^{l,(i)}) \tag{4-68}$$

$$P_{k|k}^{(i)} = P_{k|k-1}^{(i)} - K_k^{(i)} M_k^{(i)} \left(K_k^{(i)}\right)^T \tag{4-69}$$

$$K_k^{(i)} = P_{k|k-1}^{(i)} \left(C_k^{(i)}\right)^T \left(M_k^{(i)}\right)^{-1} \tag{4-70}$$

$$M_k^{(i)} = C_k^{(i)} P_{k|k-1}^{(i)} \left(C_k^{(i)}\right)^T + R_k \tag{4-71}$$

4.8.3.4. Time Update of Nested Particle Filter

Time update (prediction) of nested particle filter is done after the measurement update of Kalman filters of each particle. Since, the linear portion of the particles are used to calculate the time update of nonlinear portion of the state vector their best estimates at time k (current time) should be used. This is the reason that the time update of particle filter is done after the measurement updates of Kalman Filters. The time update of particle filter is done as follows.

$$\begin{aligned} &p\left(\boldsymbol{x}_{k+1|k}^n \middle| X_k^n, Y_k\right) = \\ &N\left(A_1^n\left(\boldsymbol{x}_{k|k}^n\right) + \right. \\ &\left. A_2^n\left(\boldsymbol{x}_{k|k}^n\right)\boldsymbol{x}_{k|k}^l,\ A_2^n\left(\boldsymbol{x}_{k|k}^n\right)P_{k|k}\left(A_2^n\left(\boldsymbol{x}_{k|k}^n\right)\right)^T + Q_k^n\right) \end{aligned} \tag{4-72}$$

Since the pdf is represented by particles, pdf update is done by particle updates. For each particle prediction is calculated as in (4-73).

Notation gets complicated from here. To simplify function of states are defined $A_1^n\left(\boldsymbol{x}_k^{n,(i)}\right) = A_1^n(k,i)$ from here on.

$$\boldsymbol{x}_{k+1|k}^{n,(i)} = A_1^n(k,i) + A_2^n(k,i)\boldsymbol{x}_{k|k}^{l,(i)} + \overline{w}_k^{n,(i)} \tag{4-73}$$

Of course this method is an approximation and it's close to optimal when number of particles is fairly large. The last term of (4-73) is the additive noise to spread particles ($\overline{w}_k^{n,(i)} \sim N(0, Q_k^n)$).

4.8.3.5. Time Update of Nested Kalman Filter

Despite this part is called time update, it is both measurement update with pseudo measurement and time update of linear portion of particles. After the time update of particle filter, the pseudo measurement defined in (4-52) is available. Since the measurements are uncorrelated, the measurement update with pseudo measurement is done as in classical Kalman Filter. The ‘*’ notation is used to distinguish between first and second measurement update.

$$\boldsymbol{x}_{k|k}^{l,(i)^*} = \boldsymbol{x}_{k|k}^{l,(i)} + L_k(z_k - A_2^n(k,i)\boldsymbol{x}_{k|k}^{l,(i)}) \tag{4-74}$$

$$P_{k|k}^{(i)^*} = P_{k|k}^{(i)} - L_k^{(i)} N_k^{(i)} \left(L_k^{(i)}\right)^T \tag{4-75}$$

$$L_k^{(i)} = P_{k|k}^{(i)} \left(A_2^n(k,i)\right)^T \left(N_k^{(i)}\right)^{-1} \tag{4-76}$$

$$N_k^{(i)} = A_2^n(k,i) P_{k|k}^{(i)} \left(A_2^n(k,i)\right)^T + Q_k^n \tag{4-77}$$

The final job is to do the time update of the linear portions of particles. Since the state noise is assumed as Gaussian, the time update can be handled by Kalman Filter. According to process model of decorrelated system (4-61) and the last estimation, the state update is accomplished as follows.

$$\boldsymbol{x}_{k+1|k}^{l,(i)} = A_1^l(k,i) + A^l(k,i)\boldsymbol{x}_{k|k}^{l,(i)^*} + Q_k^{ln}(Q_k^n)^{-1} z_k^{(i)} \tag{4-78}$$

It is not very easy to derive state covariance prediction. By combining (4-74) and (4-78), an expression analogous to measurement update occurs.

$$\begin{aligned} \boldsymbol{x}_{k+1|k}^{l,(i)} &= A^l(k,i)\boldsymbol{x}_{k|k}^{l,(i)} + Q_k^{ln}(Q_k^n)^{-1} z_k^{(i)} + A_1^l(k,i) + \\ &\bar{L}_k^{(i)}(z_k - A_2^n(k,i)\boldsymbol{x}_{k|k}^{l,(i)}) \end{aligned} \tag{4-79}$$

$$\bar{L}_k^{(i)} = A^l(k,i) L_k^{(i)} \tag{4-80}$$

By using the analogy between (4-74) and (4-79), the prediction of covariance is obtained as follows.

$$P_{k+1|k}^{(i)} = A^l(k,i)P_{k|k}^{(i)}\left(A^l(k,i)\right)^T - \bar{L}_k^{(i)}N_k^{(i)}\left(\bar{L}_k^{(i)}\right)^T + \bar{Q}_k^l \qquad (4\text{-}81)$$

By this last step, all the steps of single model MPF algorithm are completed.

4.8.4. Particle Degeneracy

Particle filtering, weights of the particles are calculated in the estimation (weight update) stage. As time passes, weights of to all particles tend to zero except few. Therefore, number of effective particles that represent the pdf is very low. In other words, pdf is not represented well. This problem is called the degeneracy problem. To overcome this problem resampling method is offered.

In resampling, the particles with weights close to zero are replaced with particles whose weights are relatively high. This way effective number of particles remains high. Effective number of particles is calculated in [22] as follows.

$$N_{eff} = \frac{1}{\sum_{i=1}^{N}\left(w_k^i\right)^2} \qquad (4\text{-}82)$$

N_{eff} represents number of effective particles which can be both used to characterize the filter performance and decide whether to do resampling or not by comparing with a threshold value. If computational burden is not a concern resampling can be done in each step of the filtering [22].

Table 4: Resampling Algorithm

```
1) Construction of Cumulative Sum of Weights (CSW)
        Initialization: $c_1 = w_k^1$
        FOR i = 1:N
            $c_i = c_{i-1} + w_k^i$
        END FOR
2) Resampling
```

```
Starting from the 1st value of CSW: $i = 1$
Draw a starting point: $u_1 \sim U(0, N^{-1})$
FOR j = 1:N
    Move in CSW: $u_j = u_i + N^{-1}(j - 1)$
    WHILE $u_j > c_i$
        $i = i + 1$
    END WHILE
Assign particle: $x_k^{j^*} = x_k^i$
Assign weight: $w_k^{j^*} = w_k^i$
END FOR
```

For Marginalized particle filter, resampling is applied to the linear particles covariance matrices as well as augmented linear and nonlinear particles.

4.9. Multiple Model Filtering

4.9.1. Interacting Multiple Model (IMM) - EKF

In this study, Interacting Multiple Model (IMM) structure is used to interpret multiple models with EKF. Two different mode-matched filters are used for the thrust mode and the ballistic mode. In the IMM estimator, these two filters are run in parallel and the state estimates at the beginning of a cycle are a mixture of the estimates of the previous cycle using the "mixing" probabilities [8]. The IMM structure can be seen in Figure 14.

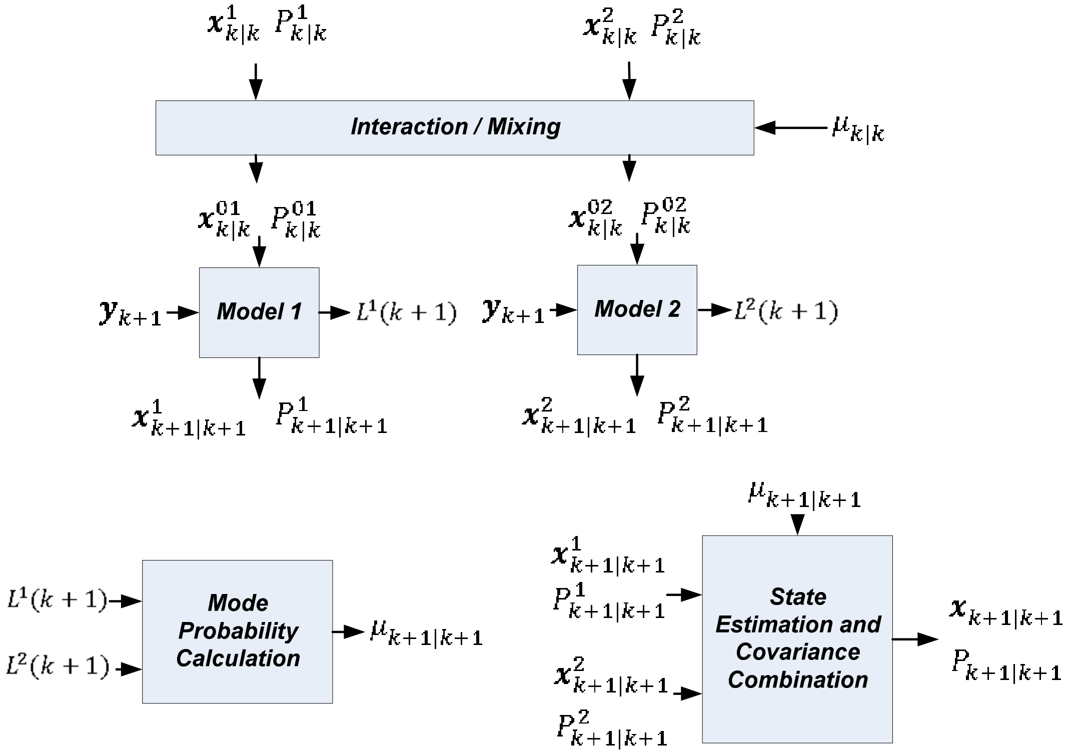

Figure 14: One Cycle of the IMM Estimator [21]

The interaction mixing stage is calculation of the weighted average of the state estimation at time k by using the mode probabilities at time k. The mixed state mean is calculated as follows:

$$x_{k|k}^{01} = \mu_{k|k}^{1|1} x_{k|k}^1 + \mu_{k|k}^{1|2} x_{k|k}^2 \tag{4-83}$$

$$x_{k|k}^{02} = \mu_{k|k}^{2|1} x_{k|k}^1 + \mu_{k|k}^{2|2} x_{k|k}^2 \tag{4-84}$$

However the state dimension is not necessarily the same in some multi model filtering applications. Therefore, the equations (4-83) and (4-84) are not applicable. An idea to augment the lesser dimensioned models to equate the state dimension for each model inserts a bias to the state estimation towards zero. Therefore, an unbiased mixing algorithm should be applied. The detailed derivation of this algorithm can be found in [8].

4.9.2. Regime Transition (RT) – Particle Filter

In this study, Regime Transition (RT) structure is used with both SIR-PF and MPF. Regime Transition is a jumped Markov process that is applied to PF for tracking with multiple models [22].

The regime variable r_k^n, determines which dynamic model is in effect from t_{k-1} to t_k. The first step of this algorithm is to generate a random set of the regime variable for each particle based on the transitional probability matrix π. After the initialization, each particle is filtered with the corresponding filter (or model). The particles with high weights live and others die in the resampling step of the algorithm. After the resampling, regime transition is applied to all particles ensure the mode transition according the transitional probability matrix π. Regime transition algorithm is given in Table 5.

Table 5: Regime Transition Algorithm

```
FOR n = 1:N
    Draw $u_n \sim U(0,1)$
    IF $r_{k-1}^n == 1 \;\&\; u_n > \pi_{11}$
        Set $r_k^n = 2$
    ELSE IF $r_{k-1}^n == 2 \;\&\; u_n > \pi_{22}$
        Set $r_k^n = 1$
    END IF
END FOR
```

CHAPTER 5

SIMULATIONS AND DISCUSSION

The tracking methodology and the algorithms that are used for short range thrusting ballistic projectile tracking are given in the previous chapters. In this chapter, the estimates of these algorithms and their performances will be discussed and compared. Monte Carlo analyses are performed and the mean RMS error of the position and the velocity estimation of the projectile are compared for each tracking filter for different scenarios.

5.1. Simulation Environment

5.1.1. Trajectory Generation

The simulation of the target trajectory is constructed by discretized motion model (4-11). For trajectory generation, the sampling period is taken very small compared to the tracking period so the accuracy of the simulated trajectory is high as if it is simulated in continuous time.

A hypothetical projectile is assumed for this study. The thrust profile of the projectile is assumed to be constant while the projectile is in the thrusting mode and zero otherwise. Parameters of the hypothetical projectile are as follows.

- Muzzle Velocity (Initial Velocity): 300 m/s
- Motor Ignition Time (Thrust Start Time): 1 second
- Motor Depletion Time (Thrust End Time) :2 seconds
- Thrust Force: 1000 N
- Diameter of the Projectile: 40 mm

- Mass of the Projectile: 2 kg
- Zero Yaw Drag Coefficient (C_d): 1 (Unitless)

It is assumed that there are only drag, thrust and gravitational forces acting on the projectile. Other forces are considered as negligible [7]. The trajectory generated for five degrees of projection angle (the pitch angle of the launcher) is seen in Figure 15. In the trajectory generation and tracking, Earth is assumed to be flat.

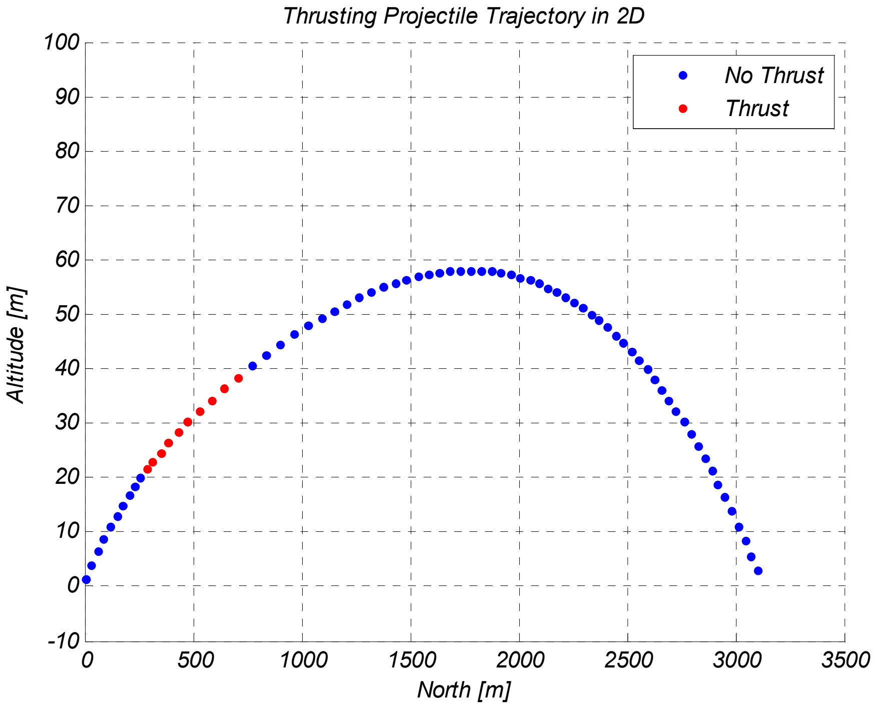

Figure 15: Thrusting Projectile Trajectory in 2D

The projectile assumed to be fired to the zero north direction and there is no deflection on the east axis. The thrust curve of the projectile is given in Figure 16.

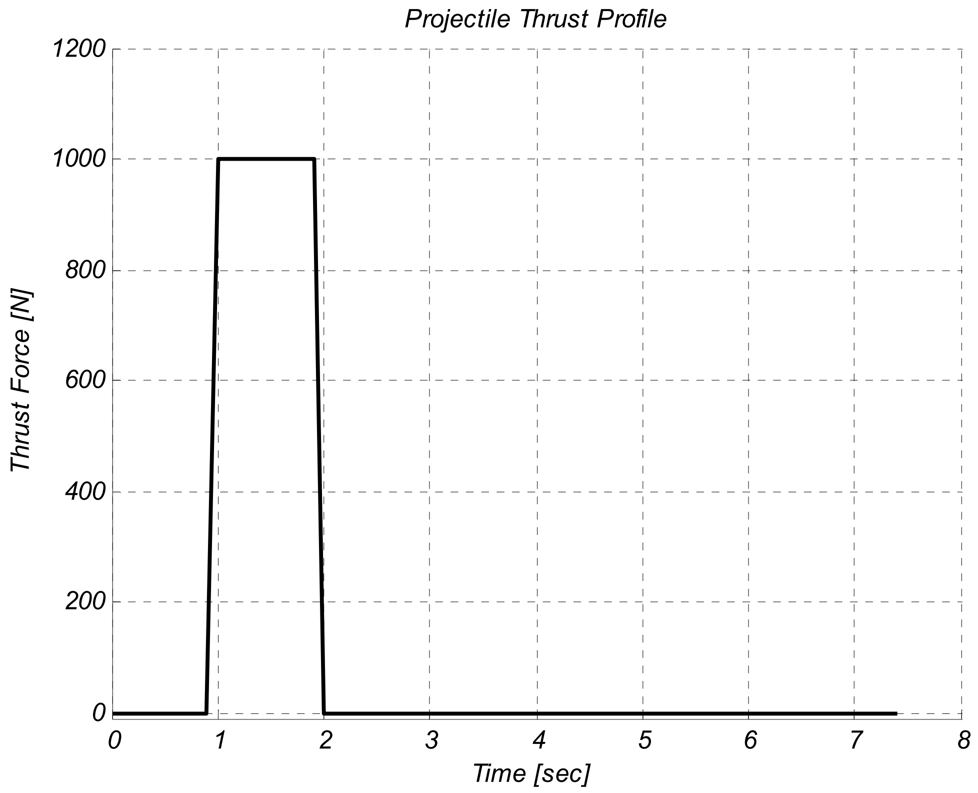

Figure 16: Projectile Thrust Profile

5.1.2. Target Scenarios

Four different scenarios are generated in this study. These scenarios address different kinds of challenges for the tracker. In two of these scenarios, projectiles are flying perpendicular to the boresight of the radar and in the other two; projectiles are flying parallel to the boresight of the radar. The reason for the selection of parallel and perpendicular trajectories is that for parallel flight the range error and for perpendicular flight the angle errors are dominant on the flight direction of the projectile. In addition to that, track initialization performances differ for parallel and perpendicular flights because of the nonlinearity in the measurement. The details of these scenarios are given in the following subsections.

5.1.2.1. Target Scenario 1: Ballistic Phase Only Flyby Projectile

In the first scenario, projectile is assumed to be fired perpendicular to the boresight of the radar and the projectile is always in the ballistic phase when it is seen by the radar.

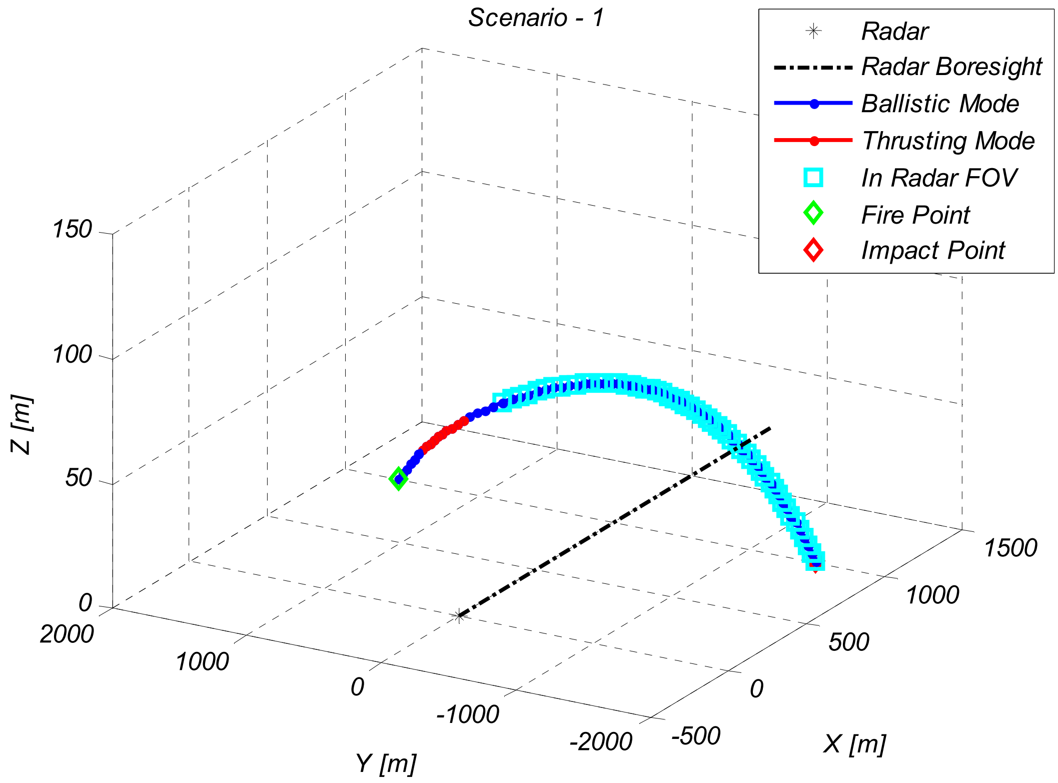

Figure 17: Scenario – 1 (Ballistic phase only flyby projectile scenario)

As the colors indicate in Figure 17, projectile enters radars field of view after its rocket motor is depleted. Initial measurements are observed on the upleg of the projectile. The projectile is tracked by the radar up to the impact point.

5.1.2.2. Target Scenario 2: Multiple Phases Flyby Projectile

In the second scenario, projectile is assumed to be fired perpendicular to the boresight of the radar and the projectile changes phases between thrusting and ballistic through the flight.

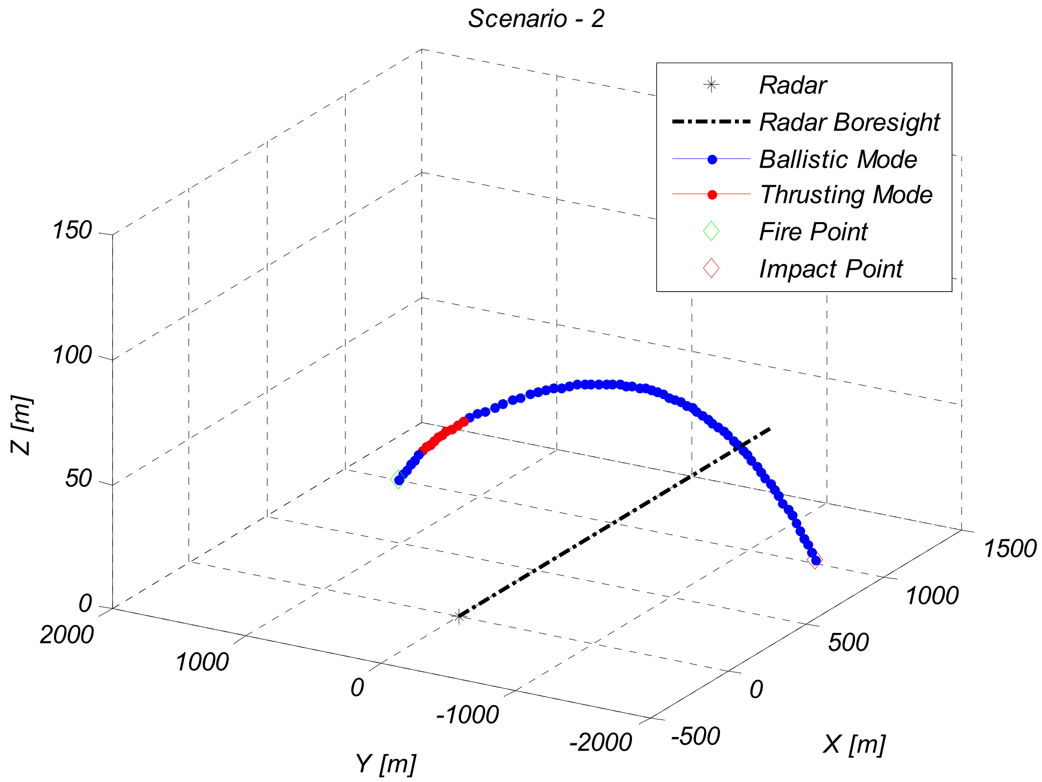

Figure 18: Scenario – 2 (Multiple phases flyby projectile scenario)

In scenario - 2, projectile is observed by the radar from the firing point to the impact point. In other words, projectile is always in the radar's field of view (FOV).

5.1.2.3. Target Scenario 3: Ballistic Phase Only Incoming Projectile

In the third scenario, projectile is assumed to be fired parallel to the boresight of the radar and the projectile is always in ballistic phase when it is seen by the radar.

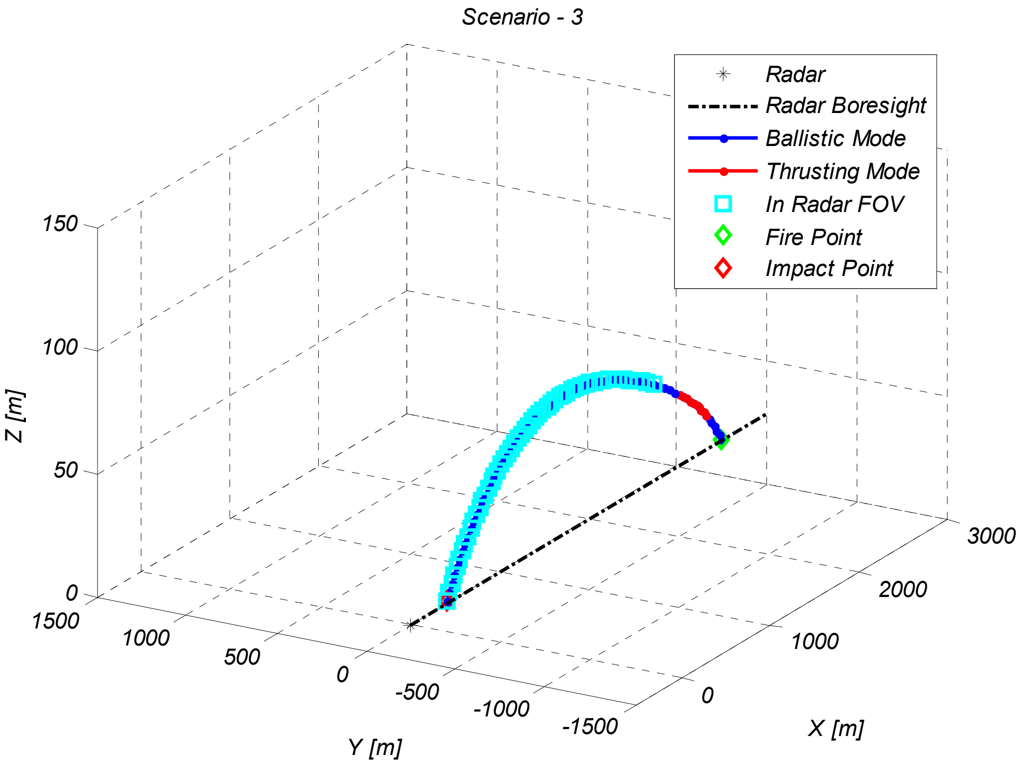

Figure 19: Scenario – 3 (Ballistic phase only incoming projectile scenario)

As the colors indicate in Figure 19, projectile enters radars field of view after its rocket motor is depleted. Initial measurements are observed on the upleg of the projectile. The projectile is tracked by the radar up to the impact point.

5.1.2.4. Target Scenario 4: Multiple Phases Incoming Projectile

In the second scenario, projectile is assumed to be fired parallel to the boresight of the radar and the projectile changes phases as ballistic-thrusting-ballistic during the radar observations as indicated in Figure 20. The projectile is observed by the radar from the firing point to the impact point.

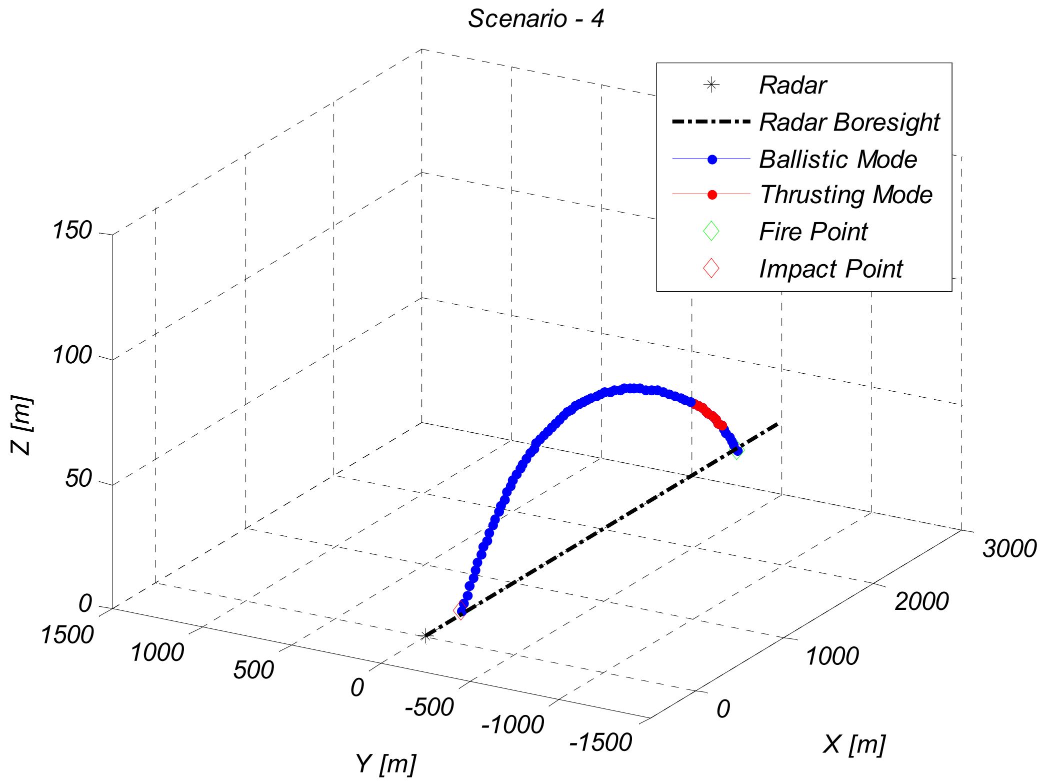

Figure 20: Scenario – 4 (Multiple phases incoming projectile scenario)

5.1.3. Radar Model

Radar is assumed to have a constant field of view and constant track period through the flight of the projectile. The measurement noise in azimuth, elevation and range are modeled as zero mean white Gaussian random variables. The standard deviations of these noises depend on the Signal to Noise Ratio (SNR) as well as the range. For measurement generation, the generated trajectory is down sampled to the desired measurement period and transformed to the spherical coordinates. Then, the drawn measurement noise samples are added to the ideal measurements.

A hypothetical radar system is designed for this study. Radar is assumed to have an approximately 3 km of instrumented range. For the instrumented range, the probability of detection (P_d) is assumed to be 0.85 and probability of false alarm is assumed to be (P_{fa}) 10^{-6}. For Swerling-1 type of targets, this leads to 19.2 dB of SNR on the instrumented range. SNR versus range graph can be seen in Figure 21. The SNR is inversely proportional to the target's range to the fourth power ($^1/_{R^4}$).

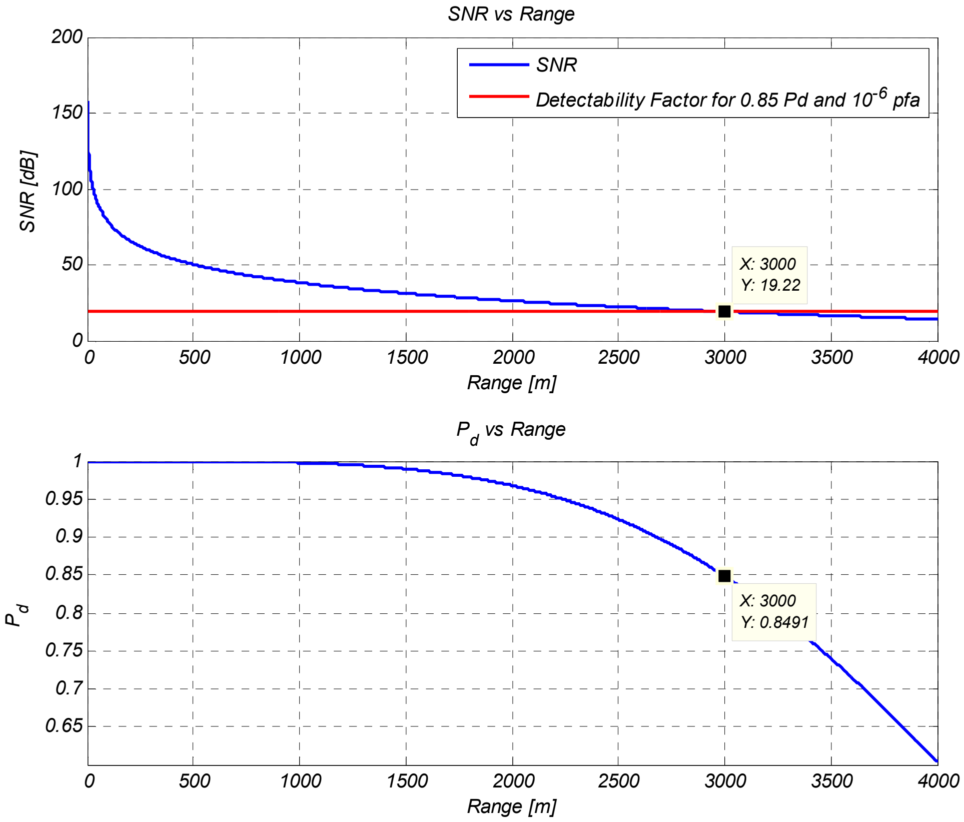

Figure 21: SNR and Pd graphs of a hypothetical simulation radar

As mentioned in section 2.3, measurement noises of the radar are inversely proportional to the square root of the SNR. A single run noisy, measured trajectory and the true trajectory can be seen in Figure 22.

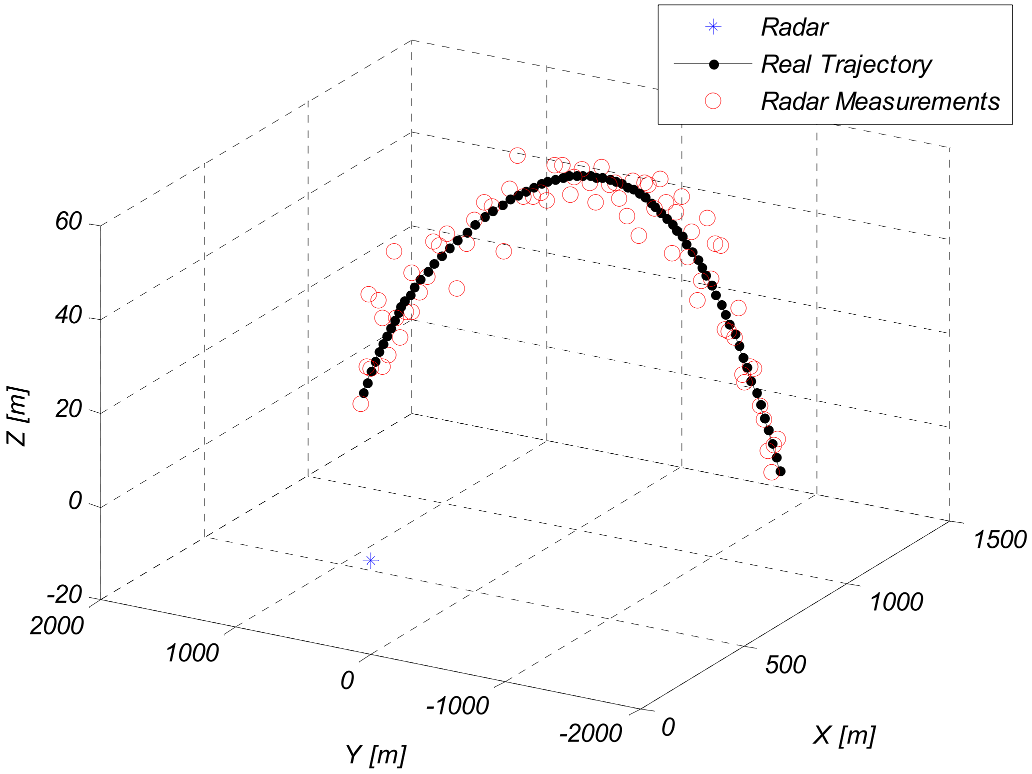

Figure 22: Radar Measurements

5.2. Simulation Results

In this subsection, simulations are performed to analyze the performances of the tracking algorithms proposed in the previous sections. Performance criteria of the simulations are root mean square error of the estimated positions, velocities and ballistic parameters.

The simulation parameters are as follows:

- Sampling period used for trajectory generation: 1 ms
- Sampling periods of measurements and tracker: 100 ms
- Target launch position: [X, Y, Z] = [1500 m, 1000 m, 0 m]
- Target launch angle (elevation) : = 5°

5.2.1. Estimation Performance

To investigate the estimation performance of the proposed tracking algorithms, Monte Carlo analysis is performed for all proposed scenarios. Each algorithm is run for 100 different randomly generated measurement sets and RMS error is calculated at each time instant.

RMS position estimation errors are calculated as follows:

$$RMS_{position}(k) = \sqrt{\frac{1}{N_{mc}}\sum_{mc=1}^{N_{mc}}[x_{error}{}^2(k,mc) + y_{error}{}^2(k,mc) + z_{error}{}^2(k,mc)]} \quad (5\text{-}1)$$

where,

$$\begin{aligned} x_{error}(k,mc) &= x_{est}(k,mc) - x_{real}(k,mc), \forall k, mc \\ y_{error}(k,mc) &= y_{est}(k,mc) - y_{real}(k,mc), \forall k, mc \\ z_{error}(k,mc) &= z_{est}(k,mc) - z_{real}(k,mc), \forall k, mc \end{aligned} \quad (5\text{-}2)$$

RMS velocity estimation errors are calculated as follows:

$$RMS_{velocity}(k) = \sqrt{\frac{1}{N_{mc}}\sum_{mc=1}^{N_{mc}}\left[Vx_{error}{}^2(k,mc) + Vy_{error}{}^2(k,mc) + Vz_{error}{}^2(k,mc)\right]} \quad (5\text{-}3)$$

where,

$$\begin{aligned} Vx_{error}(k,mc) &= Vx_{est}(k,mc) - Vx_{real}(k,mc), \forall k, mc \\ Vy_{error}(k,mc) &= Vy_{est}(k,mc) - Vy_{real}(k,mc), \forall k, mc \\ Vz_{error}(k,mc) &= Vz_{est}(k,mc) - Vz_{real}(k,mc), \forall k, mc \end{aligned} \quad (5\text{-}4)$$

RMS Thrust and Drag parameter estimation errors are calculated as follows:

$$RMS_{Drag}(k) = \sqrt{\frac{1}{N_{mc}} \sum_{mc=1}^{N_{mc}} \left(Drag_{est}(k,mc) - Drag_{real}(k,mc)\right)^2} \qquad (5\text{-}5)$$

$$RMS_{Thrust}(k) = \sqrt{\frac{1}{N_{mc}} \sum_{mc=1}^{N_{mc}} \left(Thrust_{est}(k,mc) - Thrust_{real}(k,mc)\right)^2} \qquad (5\text{-}6)$$

For these simulations, number of particles for PF and MPF are selected 30k and track update time is selected 100 milliseconds. The probability of detection at 3km is set as 0.85 and it decreases with range. It is assumed that there is no false alarm.

Position and Velocity estimation RMS error plots for each scenario are as follows:

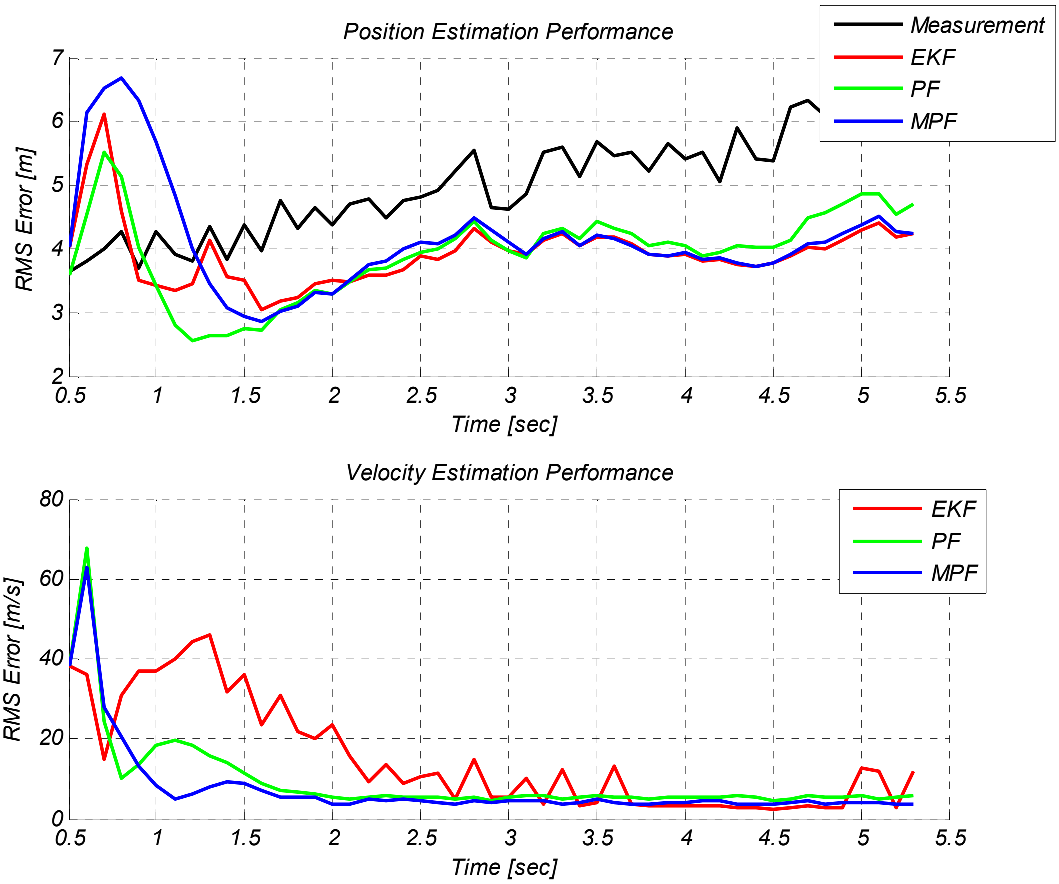

Figure 23: Position and Velocity Estimation Performances of Scenario -1

For Scenario-1, in which projectile flies perpendicular to the field of view of the radar and it is always in the ballistic mode; the position estimation performances of EKF and MPF are very close and they give better estimates compared to the PF. However, the velocity estimation performance of the EKF is worst for the half of the track but when the estimation is settled all filters give almost the same performance. It is not unfair to say that the overall performance of the MPF is best for this scenario.

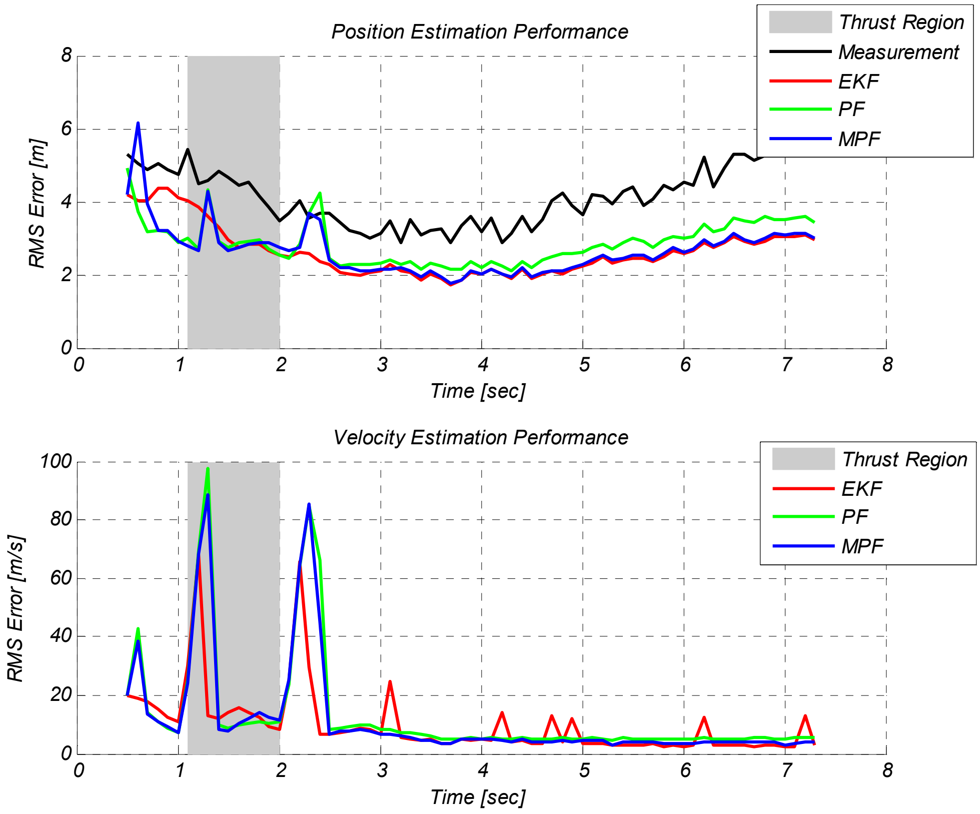

Figure 24: Position and Velocity Estimation Performances of Scenario -2

In Scenario -2, the projectile flies perpendicular to the line of sight of the radar and the flight contains both the thrusting and the ballistic modes. For this scenario, it is seen that particle filter and marginalized particle filter both have larger errors in mode transitions compared to the EKF. Both the position and velocity error plots seem to be steady state after 3 seconds. At the steady state, the EKF and the MPF show very similar performances. After a period from phase transition, PF and MPF settle and give their steady state performance, where the performances are similar to that of Scenario -1 as expected.

As it can be seen from Figure 24, the mode transition performance of the EKF is better than the other filters. However, in the steady state there are spikes in the velocity estimation of the EKF. In other words, EKF tends to switch between modes more than other filters even the same mode transition probabilities are used for all filters. Spikes in the velocity estimation error of EKF are caused by the false transition to the thrusting phase.

In Scenarios 1 and 2, the measurement error starts decreasing as the range of the projectile decreases and after passing the boresight of the radar the error increases as the range of the projectile increases. This is the reason that the position estimation error of the tracking filters is increasing after 4th second.

For Scenario -3 and Scenario -4, results are very similar but the MPF performance is slightly better compared to the other results.

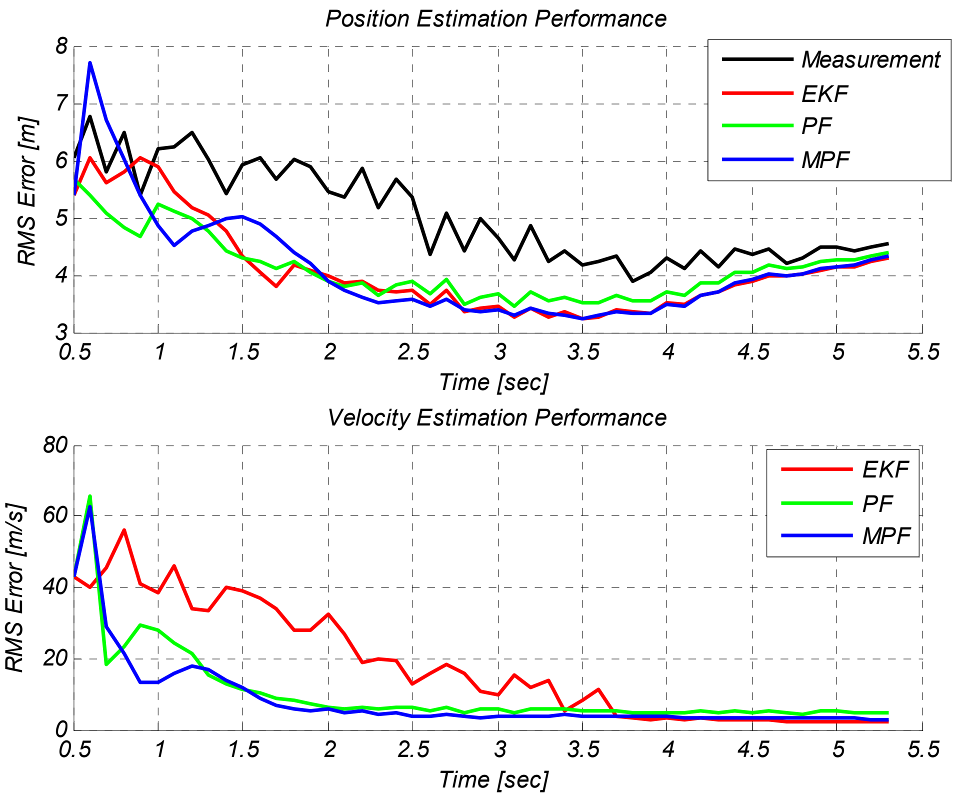

Figure 25: Scenario -3 Position and Velocity Estimation Performances of Filters

Again for the single ballistic mode case it can be seen that overall performance of the MPF is better than others.

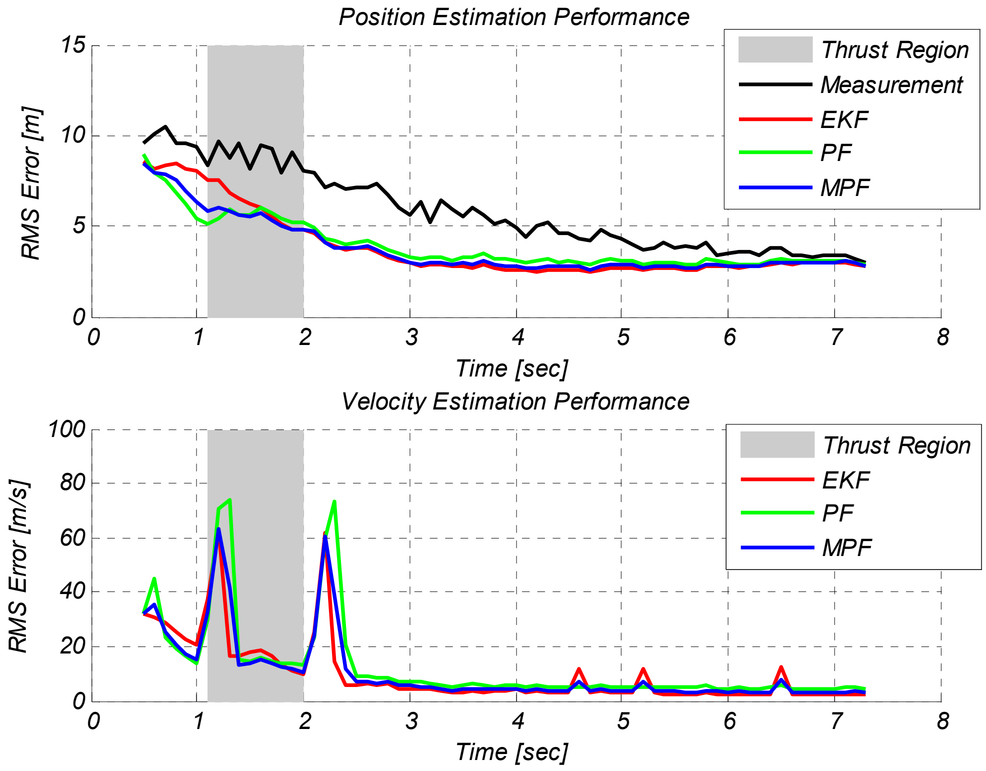

Figure 26: Scenario -4 Position and Velocity Estimation Performances of Filters

The performance of EKF is slightly better. One reason is that, the selected time step is fairly small and because of this nonlinearity is not effective. But for these kinds of applications, time steps are generally smaller than the one used in simulations. Another reason is that the state dimension is large and number of particles should be fairly large (a lot larger than 30k) to obtain the better performance than EKF.

Table 6: Mean RMS Position Error [m] Comparison

	Scenario - 1	**Scenario – 2**	**Scenario – 3**	**Scenario – 4**
EKF	3.8617	2.4239	3.7603	3.2312
PF (30k particles)	3.9717	2.7742	3.8926	3.5287
MPF (30k particles)	3.8801	2.5080	3.8047	3.6117

Table 7: Mean RMS Velocity Error [m/s] Comparison

	Scenario - 1	Scenario – 2	Scenario – 3	Scenario – 4
EKF	15.0913	7.9601	12.9662	6.6286
PF (30k particles)	9.4410	9.9252	5.9904	9.1856
MPF (30k particles)	7.6078	8.7107	4.4790	7.9827

When we look at the comparison tables, it is seen that the filter position estimation performances are very similar but EKF is slightly better than others. For velocity estimation overall performance of MPF is better than others.

The envelope of the measurements and tracker outputs are investigated for Scenario-2.

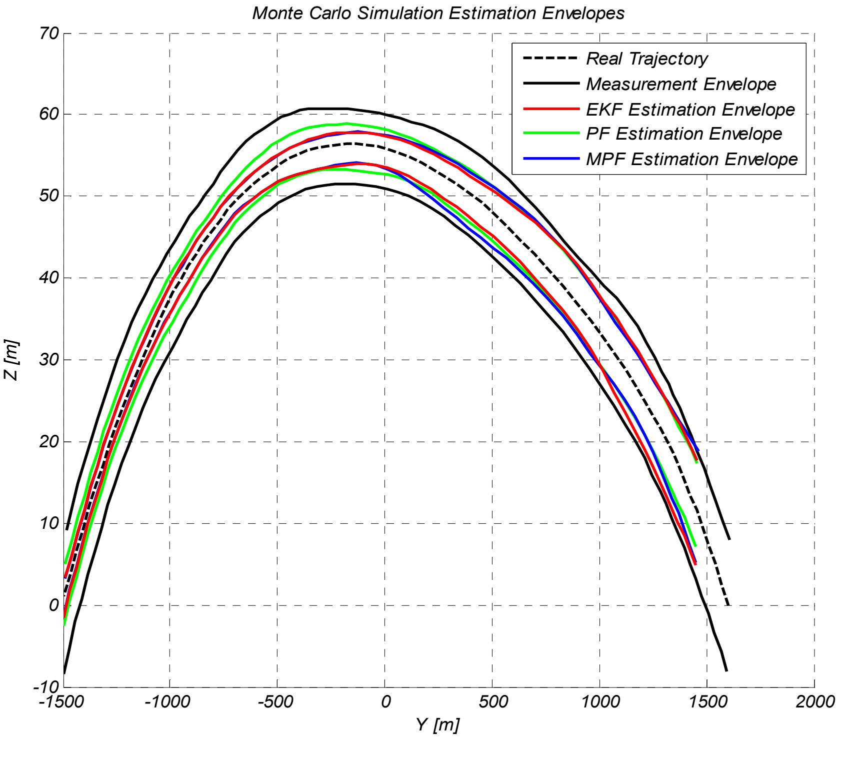

Figure 27: Scenario -2 Monte Carlo Simulation Measurement and Estimation Envelopes

As it can be seen from the Figure 27, the envelope of the tracker outputs are in parallel with the RMSE plots. The performances of the EKF and MPF are very close to each other and the PF has the worst performance. Envelopes are obtained by taking maximum and minimum values of the Monte Carlo runs in each time instant.

The estimation of the parameters of drag and thrust are important for both projectile identification and impact – fire point prediction. The drag and thrust parameters estimation performances of filters for scenario – 2 can be seen in Figure 28.

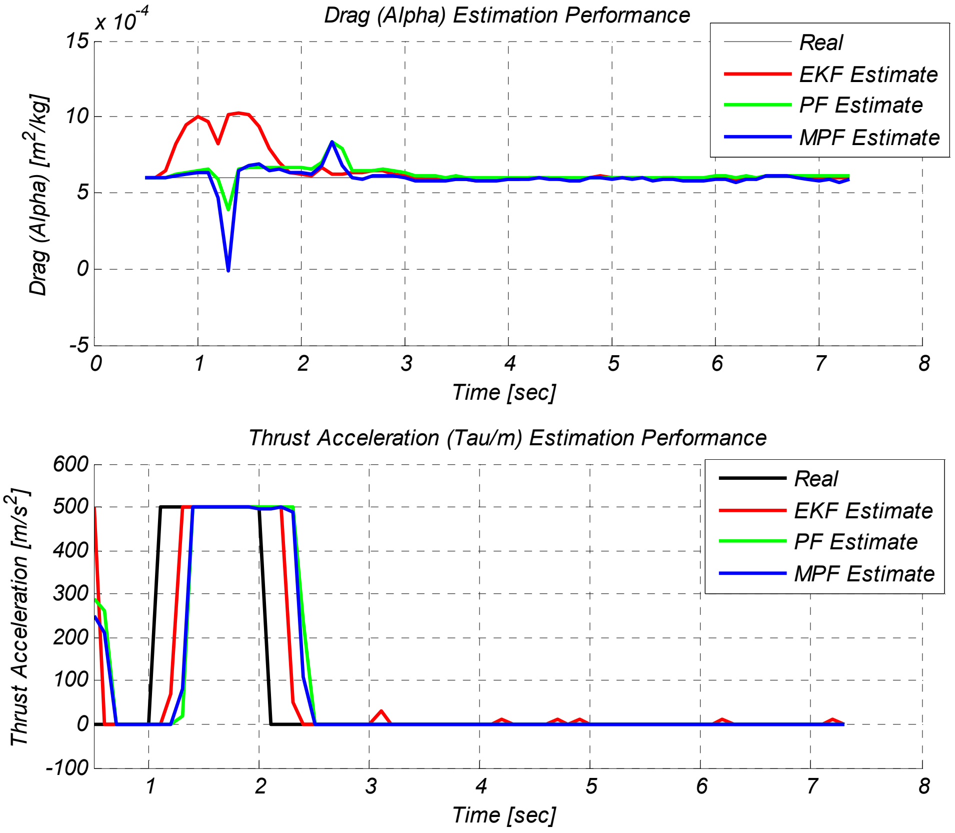

Figure 28: Scenario -2 Drag and Thrust Estimation Performances of Filters

The parameter estimation performances of EKF are better than the other filters for scenario - 2. Same parameter estimation performances observed for other scenarios too. The spike on the drag parameter estimation is caused by the transition from ballistic and thrusting phases because thrust and drag forces are coupled. In other words, total force on the same axis is trying to be estimated by sum of two independent forces. To overcome this coupling in parameters estimation, the process noise and initial variance of the drag coefficient is selected fairly small compared to thrust parameter. In other words, the drag coefficient (alpha) of the projectile assumed as known or estimated by bank of filters [8].

5.2.2. Effect of Number of Particles on Estimation Performance

In this part effect of the number of particles on position and velocity estimation performance of PF and MPF is analyzed. For the Scenario – 2, PF and MPF are applied for different number of particles. In the analysis, 1k, 3k, 10k and 30k particles are used.

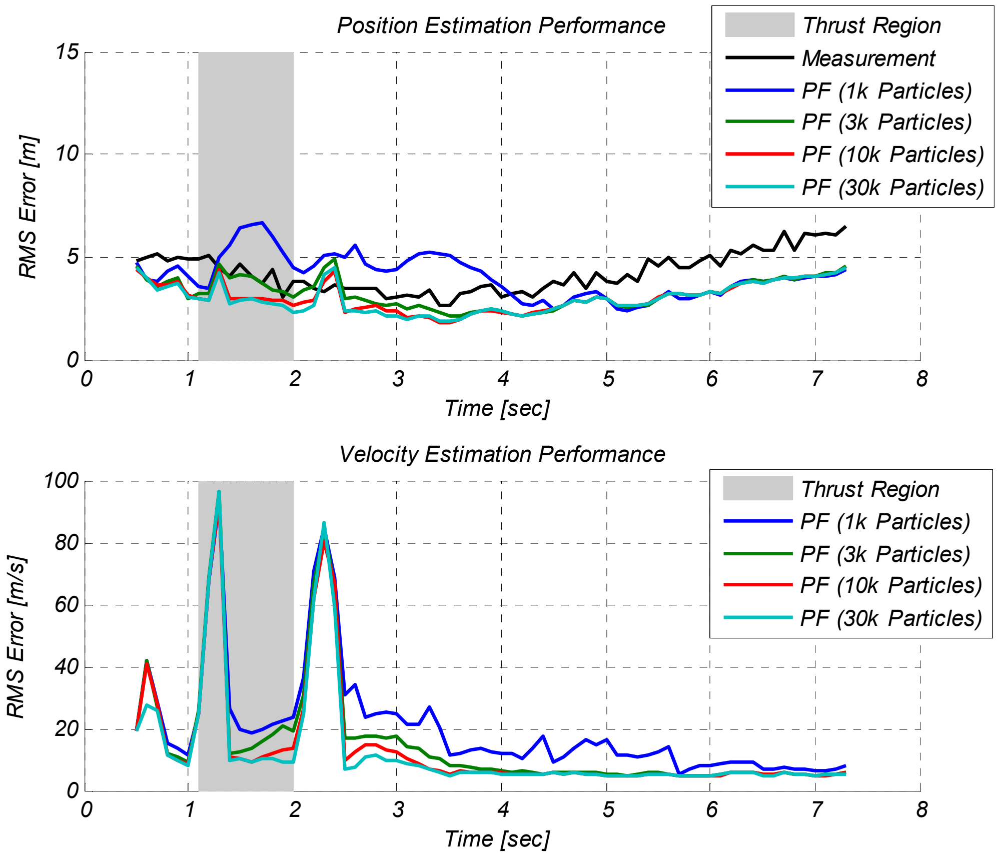

Figure 29: Effect of the Number of Particles on PF

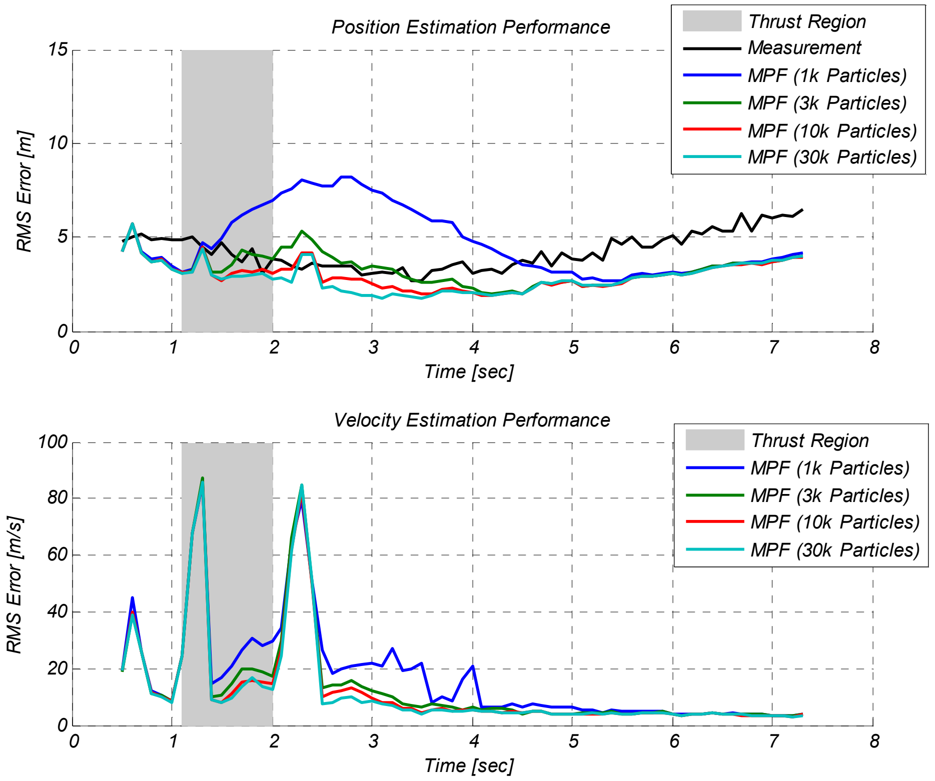

Figure 30: Effect of the Number of Particles on MPF

As it can be seen from the Figure 29 and Figure 30, estimation performance gets better as the number of particles increases as expected. The mean RMSE in time is given in Table 8.

Table 8: Mean RMSE for Different Number of Particles

Algorithm	**MPF**		**PF**	
	Pos. RMSE [m]	**Vel. RMSE [m/s]**	**Pos. RMSE [m]**	**Vel. RMSE [m/s]**
1k Particle	4.888	14.391	4.055	17.890
3k Particle	3.149	10.680	3.186	12.233
10k Particle	2.826	9.479	2.942	10.534
30k Particle	2.669	9.061	2.906	9.831

State covariance estimation performances are analyzed. The results are parallel with the RMS errors.

5.2.3. Computational Complexity

The algorithms are run one by one for different number of particles and their calculation times are observed. The calculation complexity for each algorithm can be seen in Table 9.

Table 9: Calculation Load Comparison

Algorithm	**Relative Computational Load**
EKF	1
PF (1k Particle)	240
MPF (1k Particle)	560
PF (N k Particle)	240 x N
MPF (N k Particle)	560 x N

There is a big difference in computation times of EKF and other filters. And the MPF is the filter with highest computation time. Also the computation of the particle filters is linearly depended with the number of particles. It is mentioned that the MPF performance is comparable with the EKF for 30k of particles, but the computational complexity of MPF is nearly 17000 times higher than the EKF's for this case.

CHAPTER 6

CONCLUSIONS

In this thesis, tracking of thrusting projectile with radar problem is studied. Three different multiple model tracking filters are implemented and their estimation performances are compared. These filters are the Extended Kalman Filter (EKF), the Particle Filter (PF) and the marginalized particle filter (MPF). The performances are compared by RMS error calculation of state estimations by Monte Carlo simulations.

In the 2nd chapter, fundamental information about radar is given. The probability of detection and probability of false alarm terms for radar's detection procedure are discussed. Furthermore, the mechanisms in the radar for generation of the measurements of the target are mentioned. The characteristics of the range, Doppler and direction finding (DF) measurements of the radar are given.

In the 3rd chapter, theory on dynamics of the thrusting and the ballistic phases of the thrusting projectile is given in detail. The projectile is modeled as a point mass model and forces acting on the projectile are characterized. The continuous time dynamic model of the projectile for both the ballistic phase and the thrusting phase is constructed by taking into consideration of the dominant forces acting on the projectile. The forces taken into account are the drag force, the thrust force and the gravitational force.

In the 4th chapter, firstly, the discrete time motion model of the thrusting projectile is constructed by Euler's forward method from the continuous time model. Also, the discretized motion model's linear and nonlinear portions are separated with some approximations. Algorithm blocks and pseudo codes of the classical EKF, PF and MPF are given. Most of the effort in this chapter is to apply the ballistic model and thrusting models to the MPF. In addition, process noise, measurement noise and the measurement models used in this study are given in detail.

Because of the lack of a sophisticated simulator for the thrusting projectile which takes into account of all important forces and moments on the projectile and generates trajectory, the trajectories for the simulations are generated using the constructed discrete time model. To insert model mismatch between tracker and trajectory models, noise is added to the generated trajectory in addition to the measurement noise.

Different scenarios are generated and the estimation performances of the tracking filters are compared. The filters are compared by the RMS errors in position and velocity estimations which are calculated by Monte Carlo results. In general, EKF has the best performance on position estimation and MPF has the best performance on velocity estimation. Overall worst performance was PF's. The reason is that the state dimension of the system model is fairly large for PF application. For the same performance, number of particles increases exponentially with the increase in state dimension. The MPF is the solution to this problem. The linear portion of the model is marginalized out and estimated by classical Kalman filter. In other words, a Kalman filter is constructed for each particle. As it can be seen, from the results MPF performance is better than the PF. However, it cannot surpass the EKF. The PF and MPF performances get better with the increase in the number of particles but even for the 30k particle case MPF does not give better results than EKF but their performances are comparable. Results are valid when the target ballistic coefficient is a priori known.

The MC simulation with 100 runs, PF and MPF comparison with 30k number of particles takes half a day with a commercial computer (2.8 GHz Processor, 4 GB RAM). Increasing number of particles may increase the estimation performance of the MPF further but, it is not reasonable to run simulations with higher number of particles because of the huge computational complexity.

In our simulations, results are in favor of the EKF. Very similar results are obtained when EKF and PF is compared in [3]. The EKF combines good estimation performance and low computational load. The reasons that the EKF is better than other compared filters are,

- The selected time step is fairly small, and because of this, nonlinearity is not effective. But, for these kinds of applications, time steps are generally smaller than the one used in simulations.
- The state dimension is large and number of particles should be fairly large (significantly larger than 30k) to obtain better performance than EKF.
- Converted measurements are used (observation is linear) and it is in favor of EKF.

As a future work, for a better comparison of the filters, they may be compared with trajectories generated by more sophisticated models and tools. It would be best; if there will be a chance to work on the real data acquired by radar. Another topic may be use of Doppler measurements to improve the estimation performance. For the projectiles coming towards to the radar (with high Doppler) it may improve the thrust estimation and mode transition of the filters. In addition to that, this study can be followed by the impact and the fire point prediction of the projectile using the tracker output. Finally, IMM particle filter can be implemented and its performance can be compared with regime transition particle filter. In IMM particle filter, number of particles in each mode is constant and same which may increase the state transition performance.

REFERENCES

[1] D. Shae, "Raffica The RPG-7 System," Small Arms Review, vol. 10, no. 3, 2006.

[2] "How Stuff Works," Discovery Company, 2004. [Online]. Available: http://science.howstuffworks.com/rpg3.htm. [Accessed 15 June 2012].

[3] A. Farina, B. Ristic, D. Benvenuti, "Tracking a Ballistic Target: Comparison of Several Nonlinear Filters," Aerospace and Electronic Systems, vol. 38, no. 3, pp. 854-867, 2002.

[4] X. R. Li, V. P Jilkov, "Survey of Maneuvering Target Tracking. Part II: Motion Models of Ballistic and Space Targets," Aerospace and Electronic Systems, vol. 46, no. 1, pp. 96-119, 2010.

[5] M. Tsai and F. A. Rogel, "Angle-only tracking and prediction of boost vehicle position," in SPIE Conference on Signal and Data Processing of Small Targets, Orlando, FL, 1991.

[6] R. Hutchins and A. San Jose, "IMM tracking of a theater ballistic missile during boost," in SPIE Conference on Signal and Data Processing of Small Targets, Orlando, FL, 1998.

[7] T. Yuan, Y. Bar-Shalom, P. Willett and D. Hardiman, "Impact Point Prediction for Short Range Thrusting Projectiles," in SPIE conference Signal and Data Processing of Small Targets, Orlando FL, 2010.

[8] T. Yuan, Y. Bar-Shalom, P. Willett, E. Mozeson, S. Pollak, D. Hardiman, "A Multiple IMM Approach with Unbiased Mixing for Thrusting Projectiles," Signal Processing, Sensor Fusion, and Target Recognition XX, vol. 8050, 2011.

[9] M.G.S Bruno, A. Pavlov, "A density assisted particle filter algorithm for target tracking with unknown ballistic coefficient," in IEEE International Conference on Acoustics, Speech, and Signal Processing, Philadelphia, 2005.

[10] M.G.S Bruno, A. Pavlov, "Improved Particle Filters for Ballistic Target Tracking," in IEEE International Conference on Acoustics, Speech, and Signal Processing, Quebec, 2004.

[11] M.G.S. Bruno, A. Pavlov, "Improved Sequential Monte Carlo Filtering for Ballistic Target Tracking," Aerospace and Electronic Systems, vol. 41, no. 3, pp. 1103-1108, 2005.

[12] T. B. Schön, F. Gustafsson, P. Nornlund, "Marginalized particle filters for mixed linear/nonlinear state-space models," IEEE Transactions Signal Processing, vol. 3, no. 7, pp. 2279-2289, 2005.

[13] T. B. Schön, R. Karlsson, F. Gustafsson, "The Marginalized Particle Filter in Practice," in IEEE Aerospace Conference, Big Sky, MT, 2006.

[14] M. Skolnik, Radar Handbook, McGraw-Hill, 2008.

[15] G. W. Stimson, Introduction to Airborne Radar, New Jersey: SciTech Publishing, 1998.

[16] R. L. McCoy, Modern Exterior Ballistics, Pennsylvania: Schiffer Publishing Ltd, 1999.

[17] NATO Standardization Agreement (STANAG) The Modified Point Mass and Five Degrees of Freedom Trajectory Models, 2005.

[18] M. J. L. Turner, Rocket and Spacecraft Propulsion : Principles, Practice and New Developments, Chichester: Berlin, Heidelberg : Praxis Publishing Ltd, 2005.

[19] R. A. Braeunig, "Rocket & Space Technology," 2009. [Online]. Available: http://www.braeunig.us/space/propuls.htm. [Accessed 15 July 2012].

[20] W. H. Press, S. A. Teukolsky, W. T. Vetterling and B. P. Flannery, Numerical Recipes in C the Art of Scientific Computing Second Edition, Cambridge: Cambridge University Press, 1992.

[21] Y. Bar-Shalom, P. Willett, X.Tian, Tracking and Data Fusion a Handbook of Algorithms, YBS Publishing, 2011.

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

[23] P. R. Kumar, Stochastic Systems Estimation, Identification and Adaptive Control, New Jersey: Prentice Hall, 1986.