TRACKING SHORT-RANGE BALLISTIC TARGETS

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

BY

RECEP SERDAR ACAR

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

SEPTEMBER 2011

Approval of the thesis:

TRACKING SHORT-RANGE BALLISTIC TARGETS

submitted by **RECEP SERDAR ACAR** 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. Erol Kocaoğlan
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 __________________

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

Dr. Hüseyin Yavuz
ASELSAN Inc. __________________

Date: __________________

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

Name, Last name : Recep Serdar ACAR

Signature :

ABSTRACT

TRACKING SHORT-RANGE BALLISTIC TARGETS

Acar, Recep Serdar

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

September 2011, 74 pages

The trajectories of ballistic targets are determined significantly by the characteristics that are specific to them. In this thesis, these characteristics are presented and a set of algorithms in order to track short-range ballistic targets are given. Firstly, motion and measurement models for the ballistic targets are formed and then four different filtering techniques are built on these models which are the extended Kalman filter, the unscented Kalman filter, the particle filter and the marginalized particle filter. The performances of these filters are evaluated by making Monte Carlo simulation. The simulations are run using target scenarios obtained according to six degrees-of-freedom trajectory for ballistic targets. Apart from the tracking errors of the filters, drag parameter estimations and the effect of drift calculation on the filter performances are investigated. The estimation results obtained by each filter are discussed in detail by making various simulations.

Keywords: Ballistic Target Tracking, Spin-Stabilized Projectile, Short-Range Ballistic Target, Kalman Filter, Particle Filter

ÖZ

KISA MENZİLLİ BALİSTİK HEDEFLERİN TAKİBİ

Acar, Recep Serdar

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

Eylül 2011, 74 sayfa

Balistik hedeflerin yörüngeleri önemli ölçüde kendilerine özgü olan özellikleri ile belirlenir. Bu tezde, bu özellikler sunulmakta ve kısa menzilli balistik hedefleri takip etmek için bir grup algoritma verilmektedir. İlk olarak balistik hedefler için hareket ve ölçüm modelleri oluşturulmakta; daha sonra bu modeller üzerine kurulan, genişletilmiş Kalman filtresi, kokusuz Kalman filtresi, parçacık filtresi ve ayrılmış parçacık filtresi adında dört ayrı filtreleme tekniği sunulmaktadır. Bu filtrelerin performansları Monte Carlo simülasyonu ile değerlendirilmektedir. Simülasyonlar, balistik hedefler için altı serbestlik-dereceli yörüngeye göre elde edilmiş hedef senaryoları ile koşturulmuştur. Hedef takibi hatalarının yanı sıra, sürüklenme parametresi kestirimleri ve kayma hesabının filtre performansına etkisi incelenmektedir. Çeşitli simülasyonlar yapılarak her filtre ile elde edilen kestirim sonuçları ayrıntılı olarak tartışılmaktadır.

Anahtar Kelimeler: Balistik Hedef Takibi, Dönü ile Dengelenmiş Mermi, Kısa Menzilli Balistik Hedef, Kalman Filtresi, Parçacık Filtresi

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 thesis.

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.

I also would like to thank my friend Tamer Akça for his valuable ideas and help throughout this study.

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

TABLE OF CONTENTS

CHAPTER

LIST OF FIGURES

FIGURES

LIST OF TABLES

TABLES

CHAPTER 1

INTRODUCTION

The term “ballistic targets” includes a variety of projectiles and missiles which are able to perform exo-atmospheric flight. However, in this work, the trajectories of short-range ballistic targets which have endo-atmospheric flight are in concern in order to be more specific and simple.

The trajectories of short-range ballistic targets are quite predictable especially when they are compared to those of powered vehicles such as fighter aircrafts. Ballistic target tracking has drawn attention in recent years due to the need of controlling own projectiles and missiles after launching or estimating the trajectories of hostile projectiles and missiles. Accordingly, the focus of this thesis is to present methods for estimating the trajectories of short-range ballistic targets and evaluate the tracking performances of these methods.

In this chapter, first, the coordinate systems used in ballistic target tracking and general knowledge about ballistic targets will be given. Then, the motion model used in the thesis will be explained briefly. Finally, the outline of the thesis will be presented.

1.1. Coordinate Systems

Three widespread coordinate systems used in ballistic target tracking are depicted in Figure 1.1. These are called Earth Centered Inertial, Earth-Centered Earth Fixed and East North Up Coordinate Systems [1].

1. The Earth Centered Inertial Coordinate System (ECI-CS, $Ox_I\, y_I\, z_I$) is a right handed coordinate system which is fixed with respect to fixed stars. The origin is the Earth's center given as O in Figure 1.1. Ox_I points in the vernal equinox direction, Oz_I points in the North Pole N and $Ox_I\, y_I$ plane coincides with the Earth's equatorial plane. ECI-CS does not rotate with the Earth.
2. The Earth-Centered Earth Fixed Coordinate System (ECEF-CS, $Ox_Fy_Fz_F$) is also a right handed coordinate system and has its origin at the Earth center O. Ox_F points to the prime meridian direction, and Ox_Fy_F plane coincides with the Earth's equatorial plane. However, apart from ECI-CS, ECEF-CS rotates with the Earth around its spin axis $Oz_F= Oz_I$.
3. The final coordinate system given in Figure 1.1 is the East North Up Coordinate System (ENU-CS, $Ox_Sy_Sz_S$). O_S is the origin and it is located at some point on the Earth. O_Sz_S is normal to the Earth's reference ellipsoid. Therefore, unless spherical Earth model is used, z_SO_S does not point to Earth's center O. $O_S\mathrm{x}_\mathrm{S}$ and $O_S\mathrm{y}_\mathrm{S}$ points to East and North respectively and z_S is normal to $O_S\mathrm{x}_\mathrm{S}\mathrm{y}_\mathrm{S}$ plane. $Ø$ is the angle between O_Sz_S and the Earth's equatorial plane and it represents the geodetic latitude.

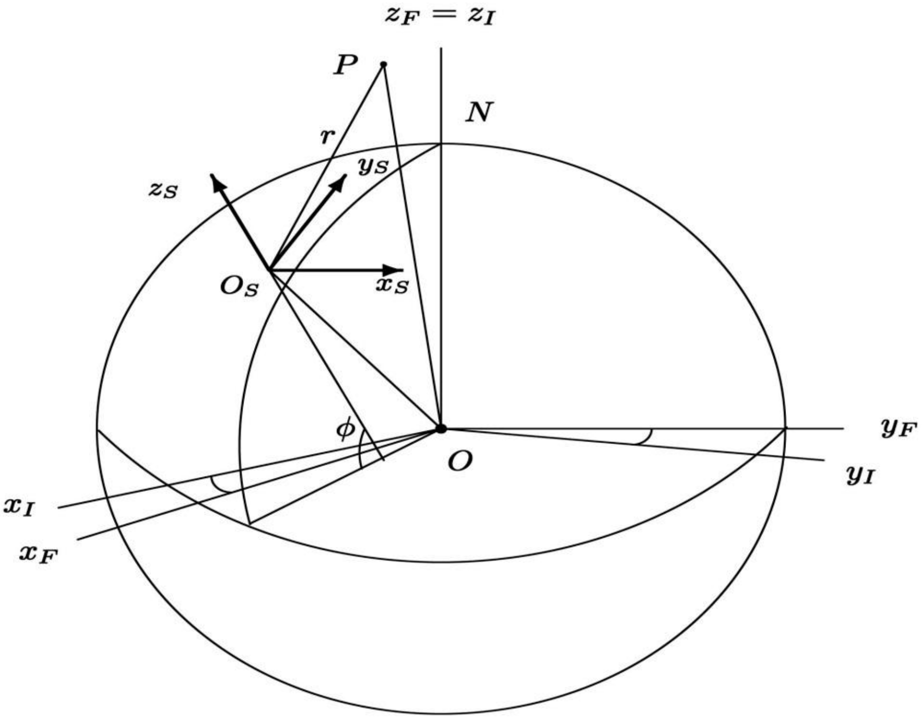

Figure 1.1: Coordinate Systems (Figure is adapted from [1])

When the point O_S is assumed to be the point where the sensor is located, the vector $\vec{r}$ appearing in Figure 1.1 can be explained as follows.

$\vec{r} = O_sP$, the vector from the sensor to the target where P is the target position.

In this study ENU coordinate system is used. Therefore, the coordinates of the target position is the projection of $\vec{r}$ onto x_s, y_s, and z_s.

1.2. Ballistic Targets

The lexical meaning of ballistics is the science of mechanics that deals with the flight, behavior and effects of projectiles. The trajectory of the ballistic targets is determined significantly by the characteristics that are specific to target type [1]. This is why these targets are called as the ballistic targets. Ballistic missiles, mortars and projectiles possess more certain motion than many other types of maneuvering targets such as aircrafts and agile missiles.

If it is assumed that the target is tracked in an inertial coordinate system such as ECI-CS, the total acceleration of the target, a, is composed of four major elements due to four major forces acting on it.

$$a = a_T + a_G + a_D + a_L \tag{1-1}$$

where a_T, a_G, a_D and a_L are acceleration due to thrust, gravity, drag force and lift force respectively. At some regions of the trajectory these components of the acceleration can be omitted or ignored. In addition, there are some other forces acting on ballistic targets such as the Coriolis and centrifugal forces which are not mentioned in (1-1). These forces should be taken into consideration if very accurate results are aimed or if the target's trajectory includes a long exo-atmospheric flight.

Trajectories of projectiles and/or missiles are chronologically divided into three phases: Boost Phase, Ballistic Flight and Reentry.

1.2.1. Boost Phase

Boost phase is the phase just after the target is launched. If the ballistic target in concern is a missile, it is exposed to a considerable thrust in this phase. Thrust can be varying which makes the trajectory estimation rather complicated. The total acceleration in boost phase can be written as follows.

$$a = a_T + a_D + a_G \tag{1-2}$$

where a_T and a_D are large in magnitude relative to a_G, the gravity. In this representation, $a_{L,}$ is omitted due to its relatively small magnitude. Furthermore, Earth rotation can also be omitted in this phase due to its small duration. Therefore, the Coriolis effect and the centrifugal force are ignored in Eq. (**1**-**2**). As a result, the inertial frame (ECI-CS) and Earth-Centered Earth Fixed Coordinate System (ECEF-CS) are practically the same.

1.2.2. Ballistic Flight

If the target is in an exo-atmospheric flight, the only force acting among the ones in Eq. (1-1) would be the gravity. However, it should be noted that if the motion is examined in a non-inertial frame such as ENU-CS, and the projectile travels for a long period of time, accelerations due to Coriolis effect and the centrifugal force should be taken into account.

Gravity models can be classified according to their approaches to modeling the Earth. The most common and well-known model is known as the *Flat-Earth Model*. As its name implies Flat-Earth Model considers the Earth as if it is flat and non-rotating which makes it the simplest possible model. For a target observed in ENU-CS the gravity vector is given below.

$$a_G = \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} \text{, where g = 9.78m/s}^2 \qquad (1\text{-}3)$$

In Eq. (1-3), the gravity is considered to be constant for different altitudes or the change in the magnitude of the gravity with the altitude is negligible. This assumption is quite feasible. For example, consider the gravity change at a height of 10000 meters. The radius of the Earth is approximately 6400km. Let the magnitude of gravity at the point of concern be 9.78m/s^2. According to Newton’s 2nd law, the gravity is inversely proportional to the distance to Earth’s center. Then, the theoretical fall in the magnitude of gravity at an altitude of 10000 meters is calculated as 0.31% in Eq. (1-4).

$$a_{G_{10000 meters}} \cong \left(\frac{r_{earth}}{r_{earth} + 10000} \right)^2 a_G = \left(\frac{6400x10^3}{6400x10^3 + 10000} \right)^2 9.78 \cong 9.75\text{m/s}^2 \qquad (1\text{-}4)$$

Note also that the magnitude of gravity is not identical on different locations on the Earth due to its non-homogeneous matter distribution. However, this change is also negligible similar to the changes in gravity due to the altitude.

Flat-Earth model is quite sufficient for most of the applications like tracking ballistic targets having endo-atmospheric flight. Nonetheless, for better results one can assume the Earth as a sphere and make use of Spherical Earth model. In this case the gravity model is called as the *Spherical Earth Model*.

If the Earth and the targets are assumed to be spherically symmetric and to possess even distributions of their masses, the gravity can be calculated according to Newton's law of universal gravitation,

$$a_G = -\frac{\mu}{\| r \|^3} r \tag{1-5}$$

where μ is the standard gravitational parameter.

For more precise calculations Ellipsoidal Earth or WGS-84 model can also be used. However, for the purpose of tracking ballistic targets like cannons or mortars, Flat Earth Model is adequate. Therefore, it is preferred to use Flat Earth Model in this study.

1.2.3. Reentry

The two main forces acting on the targets in reentry phase are the gravity and the drag force.

$$a = a_D + a_G \tag{1-6}$$

The drag force induced acceleration is given by,

$$a_D = -\frac{1}{2} \rho(h) \alpha \| v \| v \tag{1-7}$$

where ρ is the air density, h is the target altitude, α is the drag parameter and v is the target velocity.

The air density is generally estimated with an exponential function.

$$\rho(h) = c_1 e^{-c_2 h} \tag{1-8}$$

where c_1=1.227, c_2=1.0931x10^{-4} and ρ is given in kg/m^3.

1.3. Motion Model Used In the Thesis

In this study, projectiles in endo-atmospheric flight are studied such as howitzers, mortars and cannons which do not experience any thrust in any part of their trajectory and these targets are informally called as short-range ballistic targets. For this reason, some simplifications and assumptions are made in the models used. As a result, the thrust force induced acceleration a_T in Eq. (1-1) disappears. For the gravity of Earth, Flat Earth model is used and its magnitude is accepted to be 9.78m/s^2. The target is assumed to be in a simplified boost phase. Therefore, ignoring the lift force which is relatively small when compared to drag force, the projectile is subjected to gravity and drag force and Eq. (1-1) simplifies to the following.

$$a = a_G + a_D \tag{1-9}$$

The dynamics of the target and the measurement model used are explained in detail in Chapter 2 and 3.

1.4. Outline of the Thesis

The outline of the thesis is as follows.

In Chapter 2, the aerodynamic forces and moments acting on ballistic targets are explained and the major trajectory models for ballistic targets in the literature are presented. In Chapter 3, tracking methodology, that is the motion and the measurement model used in the thesis, is described. In Chapter 4, the tracking filters used in the simulations are given. One cycle of the algorithms of the filters are presented step by step. In Chapter 5, simulation results of the filters are presented and their performances are discussed. In Chapter 6, the thesis is summarized and conclusions about the results of the simulations are presented.

CHAPTER 2

DYNAMICS OF BALLISTIC TARGETS

The detailed information about the dynamics of ballistic targets is not open to general public. The few sources that exist in the literature are used in the modeling of the motion of a ballistic target in this study. Therefore, the information given in part *2.1 Aerodynamic Forces and Moments* is mainly based on the references [2], [3] and [4].

2.1. Aerodynamic Forces and Moments

On a large scale, the trajectory of a symmetric, spinning projectile is determined by the gravity of Earth, aerodynamic forces and aerodynamic moments if it is not exposed to any thrust. Gravity of Earth is more deterministic and easier to estimate when it is compared to the aerodynamic forces and moments. As it is illustrated in Eq. (1-4), especially if the projectile will not perform any exo-atmospheric flight, the variation in the magnitude of the gravity can be ignored. Therefore, the major uncertainty in estimating the trajectory of a ballistic target is due to the aerodynamic forces and moments acting on it. Some of the significant ones of these forces and moments are explained below.

2.1.1. Drag Force

Drag force can be simply defined as the air resistance. Therefore, it is in the opposite direction to the projectile’s velocity vector.

Drag force is given by the following equation.

$$Drag\ Force = -\frac{1}{2}\rho S C_D V\vec{V} \qquad (2\text{-}1)$$

where ρ is the air density, $S = \frac{\pi d^2}{4}$ is projectile reference area, C$_D$ is drag coefficient, V is the scalar magnitude of the projectile velocity and $\vec{V}$ is the vector of the projectile velocity. SC_D multiplication can also be denoted as α which is called the drag parameter given in Eq. (1-7).

The drag force is directly related to the velocity of the projectile relative to air. Therefore, the effect of wind on drag force can be inserted as follows.

$$Drag\ Force = -\frac{1}{2}\rho S C_D \tilde{V}(\vec{V} - \vec{W}) \qquad (2\text{-}2)$$

where $\vec{W}$ is the vector of the wind velocity and $\tilde{V} = |\vec{V} - \vec{W}|$.

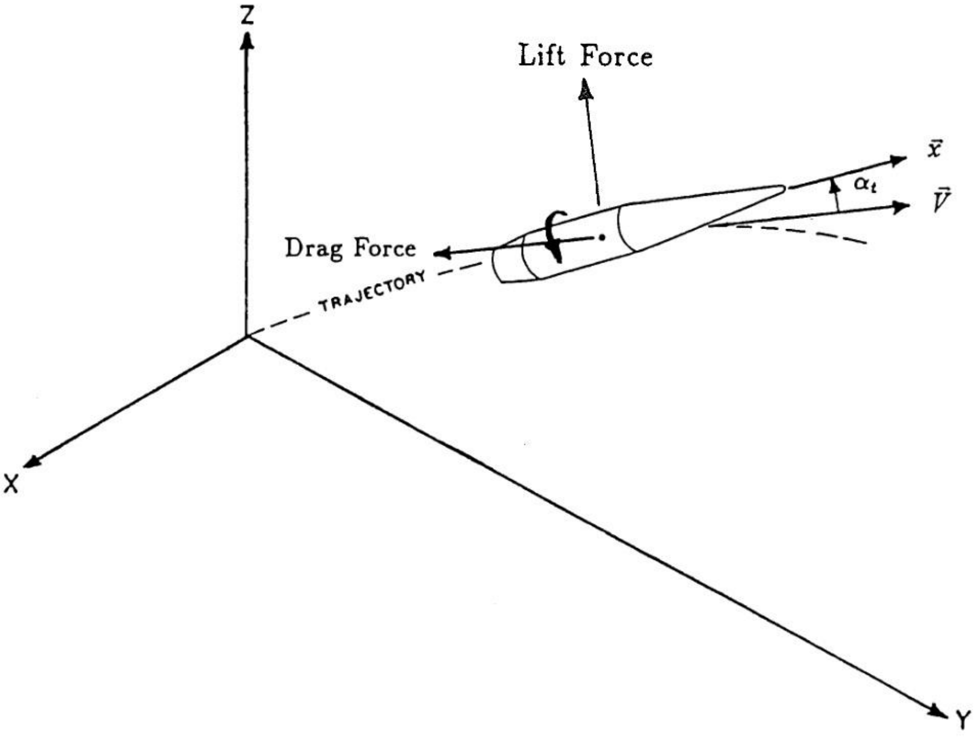

Figure 2.1: Drag & Lift Force (Figure is adapted from [2])

The drag coefficient, C_D, varies with the Mach number of the projectile. Therefore, drag coefficient versus Mach number curves are of great importance for ballisticians. These curves are calculated according to projectile geometry and the measurements made in wind tunnels, firing and spark photography ranges.

As it can be seen in Figure 2.1, the velocity vector $\vec{V}$ and the direction in which the projectile's tip points are not the same. The angle α_t is called as the total yaw angle or angle of repose and it is the resultant of the angle of attack (pitch) and angle of sideslip (yaw). Due to this total yaw angle the drag coefficient can be further decomposed as:

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

where C_{D_0} is called the zero-yaw drag coefficient, $C_{D_{\delta^2}}$ is called the yaw drag coefficient and $\delta=\sin \alpha_t$. Unless very exact results are aimed, ignoring total yaw angle in drag coefficient calculations is reasonable.

Furthermore, the total yaw angle α_t is not constant due to the epicyclic motion that spin-stabilized projectiles perform. Epicyclic motion is the helical path of the nose of the projectile around the velocity vector $\vec{V}$.

A typical drag coefficient versus Mach number curve is given in Figure 2.2. Note that at Mach 1 there is a highly nonlinear behavior of the drag coefficient which is important when tracking high velocity projectiles.

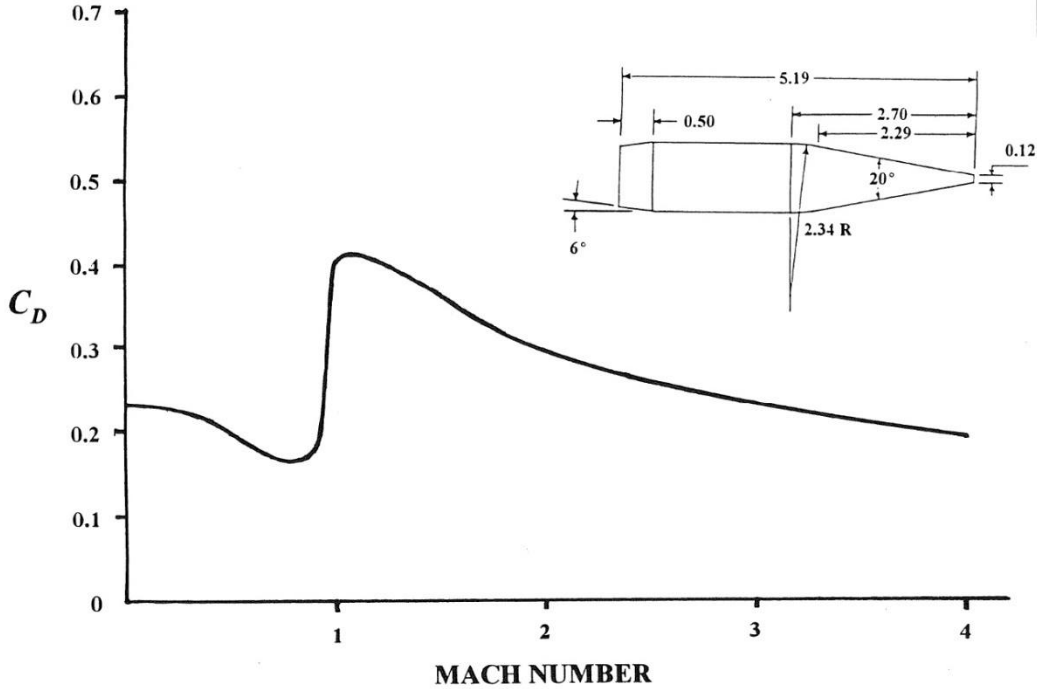

Figure 2.2: Drag Coefficient vs. Mach Number (Figure is adapted from [2])

2.1.2. Lift Force

Lift force is perpendicular to the trajectory and tries to pull the projectile in the direction its nose is pointing. The equation that describes lift force is the following.

$$Lift\ Force = \frac{1}{2}\rho S C_{L_\alpha} [\vec{V} \text{ x } (\vec{x} \text{ x } \vec{V})] \tag{2-4}$$

where C_{L_α} is the lift force coefficient and $\vec{x}$ is the unit vector in the direction the projectile's nose points. The remaining variables are previously defined.

The angle of attack of the projectile can be negative due to the epicyclic motion. For this reason, lift force need not be in upward direction as its name implies and in ballistics lift force can also be called as the Cross-Wind force or the Normal force.

2.1.3. Overturning Moment

Overturning moment is the aerodynamic moment originating from the lift force.

$$Overturning\ Moment = \frac{1}{2}\rho V^2 S d C_{M_\alpha} \left(\vec{i} \text{ x } \vec{x}\right) \tag{2-5}$$

where *d* is the projectile diameter, C_{M_α} is the overturning moment coefficient and $\vec{i}$ is the unit vector in the direction of $\vec{V}$.

$$\vec{i} = \left(\frac{\vec{V}}{V}\right) \tag{2-6}$$

If the projectile is non-spinning, fins are added to the tail of the projectile. In this way, overturning moment is controlled by the tail lift induced by the fins.

2.1.4. Spin Damping Moment

Spin damping moment tries to reduce the axial spin continuously and it is given as the following equation.

$$Spin\ Damping\ Moment = \frac{1}{2}\rho V^2 S d \left(\frac{pd}{V}\right) C_{L_p} \vec{x} \tag{2-7}$$

where C_{L_p} is the spin damping moment coefficient and *p* is the axial spin rate usually measured in radians/sec. and positive for right-hand spin.

Spin damping moment is in the direction of $\vec{x}$. Therefore, according to the convention of the vectors given in Figure 2.1, C_{L_p} should be negative since spin damping moment tries to decrease the axial spin.

2.1.5. Magnus Force

Spinning bodies experience Magnus force due to Magnus Effect named after the German physicist Heinrich Magnus. As an object spins in a viscous fluid such as air, a boundary layer is formed around itself [5]. This boundary layer of air possesses different velocities on the forward-moving side (Point A) and backward-moving side (Point B) (See Figure 2.3). The velocity of boundary layer of air due to spin is added to the velocity of the wind, V_w, on the backward-moving side and subtracted on the forward-moving side. According to Bernoulli's theorem, fluid pressure is small at the points where fluid speed is high. Therefore, the difference in velocity of air at points A and B results in a pressure difference and produces the Magnus force from the forward-moving side to the backward-moving side. The direction of this force is from A to B as it is shown in Figure 2.3.

$$Magnus\ Force = \frac{1}{2}\rho V^2 S\left(\frac{pd}{V}\right)C_{N_{p\alpha}}\left(\vec{i}\ \mathrm{x}\ \vec{x}\right) \tag{2-8}$$

where $C_{N_{p\alpha}}$ is the Magnus force coefficient.

$C_{N_{P\alpha}}$ is negative according to the direction of the Magnus Force vector given in Figure 2.4.

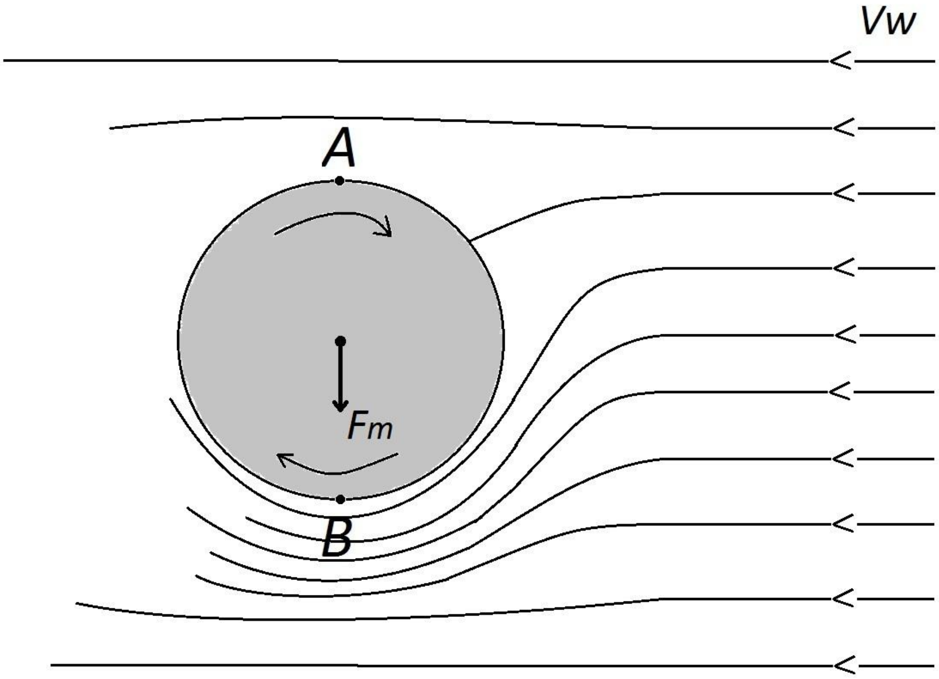

Figure 2.3: Spinning Body Moving in Air

It is obvious from Eq. (2-8) that if the total angle of yaw is zero or the projectile does not spin, the Magnus force will be zero.

The magnitude of the Magnus force acting on spinning projectiles is much smaller than that of acting on low-velocity spinning bodies. Thus, it can be neglected in exterior ballistics calculations. However, Magnus force results in Magnus moment which has great influence on the stability of spin-stabilized projectiles.

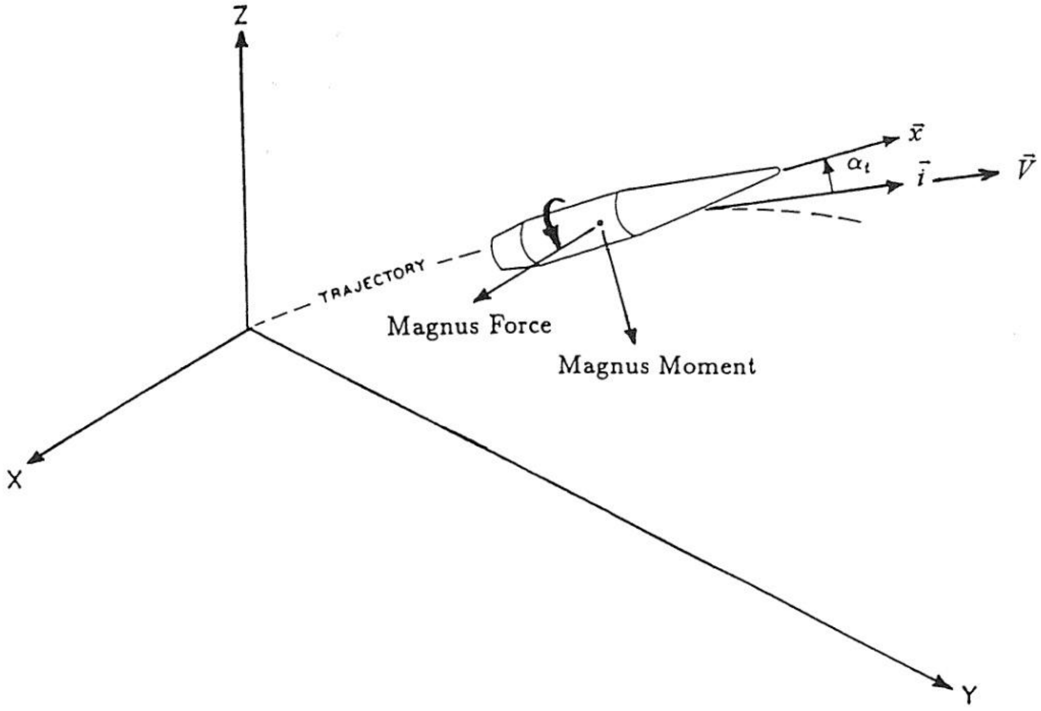

Figure 2.4: Magnus Force and Magnus Moment (Figure is adapted from [2])

2.1.6. Magnus Moment

The Magnus moment is defined as the following equation:

$$Magnus\ Moment = \frac{1}{2}\rho V^2 Sd\left(\frac{pd}{V}\right)C_{M_{P\alpha}}[\vec{x} \text{ x } (\vec{i} \text{ x } \vec{x})] \tag{2-9}$$

where $C_{M_{P\alpha}}$ is the Magnus moment coefficient.

Similar to center of gravity, a point in the projectile can be defined as the center of pressure. This is the point to which the observed force should be applied in order to obtain the observed moment.

The center of pressure and center of gravity locations on projectiles are different. It should also be noted that lift force and Magnus force centers of pressure are also not the same point. Generally, lift force center of pressure lies between the nose and center of gravity, while Magnus force center of pressure is located between the tail and center of gravity. According to the convention of vectors in Figure 2.4, Magnus force is negative. Thus, since Magnus force center of pressure is close to the tail, Magnus moment vector given in the figure shows the correct direction of the Magnus moment.

Looking from the aft, the tip of the projectile generally points to right during the epicyclic motion of the projectile. This situation results in a drift to right at the impact point in long range artillery fire. The lift force drags the projectile to right since its nose usually points to right (See Figure 2.4 – Projectile rotating clockwise). For a counter-clockwise rotating projectile the same amount of drift will be observed to the left. According to [4], the drift of the projectile can be simply approximated by the following formula.

$$drift = d_1 (ToF)^2 \tag{2-10}$$

where ToF is the total time of flight and d_1 is a constant parameter for the projectile in consideration that is ranging between 0.1 and 0.12.

2.1.7. Pitch Damping Force

The pitch damping force is defined by the following equation.

$$Pitch\ Damping\ Force = \frac{1}{2}\rho VSdC_{N_q}\left(\frac{d\vec{x}}{dt}\right) + \frac{1}{2}\rho VSdC_{N_{\dot{\alpha}}}\left(\frac{d\vec{x}}{dt} - \frac{d\vec{i}}{dt}\right) \tag{2-11}$$

where C_{N_q} and $C_{N_{\dot{\alpha}}}$ are pitch damping force coefficients due to q_t and $\dot{\alpha}_t$.

The pitch damping force vector given in Figure 2.5 is true for positive pitching angular velocity. The parameter q_t is called the total transverse angular velocity.

$$q_t = \sqrt{q^2 + r^2} \tag{2-12}$$

where q is pitching angular velocity and r is yawing angular velocity.

q_t and $\dot{\alpha}_t$ are small and close to each other in magnitude. For this reason, they are generally assumed to be the same and this assumption simplifies Eq. (2-11) as the following.

$$Pitch\ Damping\ Force = \frac{1}{2}\rho V S d (C_{N_q} + C_{N_{\dot{\alpha}}})\left(\frac{d\vec{x}}{dt}\right) \qquad (2\text{-}13)$$

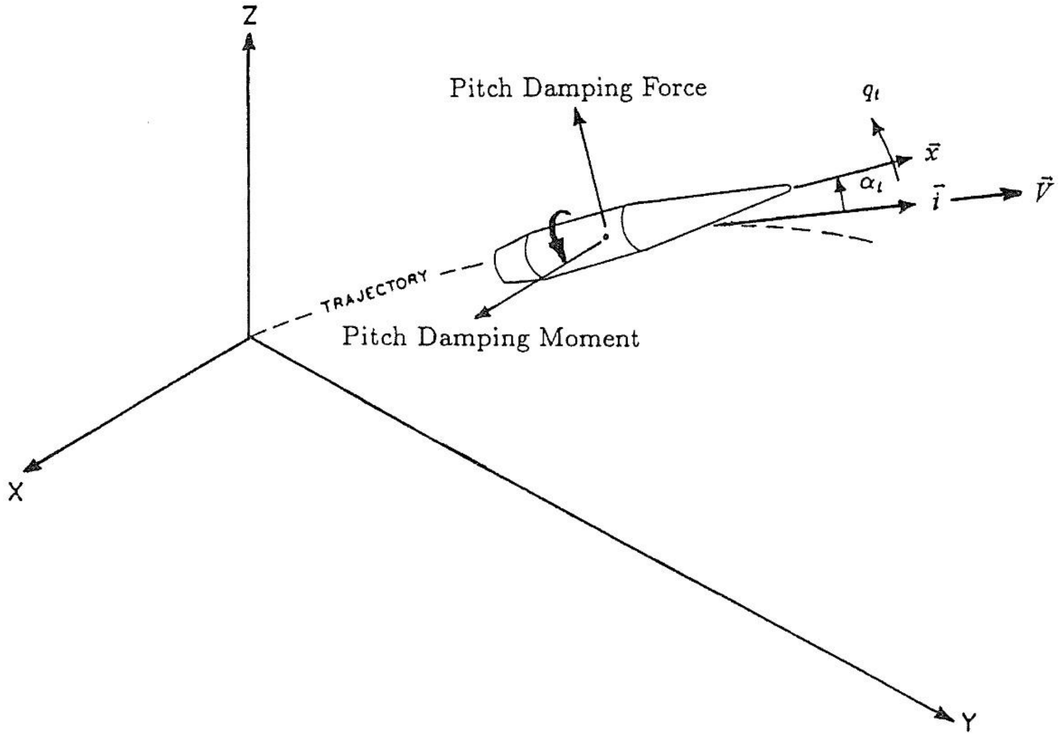

Figure 2.5: Pitch Damping Force and Pitch Damping Moment (Figure is adapted from [2])

2.1.8. Pitch Damping Moment

Pitch damping force is much smaller than the drag or the lift force and it is usually neglected. However, like the Magnus moment, pitch damping force induced pitch damping moment is crucial for the stability of spin-stabilized projectiles.

Similar to pitch damping force calculation in Eq. (2-13), assuming that q_t and $\dot{\alpha}_t$ are the same, pitch damping moment is defined as.

$$Pitch\,Damping\ Moment = \frac{1}{2}\rho V S d^2 (C_{M_q} + C_{M_{\dot{\alpha}}})\left(\vec{x} \text{ x } \frac{d\vec{x}}{dt}\right) \tag{2-14}$$

where C_{M_q} and $C_{M_{\dot{\alpha}}}$ are pitch damping moment coefficients due to q_t and $\dot{\alpha}_t$.

The sum ($C_{M_q} + C_{M_{\dot{\alpha}}}$) should be negative for the stability and this is usually the case.

2.2. The Vacuum Trajectory

According to Newton's 2nd law, the general differential equation of motion for a projectile can be defined as follows.

$$m\frac{d\vec{V}}{dt} = \Sigma F + m\vec{g} + m\vec{\Lambda} \tag{2-15}$$

where m is the mass, $\vec{V}$ is the velocity vector, F is the aerodynamic forces, $\vec{g}$ is the gravity and $\vec{\Lambda}$ is the acceleration due to Coriolis effect.

The vacuum trajectory suggests the simplest possible motion model. As its name implies it provides the trajectory of a body flying in vacuum. Hence, all the aerodynamic forces and moments are zero. It is applicable to short range firing. For this reason Coriolis acceleration can also be omitted. Consequently, Eq. (2-15) simplifies to,

$$\frac{d\vec{V}}{dt} = \vec{g} \tag{2-16}$$

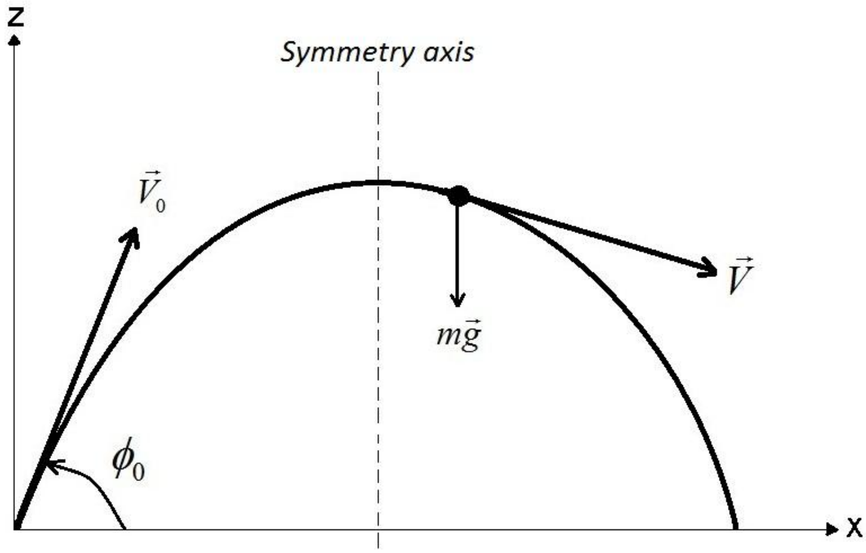

Figure 2.6: The Vacuum Trajectory

As it is seen in Figure 2.6, the trajectory is symmetric with respect to the symmetry axis passing through the apogee of the trajectory. It is obvious that the trajectory can be studied in two-dimensional coordinates. The knowledge of the initial velocity, V_0, and angle of departure, $Ø_0$, makes it possible to estimate the whole trajectory using Eq. (2-16).

2.3. The Point Mass Trajectory

The point mass trajectory, also called as the particle trajectory, is a more realistic model for the dynamics of ballistic targets than the vacuum trajectory. The term “point mass” or “particle” implies a non-spinning projectile whose complete mass is moving as a mathematical point in space.

The significant aerodynamic forces acting on ballistic targets are the drag force, the lift force and the Magnus force [2]. However, if the angle of repose is small, the lift force and the Magnus force can be neglected according to the Eqs. (2-4) and (2-8). The trajectory is shaped by the gravity and the drag force since the projectile is assumed to be a non-spinning point mass and no angle of repose exists. The Eq. (2-15) simplifies to the following.

$$m\frac{d\vec{V}}{dt} = \Sigma F_D + m\vec{g} \qquad (2\text{-}17)$$

where F_D is the drag force. The point mass trajectory with the flat-fire approximation is examined in the following section.

2.3.1. Flat-Fire Point Mass Trajectory

The flat-fire approximation holds if the trajectory is close to the x-axis everywhere. Naturally, this constraint is valid if V_x is close to $\vec{V}$ in magnitude. Thus, the main assumption in flat-fire approximation calculations is $|\vec{V}| = V_x$.

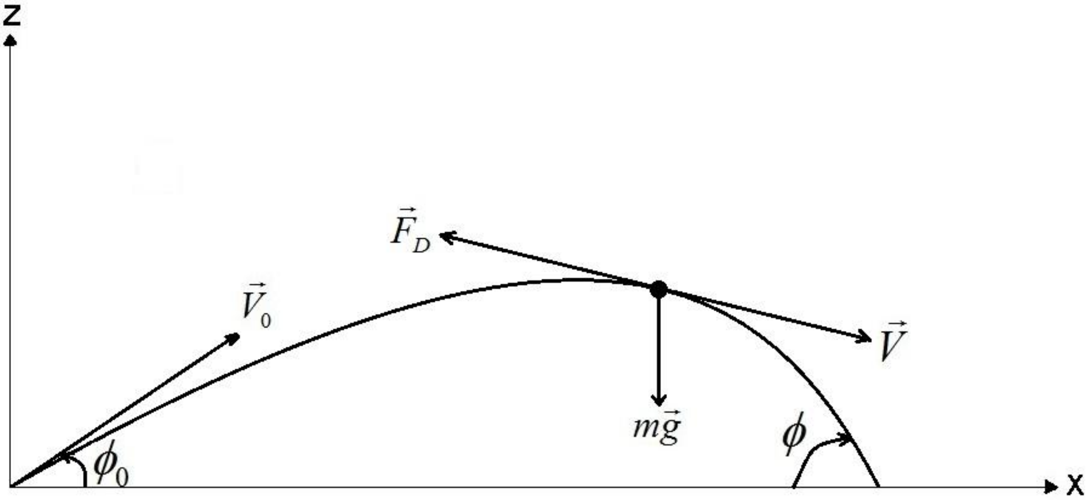

Figure 2.7: The Flat-Fire Point Mass Trajectory

Assuming that there is no velocity component other than V_x and V_z.

$$|\vec{V}| = V = \sqrt{V_x^2 + V_z^2} = V_x\sqrt{1 + (V_z / V_x)^2} \qquad (2\text{-}18)$$

If right hand side of Eq. (2-18) is expanded in binomial series the following is obtained.

$$V = V_x\left[1+\frac{1}{2}(V_z/V_x)^2-\frac{1}{8}(V_z/V_x)^4+\cdots\right] \tag{2-19}$$

It is clear from Eq. (2-19) that if $|V_z/V_x|<10^{-1}$, the magnitude of $\vec{V}$ differs from V_x by less than 0.5%. V_z/V_x ratio gives the tangent of the angle, ϕ, between the trajectory and the ground. Therefore, if $\phi < 5.7°$ everywhere on the trajectory, the flat fire approximation holds theoretically. However, in practice the angles smaller than 15 degrees is permissible [2].

2.3.2. Six-Degrees-of-Freedom (6-DOF) and Modified Point Mass Trajectories

When compared to flat-fire point mass trajectory, six-degrees-of-freedom (6-DOF) trajectory is a rather sophisticated model in which all the forces and moments mentioned in part *2.1 Aerodynamic Forces and Moments* are taken into consideration. The projectile is assumed to move in the Cartesian coordinates x, y, z and to rotate about roll, pitch and yaw axes. In other words, the motion has six degrees of freedom. If complete ballistic parameters for a projectile are present, one can obtain the most accurate possible trajectory solution using 6-DOF differential equations of motion [2].

The high frequency pitching and yawing motion of the projectile can only be calculated if small integration time steps are used. This high computation load has lead to modified point mass trajectory where epicyclic pitching and yawing motion is assumed to be slow everywhere along the trajectory. It is also observed in 6-DOF calculations that yaw of repose varies slowly along the trajectory and it can be assumed that its derivative, $\dot{\alpha}_t$, is equal to zero (See Figure 2.5 for α_t). Using these two assumptions and neglecting the relatively small pitch damping force, 6-DOF differential equations are refined and the modified point mass trajectory is obtained.

The differential equations defining 6-DOF and the modified point mass trajectories for a spinning projectile are given in Table 2-1 and Table 2-2. The modified point mass trajectory differential equations are obtained after a considerable simplification which can be found in [2].

Table 2-1: Differential equations of motion for 6-DOF trajectory (Table is adapted from [2])

6-DOF Trajectory
$\frac{d\vec{V}}{dt} = -\frac{\rho VSC_D}{2m}\vec{V} + \frac{\rho SC_{L_\alpha}}{2m}[\vec{V} \text{ x } (\vec{x} \text{ x } \vec{V})] + \frac{\rho SdC_{N_{p\alpha}}}{2m}\left(\frac{I_y}{I_x}\right)\left(\vec{h}\bullet\vec{x}\right)\left(\vec{x} \text{ x } \vec{V}\right) + \cdots$ $\cdots + \frac{\rho VSd(C_{N_q} + C_{N_{\dot{\alpha}}})}{2m}(\vec{h} \text{ x } \vec{x}) + \vec{g} + \vec{\Lambda}$
$\frac{d\vec{h}}{dt} = \frac{\rho V^2 SdC_{M_\alpha}}{2I_y}\left(\vec{i} \text{ x } \vec{x}\right) + \frac{\rho VSd^2C_{L_p}}{2I_x}\left(\vec{h}\bullet\vec{x}\right)\vec{x} + \frac{\rho Sd^2C_{M_{P\alpha}}}{2I_x}\left(\vec{h}\bullet\vec{x}\right)[\vec{V} - (\vec{V}\bullet\vec{x})\vec{x}] + \cdots$ $\cdots + \frac{\rho VSd^2(C_{M_q} + C_{M_{\dot{\alpha}}})}{2I_y}\left(\vec{x} \text{ x } \frac{d\vec{x}}{dt}\right)$

where *m* is the projectile mass, I_x is the projectile axial moment of inertia, I_y is the projectile transverse moment of inertia about any axis through the center of mass, $\vec{h}$ is the vector angular momentum divided by I_y, and $\vec{\Lambda}$ is the Coriolis acceleration vector. The remaining variables are defined previously.

Table 2-2: Differential equations of motion for the modified point mass trajectory (Table is adapted from [2])

The Modified Point Mass Trajectory
$\frac{d\vec{V}}{dt} = -\frac{\rho VSC_D}{2m}\vec{V} + \frac{\rho V^2 SC_{L_\alpha}}{2m}\vec{\alpha}_R + \frac{\rho SdC_{N_{p\alpha}}}{2m}p\left(\vec{v} \text{ x } \vec{\alpha}_R\right) + \vec{g} + \vec{\Lambda}$
$\frac{d\vec{p}}{dt} = -\frac{\rho Sd^2 V}{2I_x}pC_{L_p}$

where $\vec{\alpha}_R$ is the angle of repose calculated according to the assumptions of the modified point mass trajectory. Computation of $\vec{\alpha}_R$ can be found in [2].

CHAPTER 3

TRACKING METHODOLOGY

In order to predict the trajectory of a ballistic target, a motion model should be constructed and the predicted trajectory should be updated according to the measurements of the target and the measurement model. In this chapter, these models and the assumptions made while obtaining them are explained. The models presented will then be used in the tracking filter which will perform the trajectory estimations.

3.1. Motion Model

According to point mass approximation, the significant forces acting on the projectile are the gravity and the drag force as it is given in Eq. (2-17). However, inserting the drag force into the motion model is somewhat troublesome due to the nonlinear behavior of the drag coefficient with the Mach number of the projectile. Therefore, the drag coefficient should also be included in the state vector and should be estimated together with the trajectory. The other aerodynamic forces are neglected due to their relatively small magnitude and the nonlinear relationship between them and the angle of repose which is impossible to measure. The drift effect of the lift force is included in the model by the approximation given in Eq. (2-10).

The trajectory is estimated in Cartesian coordinates based on a flat, non-rotating Earth. Since gravity and drag force are the major forces considered, the target is assumed to be in the reentry phase or in the boost phase with no thrust.

Let x_p be the target position vector and x_v be the target velocity vector. The drag parameter α is equal to $S.C_D$ (S = projectile reference area, C_D = drag coefficient) and it is assumed to change slowly between the successive time steps of the tracking filter. Using Eq. (1-7), the state equations describing the system is obtained as follows.

$$\frac{dx_p}{dt} = x_v \tag{3-1}$$

$$\frac{dx_v}{dt} = g - \frac{1}{2}\rho\alpha \,|\, x_v \,|\, x_v \tag{3-2}$$

$$\frac{d\alpha}{dt} = 0 \tag{3-3}$$

However, using the Eqs. (3-1) through (3-3) as the motion model of the filter is not trouble-free due to the large differences in magnitudes of x_p, x_v and α. It is problematic to tune the filter parameters, to propagate covariance matrices and to perform measurement updates since $\alpha \approx 10^{-3}$ (kg/m^2) while $\| x_v \| \approx 10^2$ (m/s) and $\| x_p \| \approx 10^4$ (m). For this reason, the ballistic coefficient β is typically used in the filter instead of α [6] [16] [17].

$$\beta = \alpha^{-1} \tag{3-4}$$

Furthermore, due to the poor linearization properties of β, the following change of variable is done and $\Delta\beta$ is estimated in the filter.

$$\alpha = \beta^{-1} = \beta_0^{-1} - \beta_0^{-2}\Delta\beta \tag{3-5}$$

where β_0 is a reference ballistic coefficient and $\Delta\beta$ is the deviation of the actual value from the reference value that is modeled as a new state variable. An initial value to β_0 is assigned while starting the filter, and $\Delta\beta$ is dynamically updated. The resulting differential equations describing the system are as follows.

$$\frac{dx_p}{dt} = x_v \tag{3-6}$$

$$\frac{dx_v}{dt} = g - \frac{\rho}{2\beta_0}(1 - \frac{\Delta\beta}{\beta_0}) | x_v | x_v \tag{3-7}$$

$$\frac{d\Delta\beta}{dt} = 0 \tag{3-8}$$

Consequently, the filter is run with the following state vector, $\bar{x}$. The differential equations Eq. (3-6) through (3-8) can be rearranged and written in continuous time as it is given in Eq. (3-10).

$$\bar{x} = \begin{bmatrix} x & \dot{x} & y & \dot{y} & z & \dot{z} & \Delta\beta \end{bmatrix}^T \tag{3-9}$$

$$\frac{d\bar{x}}{dt} = \frac{d}{dt}\begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \\ z \\ \dot{z} \\ \Delta\beta \end{bmatrix} = \begin{bmatrix} \dot{x} \\ -\frac{\rho}{2\beta_0}(1-\frac{\Delta\beta}{\beta_0})\sqrt{\dot{x}^2+\dot{y}^2+\dot{z}^2}\,\dot{x} \\ \dot{y} \\ -\frac{\rho}{2\beta_0}(1-\frac{\Delta\beta}{\beta_0})\sqrt{\dot{x}^2+\dot{y}^2+\dot{z}^2}\,\dot{y} \\ \dot{z} \\ g-\frac{\rho}{2\beta_0}(1-\frac{\Delta\beta}{\beta_0})\sqrt{\dot{x}^2+\dot{y}^2+\dot{z}^2}\,\dot{z} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ u_{\dot{x}} \\ 0 \\ u_{\dot{y}} \\ 0 \\ u_{\dot{z}} \\ u_{\Delta\beta} \end{bmatrix} \tag{3-10}$$

where *x*, *y*, *z* are the position. $\dot{x}$, $\dot{y}$, $\dot{z}$ are the velocity of the target in Cartesian coordinates and the last term, u, is zero-mean white Gaussian noise vector injected for modeling the neglected forces and moments.

3.2. Discrete Time Motion Model

The motion model given in the previous section should be converted to a discrete time model since the measurements are taken in discrete times. A natural way of obtaining the discrete time motion model is to integrate the derivative of the state in between two consecutive radar measurement times. However, integration is not easy because of two reasons. First, the differential equation is not linear so integration can only be done numerically. The second reason is that it is stochastic. Solution of the stochastic differential equations is beyond the scope of this thesis. In the following part of this section, the numerical solution of the deterministic, (i.e., the process noise is assumed to be zero) continuous time state equation will be given. This will also serve as the 'predicted' state in the filtering application. Consideration of the process noise will later be described.

3.2.1. Prediction of the State with Discrete Time Model

Since the motion model is nonlinear, the discrete time conversion step will correspond to the solution of the nonlinear differential equation given in Eq. (3-10). In this study, 4th order Runge-Kutta method is used to iterate the state in time. The procedure explained below is the application of 4th order Runge-Kutta method to our problem.

Let the current time be k. Then, $\overline{x}_{k|k-1}$ will be predicted, i.e., its expected value is calculated using the following equations.

$$\overline{x}_{k|k-1} = Fx_{k-1} + K\frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{3-11}$$

$$F = \begin{bmatrix} 1 & \Delta t & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & \Delta t & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \Delta t & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad K = \begin{bmatrix} \frac{\Delta t^2}{2} & 0 & 0 \\ \Delta t & 0 & 0 \\ 0 & \frac{\Delta t^2}{2} & 0 \\ 0 & \Delta t & 0 \\ 0 & 0 & \frac{\Delta t^2}{2} \\ 0 & 0 & \Delta t \\ 0 & 0 & 0 \end{bmatrix} \tag{3-12}$$

$$F_{half} = \begin{bmatrix} 1 & \frac{\Delta t}{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & \frac{\Delta t}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \frac{\Delta t}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad K_{half} = \begin{bmatrix} \frac{\Delta t^2}{8} & 0 & 0 \\ \frac{\Delta t}{2} & 0 & 0 \\ 0 & \frac{\Delta t^2}{8} & 0 \\ 0 & \frac{\Delta t}{2} & 0 \\ 0 & 0 & \frac{\Delta t^2}{8} \\ 0 & 0 & \frac{\Delta t}{2} \\ 0 & 0 & 0 \end{bmatrix} \tag{3-13}$$

$$k_1 = \begin{bmatrix} 0 \\ 0 \\ -9.81 \end{bmatrix} - \frac{c_1 e^{-c_2 x_k^-(5)}}{2\beta_0}(1 - \frac{x_k^-(7)}{\beta_0})(\sqrt{x_k^-(2)^2 + x_k^-(4)^2 + x_k^-(6)^2}) \begin{bmatrix} x_k^-(2) \\ x_k^-(4) \\ x_k^-(6) \end{bmatrix} \tag{3-14}$$

$$x_{m_1} = F_{half} x_k^- + K_{half} k_1; \qquad x_k^- = x_{k-1} \tag{3-15}$$

$$k_2 = \begin{bmatrix} 0 \\ 0 \\ -9.81 \end{bmatrix} - \frac{c_1 e^{-c_2 x_{m_1}(5)}}{2\beta_0}(1 - \frac{x_{m_1}(7)}{\beta_0})(\sqrt{x_{m_1}(2)^2 + x_{m_1}(4)^2 + x_{m_1}(6)^2}) \begin{bmatrix} x_{m_1}(2) \\ x_{m_1}(4) \\ x_{m_1}(6) \end{bmatrix} \tag{3-16}$$

$$x_{m_2} = F_{half} x_k^- + K_{half} k_2 \tag{3-17}$$

$$k_3 = \begin{bmatrix} 0 \\ 0 \\ -9.81 \end{bmatrix} - \frac{c_1 e^{-c_2 x_{m_2}(5)}}{2\beta_0}(1 - \frac{x_{m_2}(7)}{\beta_0})(\sqrt{x_{m_2}(2)^2 + x_{m_2}(4)^2 + x_{m_2}(6)^2}) \begin{bmatrix} x_{m_2}(2) \\ x_{m_2}(4) \\ x_{m_2}(6) \end{bmatrix} \tag{3-18}$$

$$x_{m_3} = F x_k^- + K k_3 \tag{3-19}$$

$$k_4 = \begin{bmatrix} 0 \\ 0 \\ -9.81 \end{bmatrix} - \frac{c_1 e^{-c_2 x_{m_3}(5)}}{2\beta_0}(1 - \frac{x_{m_3}(7)}{\beta_0})(\sqrt{x_{m_3}(2)^2 + x_{m_3}(4)^2 + x_{m_3}(6)^2}) \begin{bmatrix} x_{m_3}(2) \\ x_{m_3}(4) \\ x_{m_3}(6) \end{bmatrix} \tag{3-20}$$

k_1 is the slope at the beginning of Δt, that is, the result of Eq. (3-10) when the noise is omitted.

k_2 is the slope at the middle of Δt, which is calculated using Euler's method with slope k_1.

k_3 is the slope at the middle of Δt, which is calculated using Euler's method with slope k_2.

k_4 is the slope at the end of Δt, which is calculated using Euler's method with slope k_3.

The drift of the projectile is also calculated in every time step according to Eq. (2-10) where d_1 is set to 0.11. For this reason, the total time of flight of the projectile is updated and kept in every time step.

3.2.2. Computation of the Covariance Matrix of the State and Discrete Time Process Noise

The angle of repose of the projectile varies slowly with time as it is observed from the trajectories obtained in real world [2]. Therefore, the noise vector in Eq. (3-10) i.e. the neglected forces and moments, which are closely related to the angle of repose, also vary slowly with time. Besides, the sampling period is on the order of milliseconds for projectile tracking radars. Thus, it is assumed that the 'noise' and the Jacobian matrix F of Eq. (3-10) is constant during the sampling period and the covariance matrices are calculated under these assumptions.

The covariance matrix of the state can be obtained from the sigma points if unscented Kalman filter is used. For the extended Kalman filter case, however, the Jacobian of the differential state equations (Eq. (3-10)) is calculated and the covariance matrix of the state is updated as follows.

$$\mathrm{F}_{j,j'} = \frac{\partial f_j(x)}{\partial x_{j'}}\Bigg|_{\bar{x}=x_{k-1}} \tag{3-21}$$

$$\Phi_{k-1} = e^{F\Delta t} \tag{3-22}$$

$$P_{k|k-1} = \Phi_{k-1} P_{k-1|k-1} \Phi_{k-1}^{T} + Q_{k-1} \tag{3-23}$$

where Φ_{k-1} is the state transition matrix, $P_{k-1|k-1}$ is the state covariance matrix at time *k-1*, Q_{k-1} is the covariance matrix of the process noise which is obtained by the method described below. Note that Q_{k-1} is also used in the filters other than the extended Kalman filter whose algorithms are given in Chapter 4.

As explained in the previous section, 4th order Runge-Kutta method solves the nonlinear deterministic differential equation in the time interval [k-1, k]. Let Eq. (3-11) be written in the following form.

$$x_k = \Phi_{k-1} x_{k-1} + w_{k-1} \tag{3-24}$$

where w_{k-1} is a white Gaussian noise. The probability density function of w_{k-1} can be obtained by calculating its mean and the covariance matrix. It is clear that the mean is zero. However, the computation of the covariance of w_{k-1} is not trivial. According to the continuous-time differential equation Eq. (3-10), the following integral should be evaluated in order to determine the covariance matrix Q_{k-1}.

$$\begin{aligned} Q_{k-1} &= \mathrm{cov}(w_{k-1}) \\ &= E\{[\int_{t_{k-1}}^{t_k} \Phi(t_k,\xi) G(\xi) u(\xi) d\xi][\int_{t_{k-1}}^{t_k} \Phi(t_k,\xi') G(\xi') u(\xi') d\xi']^T\} \end{aligned} \tag{3-25}$$

where u is the noise vector which is the last term in Eq. (3-10) and G is the matrix given in Eq. (3-30). Since the noise is assumed to be white-Gaussian, Eq. (3-25) simplifies to Eq. (3-27) given below.

$$E\{u(\xi) u^T(\xi')\} = 0 \qquad if\, \xi \neq \xi' \tag{3-26}$$

$$Q_{k-1} = \int_{t_{k-1}}^{t_k} \Phi(t_k,\xi) G(\xi) E\{u(\xi) u^T(\xi)\} G^T(\xi) \Phi^T(t_k,\xi) d\xi \tag{3-27}$$

According to [7] and [8], a systematic way of computing Eq. (3-27) is to compute the matrix exponential of the following matrix *A*.

$$A = \begin{bmatrix} -\mathrm{F} & \vdots & GUG^T \\ \cdots & \vdots & \cdots \\ 0 & \vdots & \mathrm{F}^T \end{bmatrix} \Delta t \tag{3-28}$$

$$U = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 100 \end{bmatrix} \tag{3-29}$$

$$G = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3-30}$$

where *U* is the power spectral density matrix for u and F is the Jacobian of Eq. (3-10). The derivative of the position is equal to the velocity. Therefore, the noise vector u contains zeros which imply that the position of the target is modeled perfectly and there is no need for the noise (See Eq. (3-10)). For this reason, the diagonal elements of *U* which are related to the position of the target are zero. Moreover, the diagonal element of *U* which corresponds to $\Delta\beta$ is set to 100 because of the relatively high uncertainty in the drag parameter.

Let the matrix *B* is defined as the matrix exponential of *A* and it is equal to the following matrix [7].

$$B = e^{A} = \begin{bmatrix} & \vdots & \Phi_{k-1}^{-1} Q_{k-1} \\ \cdots & \vdots & \cdots \\ 0 & \vdots & \Phi_{k-1}^{T} \end{bmatrix} \tag{3-31}$$

Then, the covariance matrix of the noise is calculated by using the upper-right and lower-right partitions of *B*.

3.3. Measurement Model

Measurements are assumed to be taken by radar as range, bearing and elevation. Hence, they are related to the states by the following equations.

$$r = \sqrt{x^2 + y^2 + z^2} + v_r \tag{3-32}$$

$$\theta = \tan^{-1}\left(\frac{y}{x}\right) + v_\theta \tag{3-33}$$

$$\phi = \sin^{-1}\left(\frac{z}{\sqrt{x^2 + y^2 + z^2}}\right) + v_\phi \tag{3-34}$$

where r, θ and ϕ are range, bearing and elevation measurements of the radar. v_r, v_θ and v_ϕ are radar measurement errors in spherical coordinates with standard deviations σ_r, σ_θ and σ_ϕ. The errors are assumed to be white zero-mean Gaussian.

The measurements are in spherical coordinates while the state vector is in Cartesian coordinates. Therefore, they are not linearly related to each other as it is observed from the equations above and nonlinear measurement update should be done by the tracking filter. However, extended Kalman filter type linearization techniques produce bias due to measurement nonlinearities. To eliminate the bias, the unbiased converted measurement method which is a popular technique for tracking in Cartesian coordinates is used [9].

3.3.1. Unbiased Converted Measurements in 2D

The basic idea of unbiased converted measurements is explained in 2D in detail since it is much easier to understand. The results for the 3D case are also given in the succeeding section. [9]

Let the measured range and bearing be r_m and θ_m.

$$r_m = r + v_r \tag{3-35}$$

$$\theta_m = \theta + v_\theta \tag{3-36}$$

where r and θ are the true range and bearing while v_r and v_θ are zero-mean measurement errors with standard deviations σ_r and σ_θ.

$$x_m = r_m \cos\theta_m \tag{3-37}$$

$$y_m = r_m \sin\theta_m \tag{3-38}$$

where x_m and y_m are the classical converted measurements which are biased due to the nonlinear transformation. Taking the expectation of x_m, we have.

$$\begin{aligned} E\{x_m\} &= E\{(r+v_r)\cos(\theta+v_\theta)\} \\ &= E\{(r+v_r)(\cos\theta\cos v_\theta - \sin\theta\sin v_\theta)\} \end{aligned} \tag{3-39}$$

Assuming that v_r and v_θ are independent random variables and have symmetric probability density functions about v=0.

$$E\{v_r\} = 0. \qquad E\{\sin v_\theta\} = 0 \tag{3-40}$$

$$\begin{aligned} E\{x_m\} &= E\{r\cos\theta\cos v_\theta\} \\ &= \lambda_\theta r\cos\theta \end{aligned} \tag{3-41}$$

where $\lambda_\theta = E\{\cos v_\theta\}$.

Similarly,

$$E\{y_m\} = \lambda_\theta r \sin\theta \qquad (3\text{-}42)$$

However, it is clear that $\lambda_\theta \neq 1$ and Eqs. (3-37) and (3-38) are biased. If it is assumed that the measurement errors are independent and possess symmetric probability density functions about *v*=0, the unbiased converted measurements are.

$$x_m^u = \lambda_\theta^{-1} r_m \cos\theta_m \qquad (3\text{-}43)$$

$$y_m^u = \lambda_\theta^{-1} r_m \sin\theta_m \qquad (3\text{-}44)$$

On the other hand, finding the uncertainty in x_m and y_m is a relatively more difficult problem. In practice, the measurement errors are on the order of milliradians, so it may be thought that bias removal is not crucial. Indeed, determining the covariance matrix properly for x_m and y_m is more important than their being biased. For instance, let the range and bearing of a target be 5000 meters and 30° which are measured by a radar that has a 1σ measurement error of 2mrads. Then, the classical and unbiased converted measurements are the following.

$$x_m = r_m \cos\theta_m = 5000\cos(30) = 4330.127\text{m} \qquad (3\text{-}45)$$

$$y_m = r_m \sin\theta_m = 5000\sin(30) = 2500\text{m} \qquad (3\text{-}46)$$

$$x_m^u = \lambda_\theta^{-1} r_m \cos\theta_m = 4330.135\text{m} \qquad (3\text{-}47)$$

$$y_m^u = \lambda_\theta^{-1} r_m \sin\theta_m = 2500.005\text{m} \qquad (3\text{-}48)$$

It is clear that the difference between the classical and the unbiased conversion is insignificant for small measurement errors. (Note that λ_θ calculation is omitted in the above equations. The detailed information about determining λ's is given in Chapter 3.3.3.)

If r and θ are known, the exact covariance matrix of x_m and y_m can be calculated by evaluating $\mathrm{var}(x_m^u \mid r, \theta)$, $\mathrm{var}(y_m^u \mid r, \theta)$ and $\mathrm{cov}(x_m^u, y_m^u \mid r, \theta)$. Although the measured range and bearing are not equal to r and θ, a quite accurate covariance matrix can still be obtained as follows [9].

$$R = \begin{bmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{bmatrix} \tag{3-49}$$

$$\begin{aligned} R_{11} &= \operatorname{var}(x_m^u \mid r_m, \theta_m) \\ &= E\{[\lambda_\theta^{-1} r_m \cos\theta_m - r\cos\theta]^2 \mid r_m, \theta_m\} \\ &= E\{[\lambda_\theta^{-1} r_m \cos\theta_m - (r_m - v_r)\cos(\theta_m - v_\theta)]^2 \mid r_m, \theta_m\} \\ &= (\lambda_\theta^{-2} - 2) r_m^2 \cos^2\theta_m + \frac{1}{2}(r_m^2 + \sigma_r^2)(1 + \lambda_\theta' \cos 2\theta_m) \end{aligned} \tag{3-50}$$

$$\begin{aligned} R_{22} &= \operatorname{var}(y_m^u \mid r_m, \theta_m) \\ &= E\{[\lambda_\theta^{-1} r_m \sin\theta_m - r\sin\theta]^2 \mid r_m, \theta_m\} \\ &= E\{[\lambda_\theta^{-1} r_m \sin\theta_m - (r_m - v_r)\sin(\theta_m - v_\theta)]^2 \mid r_m, \theta_m\} \\ &= (\lambda_\theta^{-2} - 2) r_m^2 \sin^2\theta_m + \frac{1}{2}(r_m^2 + \sigma_r^2)(1 - \lambda_\theta' \cos 2\theta_m) \end{aligned} \tag{3-51}$$

$$\begin{aligned} R_{12} &= \operatorname{cov}(x_m^u, y_m^u \mid r_m, \theta_m) \\ &= E\{(\lambda_\theta^{-1} r_m \cos\theta_m - r\cos\theta)(\lambda_\theta^{-1} r_m \sin\theta_m - r\sin\theta) \mid r_m, \theta_m\} \\ &= (\lambda_\theta^{-2} - 2) r_m^2 \cos\theta_m \sin\theta_m + \frac{1}{2}(r_m^2 + \sigma_r^2)\lambda_\theta' \sin 2\theta_m \end{aligned} \tag{3-52}$$

where $\lambda_\theta = E\{\cos v_\theta\}$ and $\lambda_\theta' = E\{\cos 2v_\theta\}$.

The intermediate steps of the derivations of Eqs. (3-50) through (3-52) can be found in [9].

3.3.2. Unbiased Converted Measurements in 3D

In the measurement update of the tracking filter, conversion from spherical to the Cartesian coordinates can be done by using the unbiased converted measurements in

3D. Measurements taken in spherical coordinates are transformed into Cartesian coordinates in an unbiased fashion which is given below.

$$x_m^u = \lambda_\theta^{-1}\lambda_\phi^{-1} r_m \cos\theta_m \cos\phi_m \tag{3-53}$$

$$y_m^u = \lambda_\theta^{-1}\lambda_\phi^{-1} r_m \sin\theta_m \cos\phi_m \tag{3-54}$$

$$z_m^u = \lambda_\phi^{-1} r_m \sin\phi_m \tag{3-55}$$

where $\lambda_\theta = E\{\cos v_\theta\}$, $\lambda_\phi = E\{\cos v_\phi\}$.

The covariance matrix of the measurements in Cartesian coordinates is calculated by the following equations [9].

$$R = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \tag{3-56}$$

$$\begin{aligned} R_{11} &= \operatorname{var}(x_m^u \mid r_m, \theta_m, \phi_m) \\ &= ((\lambda_\theta\lambda_\phi)^{-2} - 2) r_m^2 \cos^2\theta_m \cos^2\phi_m \\ &\quad + \frac{1}{4}(r_m^2 + \sigma_r^2)(1 + \lambda_\theta' \cos 2\theta_m)(1 + \lambda_\phi' \cos 2\phi_m) \end{aligned} \tag{3-57}$$

$$\begin{aligned} R_{22} &= \operatorname{var}(y_m^u \mid r_m, \theta_m, \phi_m) \\ &= ((\lambda_\theta\lambda_\phi)^{-2} - 2) r_m^2 \sin^2\theta_m \cos^2\phi_m \\ &\quad + \frac{1}{4}(r_m^2 + \sigma_r^2)(1 - \lambda_\theta' \cos 2\theta_m)(1 + \lambda_\phi' \cos 2\phi_m) \end{aligned} \tag{3-58}$$

$$\begin{aligned} R_{33} &= \operatorname{var}(z_m^u \mid r_m, \theta_m, \phi_m) \\ &= (\lambda_\phi^{-2} - 2) r_m^2 \sin^2\phi_m + \frac{1}{2}(r_m^2 + \sigma_r^2)(1 - \lambda_\phi' \cos 2\phi_m) \end{aligned} \tag{3-59}$$

$$\begin{aligned} R_{12} = R_{21} &= \operatorname{cov}(x_m^u, y_m^u \mid r_m, \theta_m, \phi_m) \\ &= ((\lambda_\theta \lambda_\phi)^{-2} - 2) r_m^2 \sin\theta_m \cos\theta_m \cos^2 \phi_m \\ &\quad + \frac{1}{4}(r_m^2 + \sigma_r^2) \lambda'_\theta \sin 2\theta_m (1 + \lambda'_\phi \cos 2\phi_m) \end{aligned} \tag{3-60}$$

$$\begin{aligned} R_{13} = R_{31} &= \operatorname{cov}(x_m^u, z_m^u \mid r_m, \theta_m, \phi_m) \\ &= (\lambda_\theta^{-1} \lambda_\phi^{-2} - \lambda_\theta^{-1} - \lambda_\theta) r_m^2 \cos\theta_m \sin\phi_m \cos\phi_m \\ &\quad + \frac{1}{2}(r_m^2 + \sigma_r^2) \lambda_\theta \lambda'_\phi \cos\theta_m \sin 2\phi_m) \end{aligned} \tag{3-61}$$

$$\begin{aligned} R_{23} = R_{32} &= \operatorname{cov}(y_m^u, z_m^u \mid r_m, \theta_m, \phi_m) \\ &= (\lambda_\theta^{-1} \lambda_\phi^{-2} - \lambda_\theta^{-1} - \lambda_\theta) r_m^2 \sin\theta_m \sin\phi_m \cos\phi_m \\ &\quad + \frac{1}{2}(r_m^2 + \sigma_r^2) \lambda_\theta \lambda'_\phi \sin\theta_m \sin 2\phi_m) \end{aligned} \tag{3-62}$$

where $\lambda'_\theta = E\{\cos 2v_\theta\}$, $\lambda'_\phi = E\{\cos 2v_\phi\}$.

3.3.3. Compensation Factor Computation for Unbiased Converted Measurements

λ's, which are also called as the compensation factors of unbiased converted measurements, are calculated using the moment generating function of Normal distribution $N(\mu, \sigma^2)$ given below.

$$M_X(t) = E\{e^{tX}\} = e^{t\mu + \frac{1}{2}\sigma^2 t^2} \tag{3-63}$$

Then, the expected value of $\cos v_\theta$ is calculated as follows.

$$
\begin{aligned}
E\{\cos v_\theta\} &= E\{\frac{e^{jv_\theta} + e^{-jv_\theta}}{2}\} = \frac{1}{2}\left(E\{e^{jv_\theta}\} + E\{e^{-jv_\theta}\}\right) \\
&= \frac{1}{2}\left\{[e^{t\mu+\frac{1}{2}\sigma_\theta^2 t^2}]_{\substack{\mu=0\\ t=j}} + [e^{t\mu+\frac{1}{2}\sigma_\theta^2 t^2}]_{\substack{\mu=0\\ t=-j}}\right\} \\
&= e^{-\frac{\sigma_\theta^2}{2}}
\end{aligned}
\tag{3-64}
$$

Remaining compensation factors are also calculated by the approach given above. The results are given in Table 3-1.

Table 3-1: Compensation Factors for the Unbiased Converted Measurements

- $\lambda_\theta = E\{\cos v_\theta\} = e^{-\frac{\sigma_\theta^2}{2}}$
- $\lambda_\phi = E\{\cos v_\phi\} = e^{-\frac{\sigma_\phi^2}{2}}$
- $\lambda'_\theta = E\{\cos 2v_\theta\} = e^{-2\sigma_\theta^2}$
- $\lambda'_\phi = E\{\cos 2v_\phi\} = e^{-2\sigma_\phi^2}$

CHAPTER 4

FILTERS USED IN THE SIMULATIONS

In Chapter 3, a discrete time model of the motion of the projectile is obtained as a nonlinear equation written as the numerical integral of the continuous time equations. The nonlinear equations relating the state vector to the measurements are also presented in the previous chapter. The nonlinear motion and measurement models resulted in the need of working with nonlinear filters.

The tracking methodology is applied to simulations by four different filters: the extended Kalman filter (EKF), the unscented Kalman filter (UKF), the particle filter (PF) and the marginalized particle filter (MPF). The details of these four filters are given in this chapter.

4.1. The Kalman Filter

Before talking about the extended and unscented Kalman filters, it is appropriate to give some information about the Kalman filter. The Kalman filter is a widely used tool for the estimation of linear discrete-time dynamic systems expressed in the following form.

$$x_k = F_{k-1} x_{k-1} + w_{k-1} \tag{4-1}$$

$$y_k = H_{k-1} x_k + v_k \tag{4-2}$$

where x is the state vector, F is the transition matrix, w is the discrete-time process noise, y is the measurement, H is the measurement model matrix and v is the measurement noise [10].

In every time step of the Kalman filter, the state estimation is composed of time update and measurement update. Time update is the step where the state and the measurement are predicted. The predicted state is then corrected in measurement update step according to the measurements [11], [10]. In the following two sections, the time update and the measurement update are explained in accordance with the tracking methodology which is presented in the previous chapter.

4.1.1. Time Update

The ballistic target tracking problem is first modeled in continuous time and then, this model is converted to discrete time. So the time update stage of the problem basically uses the integration of continuous time solution of the differential equation of the state where the noise vector is omitted. In other words, it is the computation of $E\{x_k \mid E\{x_{k-1}\}\}$. The procedure given in *Chapter 3.2.'Discrete Time Motion Model'* is used for the time update stages of all the filters in concern.

Measurement prediction is a straightforward conversion to spherical coordinates after the state is predicted in the UKF. There is no conversion for the measurement prediction in the EKF since measurement update is done using the unbiased converted measurements technique.

4.1.2. Measurement Update

Unlike the time update, measurement update is linear for EKF since unbiased converted measurements are used. The measurement update for the unscented Kalman filter, however, is nonlinear and it is performed by the help of unscented transform. The details of the algorithm are given in *Chapter 4.3.'The Unscented Kalman Filter'*.

4.2. The Extended Kalman Filter

The extended Kalman filter is a very widely used estimation algorithm for nonlinear systems. It is obtained by the linearization (a series expansion) of the nonlinearities in the dynamic system.

Since the system described by the Eqs. (3-10) and (3-32), (3-33), (3-34) is nonlinear, a first order EKF is used in the simulations. The second order EKF includes second-order correction terms and it gives better results theoretically. However, computation of second derivates is rather difficult and error-prone. Moreover, the sampling time of projectile tracking radars are so small that the system is almost linear between two consecutive measurement times. Therefore, a first order EKF is quite sufficient for this study. The EKF algorithm is summarized in Table 4-1 where z_k is the measurement at time k.

Table 4-1: The Extended Kalman Filter

[$x_{k|k}$, $P_{k|k}$] = EKF [$x_{k-1|k-1}$, $P_{k-1|k-1}$, z_k]

- Calculate $x_{k|k-1}$ using $x_{k-1|k-1}$ and 4th order Runke-Kutta. (Eq. (3-11))
- Calculate $P_{k|k-1}$. (Eq. (3-23))
- Convert the radar measurement z_k to Cartesian coordinates and calculate its covariance matrix, R (Eqs. (3-53) through (3-62))
- Calculate $x_{k|k}$ and $P_{k|k}$ (ordinary Kalman filter measurement update).

4.3. The Unscented Kalman Filter

The unscented Kalman filter is based on unscented transformation which was developed as a method to propagate mean and covariance information through nonlinear transformations [12]. A set of points, which are called as the *sigma points*, are chosen in such a way that the mean and covariance computed from these points are equal to the actual mean and the covariance matrix. The crucial point here is that the nonlinear function is applied to the sigma points individually and the sigma points are turned into transformed points. The mean and the covariance of the transformed points are the estimates of the state or the measurement after the nonlinearity is applied.

Although the methodology seems to be the same as that of particle filters, the sigma points make all the difference. Contrary to the particles in the particle filter, sigma points are drawn deterministically and they can possess weights that are out of the range [0, 1]. The sum of these weights is equal to 1 which is the same for particle filters.

After the weights of the sigma points, W_m and W_c , are determined, the unscented Kalman filter algorithm given in Table 4-3 is applied. The sigma point selection method used in this study is given in Table 4-2. 2n+1 sigma points are drawn according to this method where n is the dimension of the state vector.

Table 4-2: Sigma Point Selection Method (SPSM) [11]

[$x^{i}_{k-1|k-1}$] = SPSM [$x_{k-1|k-1}$, $P_{k-1|k-1}$]

- $\lambda = \alpha^{2}(n+k) - n$
- $W_{m}^{(0)} = \lambda / (n+\lambda)$

  $W_{c}^{(0)} = \lambda / (n+\lambda) + (1-\alpha^{2}+\beta)$

  $W_{m}^{(i)} = (1-W_{m}^{(0)}) / 2n$ i = 1,…,2n

  $W_{c}^{(i)} = (1-W_{c}^{(0)}) / 2n$ i = 1,…,2n
- $x^{(0)}_{k-1|k-1} = x_{k-1|k-1}$
- $x^{(i)}_{k-1|k-1} = x_{k-1|k-1} + \sqrt{(n+\lambda)}\left(\sqrt{P_{k-1|k-1}}\right)_{i}$ i = 1,…,n
- $x^{(i)}_{k-1|k-1} = x_{k-1|k-1} - \sqrt{(n+\lambda)}\left(\sqrt{P_{k-1|k-1}}\right)_{i}$ i = n+1,…,2n

where α, β and k are the parameters of the method and $\left(\sqrt{P_{k-1|k-1}}\right)_{i}$ is the ith column of the matrix $P_{k-1|k-1}$.

Note that the square root of a positive definite matrix is given as $A = \sqrt{P}$ where $P = AA^{T}$. If the matrix square root A of P is of the form $P = A^{T}A$, then the sigma points should be formed from the rows of $\sqrt{P_{k-1|k-1}}$ [12].

Table 4-3: The Unscented Kalman Filter

$[\, x_{k|k}\,,\ P_{k|k}\,] = \mathrm{UKF}\,[\, x_{k-1|k-1}\,,\ P_{k-1|k-1}\,,\ z_k\,]$

- Generate the set of sigma points according to the sigma point selection algorithm given in Table 4-2 and obtain $x^i_{k-1|k-1}$'s from $x_{k-1|k-1}$.
- Calculate $x^i_{k|k-1}$'s using $x^i_{k-1|k-1}$'s and 4th order Runke-Kutta (Eq. (3-11)) and obtain the predicted mean $\hat{\mu}_{k|k-1} = \sum_{i=1}^{p} W_m(i) x^i_{k|k-1}$ where $p = 2n+1$.
- Calculate the predicted covariance matrix.

  $P_{k|k-1} = Q_k + \sum_{i=1}^{p} W_m(i)\{x^i_{k|k-1} - \hat{\mu}_{k|k-1}\}\{x^i_{k|k-1} - \hat{\mu}_{k|k-1}\}^T$. (See Eqs. (3-28) through (3-31) for Q_k)
- Calculate the predicted measurements, $z^i_{k|k-1}$'s, for the sigma points and their mean, $\hat{z}_{k|k-1}$, using the Eqs. (3-32), (3-33), (3-34) and the vector W_m.

  $$z^i_{k|k-1} = h(x^i_{k|k-1}), \quad \hat{z}_{k|k-1} = \sum_{i=1}^{p} W_m(i) z^i_{k|k-1}$$
- Calculate the measurement prediction covariance matrix.

  $S_k = R_k + \sum_{i=1}^{p} W_c(i)\{z^i_{k|k-1} - \hat{z}_{k|k-1}\}\{z^i_{k|k-1} - \hat{z}_{k|k-1}\}^T$ (See Eqs. (3-56) through (3-62) for R_k.)
- Calculate the cross covariance matrix.

  $$P_{xy} = \sum_{i=1}^{p} W_c(i)\{x^i_{k|k-1} - \hat{\mu}_{k|k-1}\}\{z^i_{k|k-1} - \hat{z}_{k|k-1}\}^T$$
- Calculate $x_{k|k}$ and $P_{k|k}$.

  $$K_k = P_{xy} S_k^{-1} \qquad v_k = z_k - \hat{z}_{k|k-1}$$

  $$P_{k|k} = P_{k|k-1} - K_k S_k K_k^T \qquad x_{k|k} = \hat{\mu}_{k|k-1} + K_k v_k$$

4.4. The Particle Filter

The particle filter is a numerical approximation to nonlinear Bayesian filtering which performs sequential Monte Carlo estimation. In PF approach the probability density function is represented as point masses or *particles* as the filter's name implies. The method followed for this purpose is called as the sequential importance sampling which represents the probability density by a set of random samples with associated weights. Filtering is done using these random samples and weights [13].

The particle filter algorithm given in Table 4-4 is based on Gaussian optimal importance function which is given for a system represented by

$$x_k = f_{k-1}(x_{k-1}) + w_{k-1} \tag{4-3}$$

$$y_k = H_{k-1}x_k + v_k \tag{4-4}$$

Here, state dynamics is nonlinear and given by the function f which corresponds to the numerical solution of a differential equation by 4th order Runge-Kutta method. The measurements taken in spherical coordinates are converted into Cartesian coordinates using unbiased converted measurements technique. w_{k-1} and v_k are assumed to be mutually independent zero-mean white Gaussian noise whose covariance matrices are Q and R.

Stated in other words, the importance density and $p(z_k \mid x_{k-1})$ are assumed to be Gaussian and given by the following equations [13].

$$p(x_k \mid x_{k-1}, z_k) = \mathcal{N}(x_k; a_k, \Sigma_k) \tag{4-5}$$

$$p(z_k \mid x_{k-1}) = \mathcal{N}(z_k; b_k, S_k) \tag{4-6}$$

where a_k, b_k, Σ_k, S_k are given in Table 4-4.

4.4.1. Degeneracy Problem and Resampling

The variance of the weights of the particles can only increase in time [13]. This increase results in the degeneracy problem which means that after some time the normalized weights of a few particles approach to a reasonably large positive value while the others have negligible weights. In order to avoid the degeneracy problem, resampling is required which eliminates particles with negligible weights and multiplies particles possessing relatively high weights. The resampling algorithm used in this study is given in Table 4-5.

As it is seen in Table 4-4, resampling is performed after a threshold value, N_{thr}, exceeds the number of effective samples, N_{eff}. Although the threshold is determined as N in this study [14], still it can be chosen as a different value. If computation load of the filter is of no significance, N_{thr} can be set to N which is equal to the upper bound of N_{eff}. In this way, resampling can be performed in every time step of the filter.

It should also be noted that resampling in every time step may decrease the sample diversity and decrease the filter performance (sample impoverishment). However, in the simulations it is observed that the performance of PF does not decrease by setting $N_{thr} = N$ for our tracking problem. Contrarily, it is observed that the filter performs better if the particles are resampled in every time step. Still, in Table 4-4, the check for the number of effective particles is included for the completeness of the particle filter algorithm with Gaussian optimal importance function.

Table 4-4: The Particle Filter with Gaussian Optimal Importance Function [13]

```
$[\{x_k^i, w_k^i\}_{i=1}^N]$ = PF $[\{x_{k-1}^i, w_{k-1}^i\}_{i=1}^N, z_k]$

• FOR i=1:N                                  \\ N= number of particles
    ➢ Pick $x_k^i$ from $q(x_k \mid x_{k-1}^i, z_k) = N(x_k; a_k, \Sigma_k)$ where
          $a_k = f_{k-1}(x_{k-1}) + \Sigma_k H_k^T R_k^{-1}(z_k - b_k)$
          $\Sigma_k = Q_{k-1} - Q_{k-1} H_k^T S_k^{-1} H_k Q_{k-1}$
          $S_k = H_k Q_{k-1} H_k^T + R_k$
          $b_k = H_k f_{k-1}(x_{k-1})$
    ➢ $\tilde{w}_k^i \propto w_{k-1}^i p(z_k \mid x_{k-1}^i)$
• END FOR
• Calculate total weight $T = \sum_{i=1}^N \tilde{w}_k^i$ .
• FOR i=1:N
    ➢ Normalize $w_k^i = \frac{\tilde{w}_k^i}{T}$
• END FOR
• Calculate the number of effective samples $N_{eff} = \frac{1}{\sum_{i=1}^N (w_k^i)^2}$ and $N_{thr} = N$ .
• IF $N_{eff} \le N_{thr}$
    ➢ $[\{x_k^i, w_k^i\}_{i=1}^N]$=RSMPL[$\{x_k^i, w_k^i\}_{i=1}^N$]      \\ See Table 4-5
• END IF
```

Note that using the Gaussian optimal importance function makes it possible to update the importance weights before the particles are propagated in time. The nonlinear function $f_{k-1}(x_{k-1})$ is numerically calculated by using 4th order Runge-Kutta as it is done in EKF and UKF algorithms.

Table 4-5: Resampling Algorithm [13]

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

• $c_1 = w_k^1$                          \\ Initialize cumulative sum of weights
• FOR j=2:N
    ➢ $c_j = c_{j-1} + w_k^j$
• END FOR
• Pick $u_1 \sim U[0, \frac{1}{N}]$          \\ $U$ represents the uniform distribution.
• j=1                                    \\ Start from the first $c_j$
• FOR i=1:N
    ➢ $u_i = u_1 + \frac{i-1}{N}$
    ➢ WHILE $u_i > c_j$
        ➢ $j = j+1$
    ➢ END WHILE
    ➢ $x_k^i = x_k^j$
    ➢ $w_k^i = \frac{1}{N}$
• END FOR
```

4.5. The Marginalized Particle Filter

As the dimension of the state vector increases, the particle representation becomes too sparse to represent the posterior distribution of the state [14] [13]. In other words, the performance of the particle filter decreases with increasing state dimension. In order to overcome this problem, the marginalized particle filter is proposed which partitions the state into two as linear and nonlinear and applies the particle filter solution to the nonlinear part [14].

In this thesis a marginalized particle filter is proposed where the state vector given in (3-10) is divided into two parts as follows.

$$x_k^l = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \qquad x_k^n = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \Delta\beta \end{bmatrix} \tag{4-7}$$

where x_k^l and x_k^n are informally linear and nonlinear parts of the state. Note that both parts of the state vector possess nonlinear behavior so it is not possible to apply the existing algorithms.

Similar to the previous algorithms presented, the time update of both the linear and the nonlinear parts is performed by 4th order Runge-Kutta method. The nonlinear part of the state is represented by particles and an EKF is applied to each particle in order to obtain the posterior of the linear part since it is not truly linear. The MPF algorithm is given in Table 4-6 where $P_k^{l,i}$ represents the covariance matrix of x_k^l at time k.

Table 4-6: The Marginalized Particle Filter with Gaussian Optimal Importance Function [14]

```
$[\{x_k^i, w_k^i\}_{i=1}^N, P_k^{l,i}]$ = MPF $[\{x_{k-1}^i, w_{k-1}^i\}_{i=1}^N, P_{k-1}^{l,i}, z_k]$

• FOR i=1:N
    ➢ Pick $x_k^i$ from $q(x_k \mid x_{k-1}^i, z_k) = N(x_k; a_k, \Sigma_k)$ where
          $a_k = f_{k-1}(x_{k-1}) + \Sigma_k H_k^T R_k^{-1}(z_k - b_k)$
          $\Sigma_k = Q_{k-1} - Q_{k-1} H_k^T S_k^{-1} H_k Q_{k-1}$
          $S_k = H_k Q_{k-1} H_k^T + R_k$
          $b_k = H_k f_{k-1}(x_{k-1})$
    ➢ $\tilde{w}_k^i \propto w_{k-1}^i p(z_k \mid x_{k-1}^i)$
    ➢ Pick $x_{k-1}^{l,i}$'s from $x_{k-1}^i$'s and pick its covariance matrix $P_{k-1}^{l,i}$
    ➢ $[x_k^{l,i}, P_k^{l,i}]$ = EKF $[x_{k-1}^{l,i}, P_{k-1}^{l,i}, z_k]$
• END FOR
• Insert the linear state $x_k^{l,i}$ into $x_k^i$'s
• Calculate total weight $T = \sum_{i=1}^N \tilde{w}_k^i$.
• FOR i=1:N
    ➢ Normalize $w_k^i = \frac{\tilde{w}_k^i}{T}$
• END FOR
• Calculate the number of effective samples $N_{eff} = \frac{1}{\sum_{i=1}^N (w_k^i)^2}$ and $N_{thr} = N$.
• IF $N_{eff} \le N_{thr}$
    ➢ $[\{x_k^i, w_k^i\}_{i=1}^N]$=RSMPL$[\{x_k^i, w_k^i\}_{i=1}^N]$        \\ See Table 4-5
• END IF
```

CHAPTER 5

SIMULATIONS AND DISCUSSION

The tracking methodology and the algorithms that are used for ballistic target tracking are given in the previous chapters. In this chapter, the estimates of these algorithms and their performances will be discussed and compared.

5.1. Target Scenarios

In the simulations, the measurements are produced based on the data taken from a software which is able to produce 6-DOF trajectories of different kinds of ammunitions. The target trajectories are assumed to be taken from four kinds of simulated radars and the ground truth data obtained by this program is disturbed by adding zero mean Gaussian noise.

5.1.1. PRODAS V3

The filters mentioned in Chapter 4 are run using predetermined target scenarios. In other words, the measurements of the radar are fed to the filter offline. In this study, these measurements are based on PRODAS V3 (Projectile Rocket Ordnance Design & Analysis System), a computer program running on Windows OS. This program is used in simulating test firings, projectile modeling and estimating aerodynamics and stability. PRODAS V3 can compute 4 or 6 degrees of freedom projectile trajectories [15].

The ground truth data from PRODAS V3 is given in Cartesian coordinates and it is converted to ‘radar measurements’ by inserting noise. Before the noise is inserted, this data is converted to spherical coordinates since radar measures the position of the target in this coordinate system. All of the simulations are performed with radar measurements that are assumed to be taken in spherical coordinates. The drag parameter curves of the projectiles are also obtained from PRODAS V3

5.1.2. Radar Measurements

In order to observe the performance of the filters, 4 simulated radars with different measurement errors are assumed to exist. 1σ measurement error of the radars is given in Table 5-1. In practice, Radar 1 and 2 are not realistic as projectile tracking radars because of their small measurement frequency. However, for comparing the performances of the filters they are somewhat good examples.

Table 5-1: Measurement Frequency and Measurement Error of the Radars

	Measurement Frequency (Hz)	1σ Range Measurement Error (meters)	1σ Bearing & Elevation Error (degrees)
Radar 1	1	10	1 (~17.5mrad)
Radar 2	1	10	0.1 (~1.75mrad)
Radar 3	10	10	1 (~17.5mrad)
Radar 4	10	10	0.1 (~1.75mrad)

All of the radars given in Table 5-1 are assumed to have a probability of detection of 0.9 in the simulations.

$$P_D = 0.9 \tag{5-1}$$

In case the measurement of the target is not obtained at a particular time step, the filter is not run and the state estimate is calculated using the previous state estimate and 4th order Runge-Kutta method. In other words, the state is propagated to the next time step using the previous estimate.

Apart from the radars given in Table 5-1, it is assumed that an additional radar exists with 100 Hz measurement frequency. This radar is denoted as Radar 5 and it is used for drag parameter estimation simulations. 1 σ measurement error of Radar 5 is equal to 10 meters for range and 0.1° for bearing and elevation.

5.1.3. Obtaining the Target Scenarios

Three ground truth data is obtained from PRODAS V3 for 40, 35 and 25mm projectiles according to 6-DOF trajectory. The trajectories are for the above 3 projectiles fired at 30° elevation and they are given the name Target Scenario 1, 2 and 3 respectively. The curve given in the following figure, for example, is the ground truth data for Target Scenario 2 and the measurements from Radar 2 are the stars. The projectile is launched from the ground at 90° bearing and 30° elevation. Note that the grids for x, y and z axes are not identical in the figure.

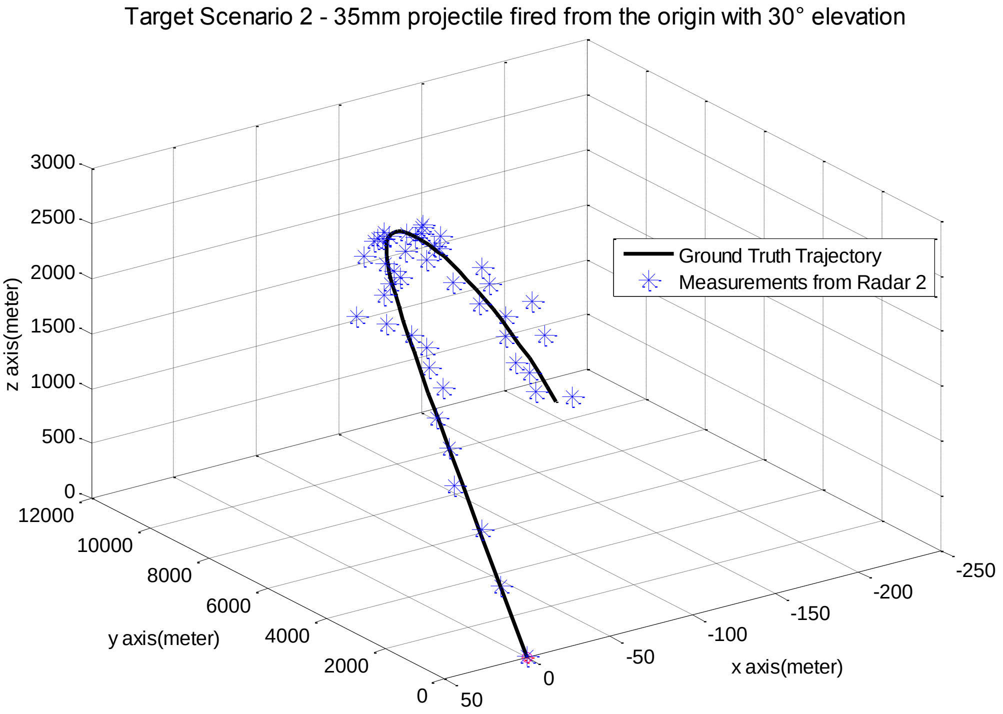

Figure 5.1: Target Scenario 2 - 35HEI_MSDCartridge

5.1.4. Performance Evaluation

The filter performances are compared with Monte Carlo simulation using the 3 target scenarios and the set of radars given in Table 5-1. In every run of the Monte Carlo simulation, the ground truth trajectory is disturbed with a new zero-mean Gaussian noise to obtain the current measurement. The filter's estimation error is calculated by comparing the state estimate with the ground truth. Root-Mean-Square Error (RMSE) of the state estimate is the main criterion of performance evaluation. It is calculated as follows.

$$RMSE = \frac{1}{N_{MC}} \sum_{n=1}^{N_{MC}} \sqrt{\frac{1}{K} \sum_{k=1}^{K} ((x_k - x_k^{GT})^2 + (y_k - y_k^{GT})^2 + (z_k - z_k^{GT})^2)} \qquad (5\text{-}2)$$

where N_{MC} is the number of Monte Carlo runs and K is the number of time steps in the target scenario. x_k^{GT}, y_k^{GT} and z_k^{GT} are the ground truth data while x_k, y_k and z_k are the position estimates of the filter at time k.

The RMSE results for EKF and UKF are obtained after 1000 Monte Carlo runs while the ones for PF and MPF are obtained after 100 Monte Carlo runs. This is due to the high computation load of particle filters.

5.2. Filter Initialization

In order to create a fair comparison, the filters are started using the same method to obtain the initial state and state covariance matrix. After converting the first two data of the radar to Cartesian coordinates, all of the filters are initialized with the following state vector and the covariance matrix related to it.

$$x_{0|0} = \begin{bmatrix} x & V_x & y & V_y & z & V_z & 0 \end{bmatrix}^T \qquad (5\text{-}3)$$

$$P_{0|0} = \begin{bmatrix} R_{11} & 0 & R_{12} & 0 & R_{13} & 0 & 0 \\ 0 & \sigma_v^2 & 0 & 0 & 0 & 0 & 0 \\ R_{12} & 0 & R_{22} & 0 & R_{23} & 0 & 0 \\ 0 & 0 & 0 & \sigma_v^2 & 0 & 0 & 0 \\ R_{12} & 0 & R_{23} & 0 & R_{33} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_v^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \sigma_{\Delta\beta}^2 \end{bmatrix} \qquad (5\text{-}4)$$

where *x*, *y* and *z* are the radar measurement taken in second time step. V_x, V_y and V_z are the velocity estimations of the target using the first two radar measurements. They are obtained by dividing the distance covered by Δt. σ_v^2 is the variance of the velocity in Cartesian coordinates. $\sigma_{\Delta\beta}^2$ is the variance of $\Delta\beta$. and R_{ij}'s are obtained from the Eqs. (3-56) through (3-62). $\sigma^2_{\Delta\beta}$ and σ^2_v are set to 10000 and 1000 respectively.

For Radar 1 and 2 the perceived motion of the projectile is quite nonlinear at the very beginning of the trajectory. This is due to two reasons. First, the projectile that is close to the muzzle experiences huge drag force since the velocity is at its maximum. Second, the measurement frequency of Radar 1 and 2 is 1 Hz which increases the nonlinearity of the motion since sampling period is relatively large. For this reason, at the initialization of the particle filter simulations with Radar 1 and 2, the particles are drawn from a Gaussian distribution whose mean and covariance are $x_{0|0}$ and $10P_{0|0}$ in order to deal with the high nonlinearity.

Apart from the state variables, there is also one more parameter to be set before filter initialization which is the β_0 of Eq. (3-5). The reference ballistic coefficient β_0 should be selected close to α^{-1} for the filter to converge to the true parameter as soon as possible. It is known that the drag parameter α of spin-stabilized projectiles is on the order of 10^{-3} [2]. Therefore, β_0 is set to 1000 for all the simulations. Note that the deviation of the actual value from the reference value is $\Delta\beta$ and it is estimated by the filter.

5.2.1. Number of Particles in PF and MPF

Particle filter's accuracy is definitely a function of number of particles used in the filter. However, the increase in the number of particles makes the filter impractical since it increases the computation load significantly. Therefore, particle filter's performance is observed for a set of number of particles for Target Scenario 2 tracked with Radar 2. The mean of the RMSE curves given in the following figure are obtained after 100 Monte Carlo runs.

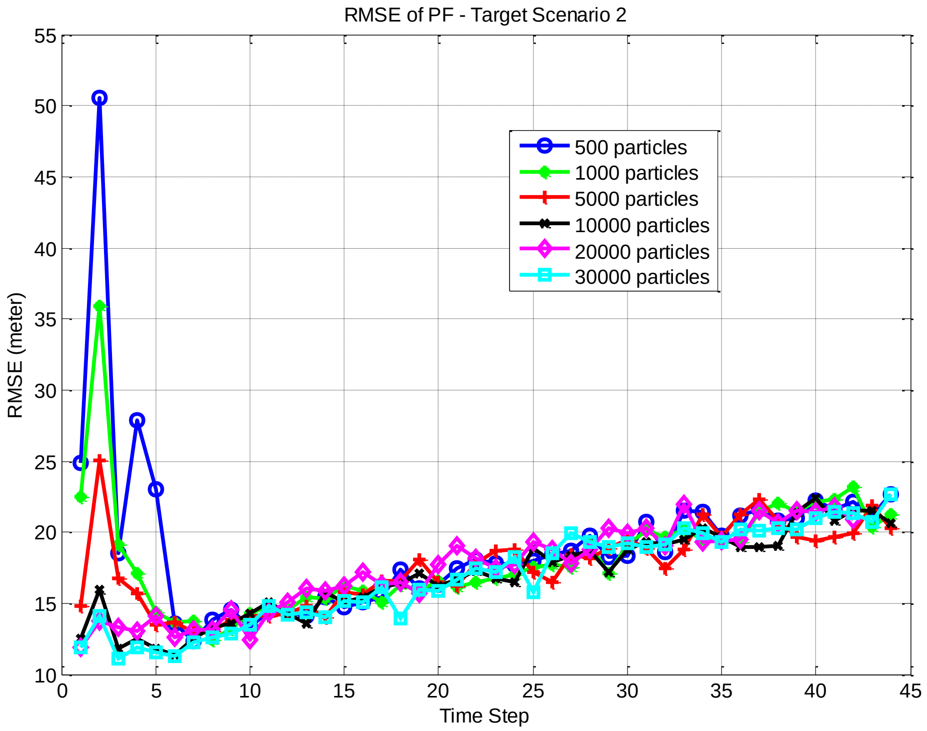

Figure 5.2: Target Scenario 2 – PF RMSE (100 Monte Carlo Runs)

It is observed that the performance of the filter at the beginning of the trajectory is directly related to the number of particles. However, it is also seen in Figure 5.2 that the performance does not differ much after a few steps from the filter initialization. Moreover, the RMSE results in Table 5-2 show that the performance improvement is insignificant after the number of particles exceed 10000. Thus, keeping the computation load in mind, the number of particles for PF and MPF in the simulations is set to 10000.

Table 5-2: RMSE of PF for Target Scenario 2 (Radar 2)

Number of Particles	**PF-RMSE (meter)**
500 particles	19.83
1000 particles	18.46
5000 particles	17.65
10000 particles	17.13
20000 particles	17.51
30000 particles	17.01

5.3. Simulation Results

The performances of the filters mentioned in Chapter 4 are evaluated by Monte Carlo simulation. The performance evaluation of all of the filters for the target scenarios 1 to 3 are presented in this section.

In order to obtain the possible best results, the power spectral density matrix for the noise which is given in Eq. (3-29) is multiplied by a scaling factor in every particular tracking simulation. In this way, the most suitable process covariance matrix calculation for the filters is obtained for each individual filter. These scaling factors are determined beforehand again by Monte Carlo runs.

Note that the time of flight and maximum range of the three projectiles are not the same. In other words, the target scenarios are quite different from each other. Therefore, the RMSE results are presented at separate tables for the three target scenarios.

5.3.1. Target Scenario 1

Target Scenario 1 is the trajectory of 40mm projectile launched at 30° elevation. The muzzle velocity of the projectile is 871m/s which is the smallest among the three target scenarios. Therefore, Target Scenario 1 possesses relatively the most linear trajectory and the RMSE of the trajectory estimation is the smallest. The RMSE results of the estimation obtained by Monte Carlo simulation are given in Table 5-3 below.

Table 5-3: RMSE for Target Scenario 1

	EKF-RMSE (meter)	UKF-RMSE (meter)	PF-RMSE (meter)	MPF-RMSE (meter)
Radar 1	41.87	40.99	78.34	72.98
Radar 2	14.05	12.58	15.76	15.56
Radar 3	14.62	14.33	24.14	23.83
Radar 4	5.87	5.51	6.35	6.42

It is observed from Table 5-3 that UKF performs the best for all of the cases. PF and MPF perform relatively poorly due to several reasons. In order to prevent degeneracy, resampling is done at each step in PF and MPF algorithms. However, since the actual process noise has low power, sample impoverishment problem occurs. To overcome this problem process noise covariance is unrealistically enlarged. Such an approach gives poor estimations as the results show. As a future study a new technique should be generated to prevent sample impoverishment. Moreover, the large dimension of the state decreases the performances of PF and MPF [14]. However, it should be stated that partitioning the state decreases the RMSE of PF as MPF's performance is superior to that of PF according to the results given in the table. Even though PF and MPF are run with 10000 particles whose computation load is enormous, the RMSE of these filters are the biggest.

It is also clear that the RMSE of all of the filters increases inevitably as the measurement frequency of the radar decreases. This is because the increase in measurement frequency decreases the nonlinearity of the target's perceived motion. Moreover, it enables the filter to estimate the same trajectory with a larger number of measurements.

The RMSE results of the filters obtained for Target Scenario 1 tracked with Radar 2 is given in Figure 5.3. The given error curves represent the mean of the error curves obtained after Monte Carlo simulation for a particular filter. As it is mentioned earlier, the curves for EKF and UKF are obtained after 1000 Monte Carlo runs while the curves for PF and MPF are obtained after 100 Monte Carlo runs. One more time, it can be seen in Figure 5.3 that UKF has the minimum RMSE. PF and MPF, on the other hand, perform almost the same. The slight difference is due to the relatively fast convergence of MPF when it is compared to PF.

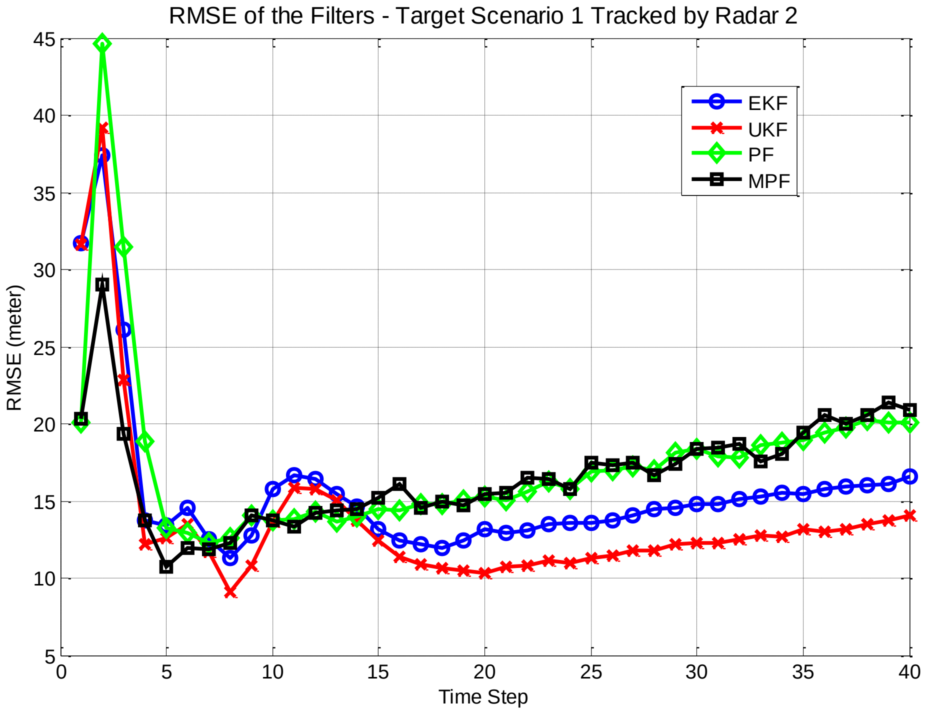

Figure 5.3: RMSE for Target Scenario 1 Tracked by Radar 2

5.3.2. Target Scenario 2

Target Scenario 2 belongs to the 35mm projectile fired at 30° elevation with a 1170m/s of muzzle velocity. The ground truth trajectory for this scenario can be seen in Figure 5.1.

The RMSE of the trajectory estimations obtained with this scenario is given in Table 5-4. It is observed that the general trend of the results is almost the same with the ones obtained for Target Scenario 1 in Table 5-3. The only difference is that the error values increased in Target Scenario 2 because of the increase in the nonlinearity of the projectile motion.

Table 5-4: RMSE for Target Scenario 2

	EKF-RMSE (meter)	**UKF-RMSE (meter)**	**PF-RMSE (meter)**	**MPF-RMSE (meter)**
Radar 1	52.55	42.93	89.48	88.46
Radar 2	16.50	12.83	17.13	16.23
Radar 3	17.36	16.39	29.04	27.03
Radar 4	6.98	6.43	7.82	7.34

The RMSE results for UKF where the target is tracked with Radar 2 are given in the figure below. The figure is obtained after 1000 Monte Carlo runs.

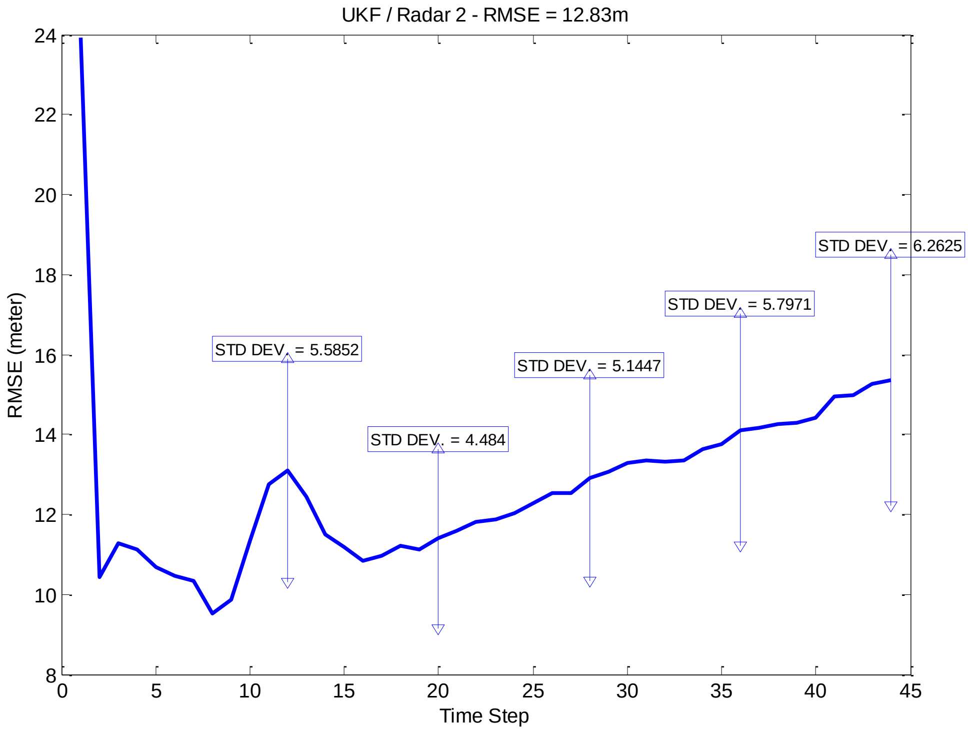

Figure 5.4: Target Scenario 2 – UKF RMSE (1000 Monte Carlo Runs)

It is clear in Figure 5.4 that the standard deviation of the error decreases first and then increases again towards the end of the trajectory. The first decrease is due to the decrease in the uncertainty of the state as radar measurements are received. At time step 20, it reaches its minimum with 4.48. However, at time step 44 it increases to 6.26 as the measurement errors in Cartesian coordinates increase with the increasing range of the projectile. For all of the filters, the behavior of the standard deviation of the errors is approximately the same as the one given in Figure 5.4.

5.3.3. Target Scenario 3

Target Scenario 3 is the trajectory of 25mm projectile launched at 30° elevation. The muzzle velocity is 1090 m/s. Although the muzzle velocity is less than that of 35mm, it is observed that the trajectory of 25mm projectile possesses more nonlinear motion especially in the first three seconds of its flight. The velocity curves of all of the projectiles studied are given in the figure below.

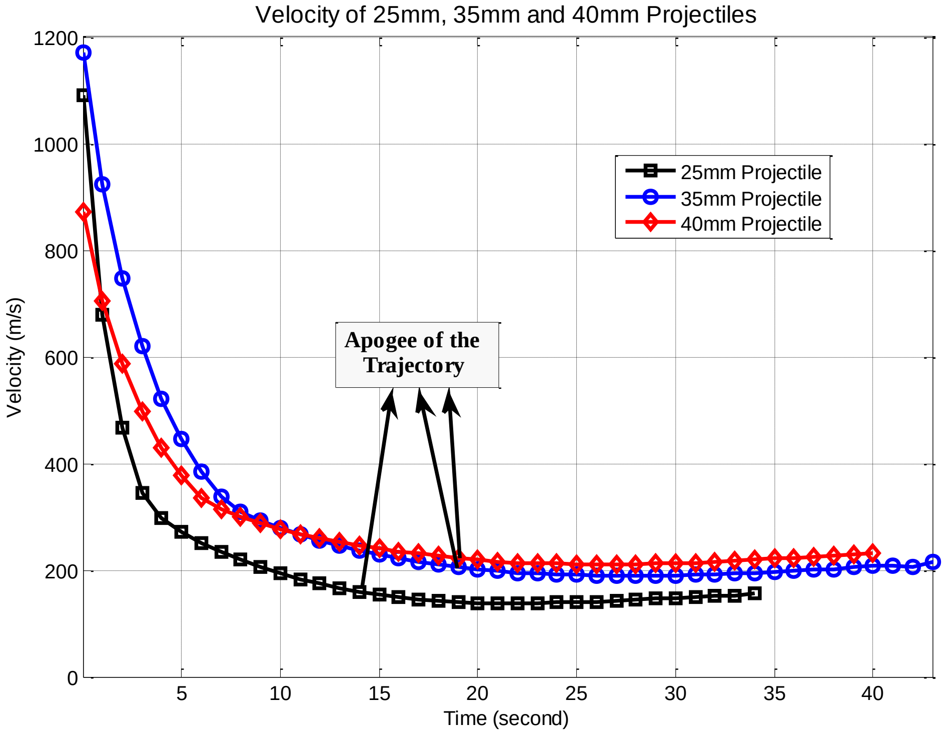

Figure 5.5: Velocity of 25mm, 35mm and 40mm Projectiles

It can be seen in Figure 5.5 that in the earlier parts of the trajectory, the velocity of 25mm projectile decreases sharply and the motion is quite nonlinear when it is compared to the trajectories of 35mm and 40mm. As a result, the performances of PF and MPF approach the performance of EKF which performs worse due to severe nonlinearity. Moreover, for some cases of Target Scenario 3, PF and MPF are superior to EKF as it is seen in Table 5-5. It is clear that UKF performs the best with the smallest RMSE as in the case of Target Scenarios 1 and 2. However, contrary to the results obtained for Target Scenarios 1 and 2, EKF has larger RMSE than PF and MPF for the projectile tracked by radar 2, 3 and 4.

Table 5-5: RMSE for Target Scenario 3

	EKF-RMSE (meter)	UKF-RMSE (meter)	PF-RMSE (meter)	MPF-RMSE (meter)
Radar 1	55.73	34.69	63.45	61.59
Radar 2	14.51	11.93	13.35	12.91
Radar 3	23.61	15.38	21.69	20.71
Radar 4	8.19	6.75	6.94	6.87

The particle filter performs quite well in low-dimensional state space [14]. However, after examining the RMSE results given for Target Scenarios 1 to 3, it is observed that for most of the cases, UKF and EKF outperform PF and MPF as a result of the relatively large state dimension [14] [13]. Still, it should be noted that the particle filter algorithms given in this thesis can be improved since we believe that poor performances of the PF and the MPF are due to the artificial increase of the process noise covariance to prevent sample impoverishment. Sample impoverishment problem is a hot topic in particle filter literature that some results can be applied to our problem [18].

Among the algorithms presented in this work, UKF performs obviously the best for all of the target scenarios. If the PF and MPF are put aside, this result can be explained as follows. The time update of UKF uses unscented transform with 4th order Runge-Kutta, while the time update of EKF uses Taylor series expansion with 4th order Runge-Kutta. UKF performs nonlinear measurement update where EKF has linear measurement update thanks to the unbiased converted measurements. Consequently, as the nonlinearity increases, the performance of EKF decreases considerably and UKF performs better.

5.3.4. Drag Parameter Estimation

As it is given in Chapter 2, drag parameter versus Mach number curves are extremely nonlinear (See Figure 2.2). In other words, the drag parameter varies continuously throughout the trajectory. In order to estimate it properly, the measurement frequency of the radar should be high. For this reason, the drag parameter estimation simulation is made by tracking the projectile with Radar 5 whose measurement frequency is assumed to be 100Hz.

The drag parameter, α, of the projectile is indirectly estimated since $\Delta\beta$ is in the state vector together with the position and velocity of the projectile (See Eq. (3-5)). β_0 , which does not change throughout the simulation, is set to 1000 and $\Delta\beta$ is set to 0 in the filter initialization as it is mentioned previously. According to the value set to β_0 and Eq. (3-5), the correct value of $\Delta\beta$ can be calculated from the drag parameter curves obtained from PRODAS V3. In addition to the estimated $\Delta\beta$ curve, the curve for the correct $\Delta\beta$ is also given in Figure 5.6 below.

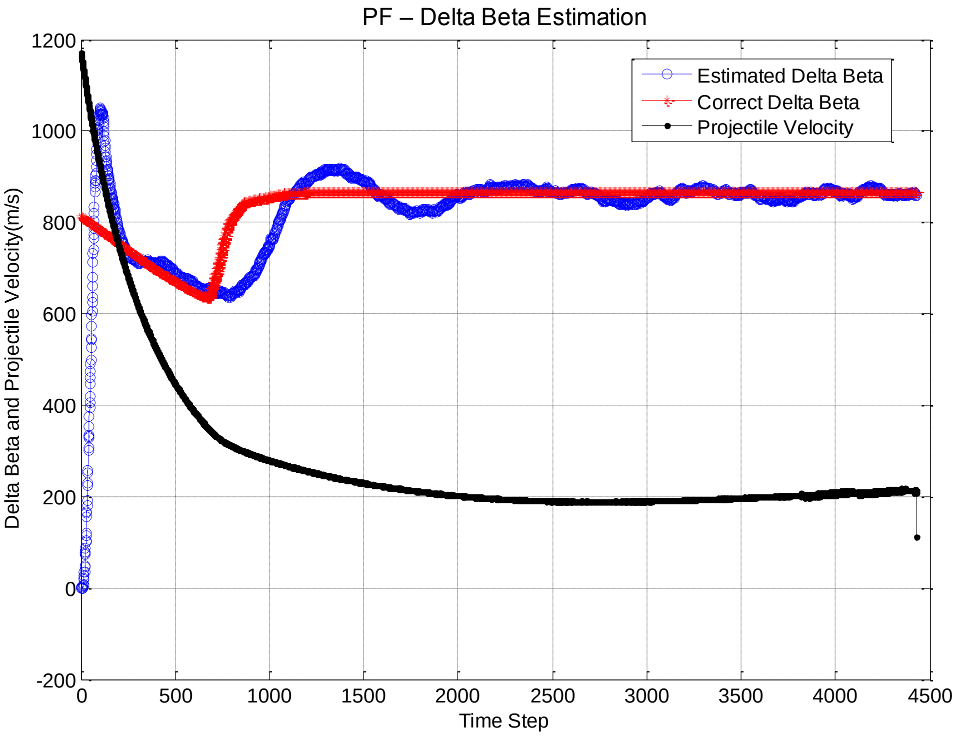

Figure 5.6: Target Scenario 1 tracked by Radar 5 – PF – $\Delta\beta$ Estimation

As a drag parameter estimation example, $\Delta\beta$ curve obtained by the PF is given in Figure 5.6 for the Target Scenario 1 tracked with radar 5. It is clear from the figure that when the measurement frequency is high, the drag parameter of the projectile can be estimated effectively. Thus, it is observed that if drag parameter curves of various projectiles are available, the target can be identified from its $\Delta\beta$ estimations.

5.3.5. Effect of Drift Calculation on Filter's Performance

Spin-stabilized projectiles are subjected to significant lateral drift, especially in long range firing. For instance, the drift of the projectile in Target Scenario 2 can be observed in Figure 5.7. Although the projectile is fired at 90° bearing, the impact point of the projectile drifts 208 meters in x axis.

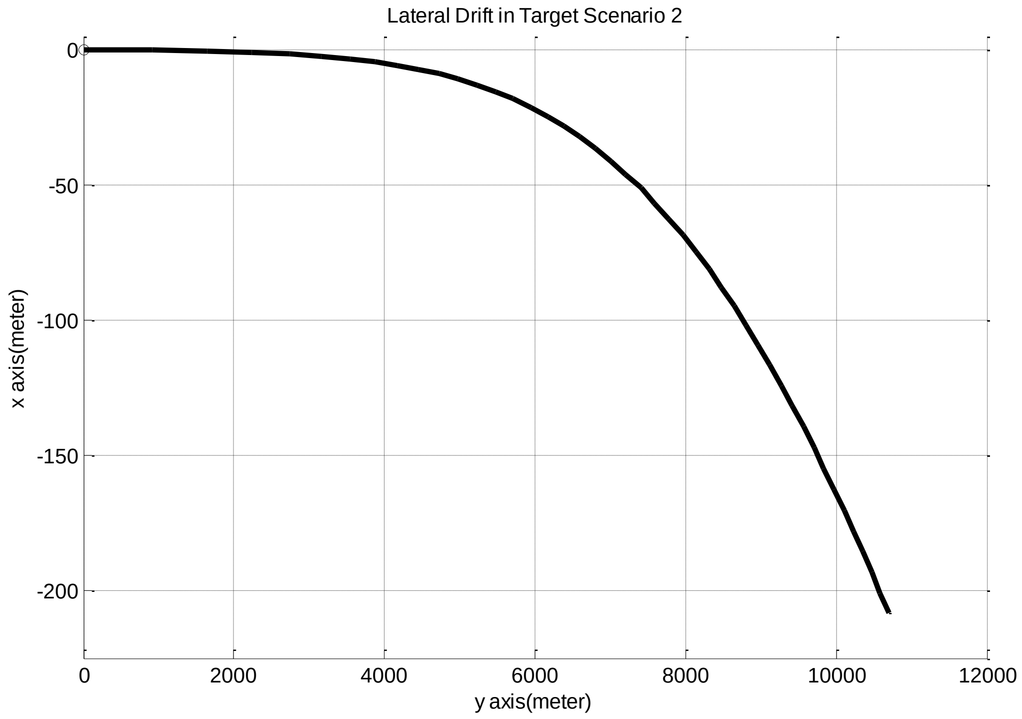

Figure 5.7: Target Scenario 2 (x-y view)

The drift of the projectile is simply approximated by Eq. (2-10) in the simulations. The results given on the left column of Table 5-6, for example, are obtained for UKF where drift calculation is not taken into account. It is clear that the performance of the filter decreases without drift correction especially for Radar 1 and 3 which have relatively large measurement errors. In the simulations, similar results are obtained for the other filters as well. Therefore, it is observed that if the measurement error of the radar is large, drift calculation should be considered in order to improve the filter’s performance. Note that if the projectile is not spin-stabilized, there is no need for drift calculation in the filter since the projectile will not experience any drift.

Table 5-6: RMSE of UKF in Target Scenario 2 – Effect of Drift Correction

	RMSE of UKF with Drift Correction (meter)	**RMSE of UKF without Drift Correction (meter)**
Radar 1	42.93	53.84
Radar 2	12.83	13.20
Radar 3	16.39	38.63
Radar 4	6.43	7.51

CHAPTER 6

CONCLUSIONS

In this thesis, ballistic target tracking problem is examined and four recursive estimation algorithms are studied in this context. The dynamics of ballistic targets is presented and the nonlinear system is estimated using extended Kalman filter, unscented Kalman filter, particle filter and marginalized particle filter. The performances of these filters are compared by Monte Carlo simulation using 6-DOF target trajectories.

Initially, the forces and moments acting on ballistic targets are given in detail. Then, the motion of the projectile is modeled considering the drag force and gravity which are the main factors determining the trajectory of a short-range ballistic target. The state vector is determined by augmenting the usual vector that contains the position and velocity of the target with a parameter called as $\Delta\beta$ in order to estimate the drag force effectively. Unbiased converted measurements are reviewed since the measurements of projectile tracking radars are in spherical coordinates while the position and the velocity of the target are in Cartesian coordinates in the state vector.

After the state space model is described, the aforementioned possible four techniques for tracking short-range ballistic targets are presented. First, brief information about the Kalman filter is given. Then, the extended Kalman filter, which is probably the most widely used estimation algorithm for nonlinear systems, and the unscented Kalman filter are introduced. Thereafter, the particle filter and the marginalized particle filter algorithms used in the thesis are presented. One cycle of all of the algorithms are given step by step in tables.

In order to measure the performance of the filtering techniques presented, 6-DOF projectile trajectories that are provided by PRODAS V3 are used. Since it is desired to see filter performances in a variety of situations, the targets are assumed to be tracked by four different simulated radars with different specifications.

One of the important and expected results of the simulations is that the RMSE of estimations of all the filters decreases as the frequency of the measurements increases. This is because the nonlinearity of the perceived motion decreases with decreasing sampling period and the trajectory is estimated with more measurements as the measurement frequency increases.

It is found that the most accurate filter is the unscented Kalman filter with the smallest RMSE in all target scenarios. Better performance of the UKF compared to EKF is expected especially at the highly nonlinear regions of the state equation. The unexpected, but reported result [17] is the poor performance of the particle filters.

Measurement update is nonlinear in UKF while it is linear in EKF where unbiased converted measurements are used for the conversion from the spherical coordinates to the Cartesian coordinates. The time update, on the other hand, is performed by sigma points in UKF with the help of unscented transform and 4th order Runge-Kutta. EKF uses linearization (a series expansion) of the nonlinearities and 4th order Runge-Kutta for the time update. Consequently, UKF gives better trajectory estimations than EKF as it is verified by the Monte Carlo simulation.

The reasons for the poor performances of the particle filters can be described as follows.

The dimension of the state vector is large for particle filter to represent the posterior distribution of the state effectively. Marginalized particle filters aim to overcome this problem. Accordingly, it is observed that the performance of the particle filter improves when the state vector is partitioned and marginalized particle filter is used. However, in the context of our tracking problem, the performance of MPF is still inferior to that of UKF. We believe that this result is due to the mismatch of the actual process noise power and the one that is used in the simulations to overcome the sample impoverishment problem. The artificial increase of the process noise power obviously generates a deviation from the correct model that may cause such degradation in the performance. Sample impoverishment problem is a well known problem that is still hot in the particle filter literature [18].

Lateral drift of the spinning projectile is approximated by a simple formula in the simulations. It is observed that this simple correction to target dynamics resulted in a remarkable improvement especially when the measurement error of the radar is relatively high.

As a future work, the tracking algorithms can be modified so that when the radar measurements of a particular segment of the target trajectory are available, the impact point or the launching point of the projectile is estimated. For these purposes, one can benefit from 4th order Runge-Kutta method in a similar approach used in this work. If a library for various projectiles can be constructed, the type of the projectile can be identified by the drag parameter estimations. Again as a future work, PF and MPF algorithms presented in this thesis can be studied further. These algorithms have several variations and they are open to improvement. Therefore, it wouldn’t be wrong to say that better results can be obtained for PF and MPF than the ones presented in this thesis. Moreover, the method applied for filter initialization can also be modified in order to decrease the time required for the filter to converge and in this way the RMSE of the filters can be reduced. Finally, in order to see the effect of the observer’s location, the simulations made for this study can be repeated with different simulated radars that are observing the target in different locations other than the origin.

REFERENCES

[1] X. Rong Li and Vesselin P. Jilkov,"Survey of Maneuvering Target Tracking. Part II: Motion Models of Ballistic and Space Targets", *IEEE Transactions On Aerospace And Electronic Systems*, pp. 96-119, January 2010.

[2] Robert L McCoy, *Modern Exterior Ballistics*, Schiffer Publishing Ltd, 1999.

[3] STANAG 4335, *NATO Standardization Agreement (STANAG) The Modified Point Mass and Five Degrees of Freedom Trajectory Models*, 2005.

[4] Ruprecht Nennstiel, *How do bullets fly?* http://www.nennstiel-ruprecht.de/bullfly/index.htm.

[5] Mehmet AKÇAY, *Balistik*, Kazmaz Matbaacılık, 2010.

[6] David F. Hardiman, J. Clayton Kerce and George C. "Brown, Nonlinear Estimation Techniques for Impact Point Prediction of Ballistic Targets", *Proceedings of SPIE*, Vol. 6236, 2006.

[7] Robert Grover Brown and Patrick Y.C. Hwang, *Introduction to Random Signals and Applied Kalman Filtering.* John Wiley and Sons Inc., pp. 198-204, 1997.

[8] Charles F.Van Loan, "Computing Integrals Involving the Matrix Exponential", *IEEE Transactions on Automatic Control*, Vols. AC-23, pp. 395-404, June 1978.

[9] Yaakov Bar-Shalom, "Unbiased Converted Measurements for Tracking", *IEEE Transactions On Aerospace And Electronic Systems*, Vol. 34, pp. 1023-1027, July 1998.

[10] Yaakov Bar-Shalom, X. Rong Li and Thiagalingam Kirubarajan, *Estimation with Applications To Tracking and Navigation.* Wiley-Interscience Publication, 2001.

[11] Jouni Hartikainen and Simo Särkkä, "Optimal Filtering with Kalman Filters and Smoothers - a Manual for Matlab Toolbox EKF/UKF", 2008.

[12] Simon J. Julier and Jeffrey K. Uhlmann, "Unscented Filtering and Nonlinear Estimation", *Proceedings Of The IEEE,* Vol. 92, pp. 401-422, March 2004.

[13] Branko Ristic, Sanjeev Arulampalam and Neil Gordon, *Beyond the Kalman Filter Particle Filters for Tracking Applications*, Artech House, 2004.

[14] Fredrik Linköping Gustafsson, "Particle Filter Theory and Practice with Positioning Applications" *IEEE A&E Systems Magazine*, Vol. 25, pp. 53-81, July 2010.

[15] PRODAS V3 - Software Tool for the Ballistics Professional, Arrow Tech Associates Inc., http://www.prodas.com.

[16] Steve Conover, J. Clayton Gerce, George Brown, Lisa Ehrman and David Hardiman, "Impact Point Prediction of Small Ballistic Munitions with an Interacting Multiple Model Estimator", *Proceedings of SPIE*, Vol. 6569, 2007.

[17] V. C. Ravindra, Y. Bar-Shalom and P. Willett, "Impact Point Prediction and Projectile Identification" *Proceedings of SPIE*, Vol. 6699, 2007.

[18] Umut Orguner, Fredrik Gustafsson, "Risk-Sensitive Particle Filters for Mitigating Sample Impoverishment", *IEEE Transactions on Signal Processing*, Vol. 56, pp. 5001-5012, October 2008.