TRACKING OF SUBSEQUENTLY FIRED PROJECTILES

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

BY

MEHMET POLAT

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

JULY 2012

Approval of the thesis:

TRACKING OF SUBSEQUENTLY FIRED PROJECTILES

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

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

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

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

Examining Committee Members:

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

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

Y. Doç. Dr. Afşar Saranlı
Electrical and Electronics Engineering Dept., METU ____________________

Y. Doç. Dr. Umut Orguner
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	: Mehmet POLAT
Signature	:

ABSTRACT

TRACKING OF SUBSEQUENTLY FIRED PROJECTILES

Polat, Mehmet

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

JULY 2012, 76 pages

In conventional tracking algorithms the targets are usually considered as point source objects. However, in realistic scenarios the point source assumption is often not suitable and estimating the states of an object extension characterized by a collectively moving ballistic object group (cluster) becomes a very critical and relevant problem which has applications in the defense area. Recently, a Bayesian approach to extended object tracking using random matrices has been proposed. Within this approach, ellipsoidal object extensions are modeled by random matrices and treated as additional state variables to be estimated. In this work we propose to use a slightly modified version of this new approach that simultaneously estimates the ellipsoidal shape and the kinematics of a group of ballistic targets. Target group that is tracked consists of subsequent projectiles. We use JPDAF framework together with the new approach to emphasize the pros and cons of both approaches. The methods are demonstrated and evaluated in detail by making various simulations.

Keywords: Subsequent Projectiles, Random Matrices, Extended Objects, Group Target Tracking, Ballistic Target Tracking, Cluster Tracking, Multi-Target Tracking, JPDAF, Filtering, Estimation.

ÖZ

PEŞ PEŞE ATILAN MERMİLERİN TAKİBİ

Polat, Mehmet

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

Temmuz 2012, 76 sayfa

Geleneksel takip algoritmalarında ilgilenilen hedeflerin nokta kaynaklı objeler olduğu varsayılır. Ancak gerçek uygulamalarda bu varsayım uygun olmamakta ve genişletilmiş obje şeklindeki kümelenmiş balistik hedeflerin takip edilmesi savunma sektöründe gittikçe önemli hale gelmektedir. Son zamanlarda rastgele matrisler kullanılarak genişletilmiş hedeflerin takip edilmesiyle ilgili bir yaklaşım önerilmiştir. Bu yaklaşımda, genişletilmiş objelerin şekli elips olarak kabul edilmekte ve bu şekil kinematik değişkenler ile birlikte tahmin edilmesi gereken ek bir değişken olarak tanımlanmaktadır. Tezde, sözü edilen yöntem peş peşe atılan bir grup merminin takibine uyarlanmıştır. Uygulamamızda daha gürbüz hale getirilen bu yeni yaklaşımın, genişletilmiş obje şeklindeki kümelenmiş balistik hedeflerin takibi konusundaki performansı JPDA yöntemi ile karşılaştırılarak incelenmiştir. Her iki yaklaşımın da avantaj ve dezavantajları tanımlanmış; değişik simulasyonlarla metotlar detaylandırılmıştır.

Anahtar Kelimeler: Peş Peşe Atılan Mermiler, Rastgele Matrisler, Genişletilmiş Hedefler, Grup Hedef Takibi, Balistik Hedef Takibi, Küme Takibi, Çoklu Hedef Takibi, JPDAF, Süzme, Tahmin Etme.

To my family,

ACKNOWLEDGEMENTS

I would like to express my sincerest thanks to my supervisor Prof. Dr. Mübeccel Demirekler for her guidance, support and valuable contributions throughout the preparation of this thesis.

I would like to acknowledge the support of ASELSAN Inc. for the realization of this thesis.

I would like to forward my appreciation to my family for their continuous support.

TABLE OF CONTENTS

CHAPTERS

APPENDICES

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

JPDA	Joint Probabilistic Data Association
JPDAF	Joint Probabilistic Data Association Filter
MHT	Multiple Hypothesis Tracking
PDA	Probabilistic Data Association
PDAF	Probabilistic Data Association Filter
SPD	Symmetric and Positive Definite

CHAPTER 1

INTRODUCTION

Tracking ballistic objects is an important problem which has applications in defense area. Accordingly, the objective of this thesis is to estimate the states of a group of subsequently fired projectiles using radar measurements. To do this we present two methods. One of them is the Joint Probabilistic Data Association Filter (JPDAF) [23] and the other is a novel Bayesian approach developed for cluster tracking using random matrices [1]. Within this approach, ellipsoidal object extensions are modeled by random matrices and these matrices are treated as additional state variables to be tracked. Finally we evaluate and compare the tracking performances of these two methods.

In this chapter, first, the motion and measurement equations that are used in ballistic target tracking is given. Then, the motion model used to simulate the trajectories of the projectiles in the thesis is explained briefly. Finally, the outline of the thesis is presented.

1.1 Motion Equations

The nonlinear equations of motion for a ballistic projectile in the (x, y, z) coordinate system (where x and y are the horizontal coordinates and z the vertical), is

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = P\alpha \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} - g \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{1.1}$$

where

- $g = 9.832\ m/s^2$ (acceleration due to gravity)
- α= drag coefficient of the projectile (in m^2/kg)
- $P = -\rho(z)\, V/2$(in kg/m^2s)

- $V = \sqrt{(\dot{x})^2 + (\dot{y})^2 + (\dot{z})^2}$ (speed in m/s)
- $\rho(z) = 0.002378 e^{\frac{-z}{30{,}000 * 0.3048}}$ (air density in kg/m^3 at altitude z in m)

and, $\dot{x}, \dot{y}$ and $\dot{z}$ are the velocity components with $\ddot{x}, \ddot{y}$ and $\ddot{z}$ being their corresponding acceleration components of the projectile in the three coordinates. In Equation (1.1) it is assumed that the forces acting on the object are gravity and drag furthermore the equations are obtained for a flat Earth [20]. Drag coefficient $\alpha,$ is assumed to be constant in the velocity range of the projectiles.

The set of equations given above is only a reasonable approximation to a much detailed model given in [18] however the model is sufficient for our purpose of observing the performance of different tracking algorithms. Flat Earth assumption is realistic for our problem since the ranges of the projectiles are small compared to the Earth radius.

The above differential equations are solved using Euler's method to iterate states in time. Time step of integration is taken as 0.01 sec. The MATLAB codes for the solution of the nonlinear ballistic equations are taken from [15].

1.2 Measurement Equations

To obtain the measurement equations it is assumed that stationary radar is located on the ground at the origin of the coordinate system. It measures the range (in meters), azimuth and elevation (both in radians) of the target located at (x, y, z), with respect to its position at periodic time instants. The range $r,$ the azimuth θ and the elevation ϵ of the target, with respect to the position of the radar (0, 0, 0), are measured at time intervals of length T, where

$$r = \sqrt{x^2 + y^2 + z^2} \tag{1.2}$$

$$\theta = \tan^{-1}\left(\frac{y}{x}\right) \tag{1.3}$$

$$\epsilon = \tan^{-1}\left(\frac{z}{\sqrt{x^2 + y^2 + z^2}}\right) \tag{1.4}$$

These spherical measurements have to be converted to Cartesian (via a nonlinear transformation) because the motion equations of the target are naturally expressed in Cartesian coordinates [20].

The observations in spherical coordinates are

$$r_m = r + w_r \qquad \theta_m = \theta + w_\theta \qquad \epsilon_m = \epsilon + w_\epsilon \tag{1.5}$$

where r, θ and ϵ are the true range, azimuth and elevation of the target; w_r, w_θ and w_ϵ are independent white, additive Gaussian noise sequences such that $w_r \sim N(0, {\sigma_r}^2)$, $w_\theta \sim N(0, {\sigma_\theta}^2)$ and $w_\epsilon \sim N(0, {\sigma_\epsilon}^2)$ with σ_r , σ_θ and σ_ϵ denoting the corresponding range, azimuth and elevation measurement error standard deviations.

The standard conversion from spherical to Cartesian coordinates is given by the Equations (1.6), (1.7) and (1.8) written below:

$$x_m = r_m \cos \epsilon_m \cos \theta_m \tag{1.6}$$

$$y_m = r_m \cos \epsilon_m \sin \theta_m \tag{1.7}$$

$$z_m = r_m \sin \epsilon_m \tag{1.8}$$

Note that the notion of ‘unbiased converted measurement’ is not used in our implementation.

1.3 Motion Model Used In the Thesis

The model given in the previous section, although not very accurate, is complicated for our application due to its nonlinear character. In this study we propose to use a linear model since we assume that the sampling rate is high.

We define $x(k)$ as the state of the projectile at time k as

$$x(k) = [x_k \quad \dot{x_k} \quad \ddot{x_k} \quad y_k \quad \dot{y_k} \quad \ddot{y_k} \quad z_k \quad \dot{z_k} \quad \ddot{z_k}]. \tag{1.9}$$

The state vector equations of a projectile are

$$x(k+1) = \begin{bmatrix} A & 0 & 0 \\ 0 & A & 0 \\ 0 & 0 & A \end{bmatrix} x(k) + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -g \end{bmatrix}' \tag{1.10}$$

$$where\ A = \begin{bmatrix} 1 & T & T^2/2 \\ 0 & 1 & T \\ 0 & P\alpha & 0 \end{bmatrix} \tag{1.11}$$

The velocity components of the state vector are evaluated according to the Equations (1.12), (1.13) and (1.14) given below

$$\dot{x_k} = V_t \cos \epsilon_m \cos \theta_m \tag{1.12}$$

$$\dot{y_k} = V_t \cos \epsilon_m \sin \theta_m \tag{1.13}$$

$$\dot{z_k} = V_t \sin \epsilon_m \tag{1.14}$$

Initially, the state vector is represented as $x(0) = \begin{bmatrix} 0 & \dot{x}_0 & 0 & 0 & \dot{y}_0 & 0 & 0 & \dot{z}_0 & 0 \end{bmatrix}'$ where $V_t = 500, \epsilon_m = {}^{\pi}/_{4}, \theta_m = {}^{\pi}/_{4}$ and $T = 0.01$.

1.4 Outline of the Thesis

The outline of the thesis is as follows.

In Chapter 2, the JPDAF algorithm is explained and is illustrated by two different scenarios.
In Chapter 3, the theory behind the Bayesian approach to extended object and cluster tracking using random matrices has been explained.
In Chapter 4, this new Bayesian approach is illustrated by a simulation example and the two approaches are compared briefly. Also all simulation results of the filters are presented and their performances are discussed in the same chapter.
In Chapter 5, the thesis is summarized and conclusions about the results of the simulations are presented.

CHAPTER 2

JOINT PROBABILISTIC DATA ASSOCIATION FILTER (JPDAF)

2.1 Introduction

Numerous methods and algorithms have been devoted to the problem of tracking multiple maneuvering targets in heavy clutter which is one of the most complicated cases in target tracking. One of these methods is the Multiple Hypothesis Tracking (MHT) method which seems to be the theoretically most powerful approach. But this method requires much computational power that restricts its implementation. Another less complicated approach is the JPDA (Joint Probabilistic Data Association). When tracking multiple closely spaced targets, the JPDA algorithm can be implemented successfully even in the presence of heavy clutter. The most important advantage of JPDA algorithm is it's relatively less computational power requirement when compared with MHT. Our problem is to track a group of subsequently fired projectiles. Therefore we assume that we know the number of targets and this number does not change during the tracking process. So the limitations of the JPDAF seem to be satisfied for our case.

In this chapter we first briefly explain the JPDAF algorithm and then illustrate its performance by simulation examples.

Motivation and Problem Formulation:

A situation of two targets in the same neighborhood as well as clutter (or false alarms) is illustrated in Figure 2.1. for a given time.

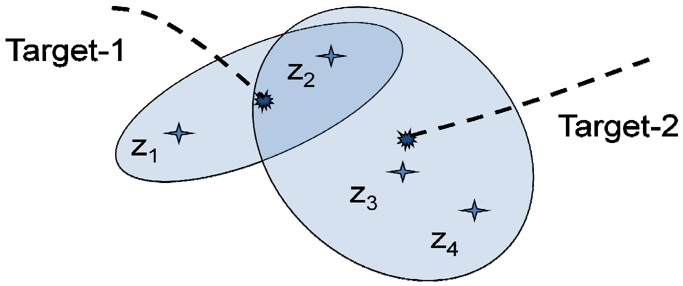


Figure 2.1. Two targets in the same neighbourhood

In this figure the following measurement-target associations are possible:

- is from Target-1 or clutter
- is either from Target-1 or Target-2 or clutter
- is either from Target-2 or clutter
- is either from Target-2 or clutter

The association of measurements in a multitarget environment has to be done by considering *simultaneously* all targets. The basic rule is that *a target may generate at most one measurement and a measurement may belong to only one target*.

2.2 The Joint Probabilistic Data Association Filter- JPDAF

2.2.1 Overview

Joint Probabilistic Data Association Filter (JPDAF) method is a popular Bayesian approach for multi target tracking problem. JPDAF is an extension of the Probabilistic Data Association Filter (PDAF) to multi target case. The basic assumption of this method is that the number of the target whose tracks have been established is known. It evaluates the measurement to target association probabilities and then combines them into the state estimates. Similar to Probabilistic Data Association (PDA), The JPDA approach assumes a sufficient statistic for the past observations (as a Gaussian pdf), in other words, it does not consider the past measurements anymore. Since the exact sufficient statistics is really a mixture of Gaussians replacing it by a single Gaussian is an approximation. The assumptions of the JPDAF are given below:

- There are a known number of targets in clutter.

- A target can give rise to at most one measurement.
- A measurement could have originated from at most one target.
- Measurements from one target may fall in the validation region of a neighboring target- this can happen over several sampling times.
- The past is summarized by an approximate sufficient statistic- state estimates (approximate conditional means) and covariance matrices for each target.
- The states are assumed Gaussian distributed with the above mentioned means and covariance matrices.
- The models for the various targets do not have to be the same.
- The targets are resolved- there are no unresolved (merged) measurements.

In the remaining part of this section we will give a brief explanation of the filter. The explanation of the algorithm is based on [23].

2.2.2 The Feasible Joint Events

Feasible joint events are defined by a matrix called the ‘event matrix’. Event matrices are defined by the ‘validation matrix’. Below we define both of these concepts.

The Validation Matrix

The validation matrix is a binary matrix where each row corresponds to a measurement while each column corresponds to a target or clutter so the size of the matrix is number of measurements × (number of targets +1).

Define the validation matrix

$$\Omega = [\omega_{jt}] \qquad j = 1 \dots m, \;\; t = 0,1 \dots N \tag{2.1}$$

with binary elements that indicate if the measurement j lies in the validation gate of the target t. The index $t = 0$ is used for “none of the targets” and the corresponding column of Ω has all ones since each measurement could have been originated from the clutter or false alarm [23].

An example of a validation matrix for two targets is given below.

$$\Omega = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix} \qquad j = 1 \dots 4, \qquad t = 0 \dots 2 \tag{2.2}$$

This corresponds to the situation depicted in Figure 2.1. The two targets in this example are "coupled" by measurement $j = 2$ because it lies in the intersection of the two validation regions.

The Event Matrix

A joint association event θ is represented by the event matrix

$$\theta = [\widehat{\omega}_{jt}] \tag{2.3}$$

consisting of the ones in Ω corresponding to the associations in θ,

$$\widehat{\omega}_{jt} = \begin{cases} 1 & if\ \theta_{jt} \in \theta \\ 0 & otherwise \end{cases} \tag{2.4}$$

where θ_{jt} is the event that measurement j originated from target t ($j = 1 \dots m,\ t = 0,1 \dots N$).

A feasible association event is the one where

(i) a measurement can have only one source, i.e.,

$$\sum_{t=0}^{\mathrm{N}} \widehat{\omega}_{jt} = 1 \quad \forall j \tag{2.5}$$

(ii) at most one measurement can originate from a target

$$\delta_t(\theta) \triangleq \sum_{j=1}^{m} \widehat{\omega}_{jt} \le 1 \qquad t = 1 \dots N \tag{2.6}$$

Generation of the Feasible Joint Association Events

The generation of the event matrices θ can be done by scanning Ω and picking one unit per row, and one unit per column except for $t = 0$ since the number of units (which is the number of false measurements) is not restricted [23].

The binary variable $\delta_t(\theta)$ defined in (2.6) is called the ***target detection indicator*** since it indicates whether it has been detected or not.

Another binary variable which is called ***measurement association indicator*** is defined as

$$\tau_j(\theta) \triangleq \sum_{t=1}^{N} \widehat{\omega}_{jt} \tag{2.7}$$

to indicate if the measurement j is associated with any target in the event θ. By the help of this definition, the number of false (unassociated) measurements in the event θ can be defined as

$$\emptyset(\theta) = \sum_{j=1}^{m} [1 - \tau_j(\theta)] \tag{2.8}$$

The next step will be the calculation of the prior and the posterior probabilities of joint association events.

The following simple example clarifies the definitions given up to this point,

$$\Omega = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix} j = 1,2,3 \quad t = 0,1,2 \tag{2.9}$$

From this validation matrix, one will obtain seven different event matrices:

$$\theta_1 = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad \theta_2 = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \theta_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \quad \theta_4 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

$$\delta(\theta_1) = [0 \quad 0] \quad \delta(\theta_2) = [0 \quad 1] \quad \delta(\theta_3) = [0 \quad 1] \quad \delta(\theta_4) = [1 \quad 0]$$

$$\theta_5 = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad \theta_6 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \quad \theta_7 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.10}$$

$$\delta(\theta_5) = [0 \quad 1] \quad \delta(\theta_6) = [1 \quad 1] \quad \delta(\theta_7) = [1 \quad 1]$$

θ_1: No measurement is detected. All of them are false measurements (clutter).

θ_2: Third measurement is associated with Target-2. The other measurements are clutter.

θ_3: Second measurement is associated with Target-2. The other measurements are clutter.

θ_4 : First measurement is associated with Target-1. The other measurements are clutter.

θ_5 : First measurement is associated with Target-2. The other measurements are clutter.

θ_6 : First measurement is associated with Target-1, second measurement is associated with Target-2 and third measurement is clutter.

θ_7 : First measurement is associated with Target-1, third measurement is associated with Target-2 and second measurement is clutter.

2.2.3 Evaluation of the Joint Probabilities

We define $Z(k)$ as the set of all measurements at time k and $Z^k = (Z(1), Z(2), \dots, Z(k))$ as the set of all measurements up to and time k. We also define $m(k)$ as the number of measurements in the union of the validation regions at time k. The ***joint association event probabilities*** are, with Bayes' formula,

$$\begin{aligned} P\{\theta(k)\big|Z^k\} &= P\{\theta(k)\big|Z(k), m(k), Z^{k-1}\} \\ &= \frac{1}{c} p[Z(k)|\theta(k), m(k), Z^{k-1}] P\{\theta(k)\big|Z^{k-1}, m(k)\} \\ &= \frac{1}{c} p[Z(k)|\theta(k), m(k), Z^{k-1}] P\{\theta(k)|m(k)\} \end{aligned} \tag{2.11}$$

where c is the normalization constant.

The Likelihood Function of a Joint Association Event

The ***likelihood function of the joint association event*** on the right hand side of (2.11) is

$$p[Z(k)\big|\theta(k), m(k), Z^{k-1}] = \prod_{j=1}^{m(k)} p[z_j(k)|\theta_{jt_j}(k), Z^{k-1}] \tag{2.12}$$

where t_j is the index of the target to which measurement j is associated in the θ_{jt_j} under consideration.

The conditional pdf of a measurement given its origin is

$$p\left[z_j(k)\middle|\theta_{jt_j}(k), Z^{k-1}\right] = \begin{cases} f_{t_j}\left[z_j(k)\right] & if\ \tau_j[\theta(k)] = 1 \\ V^{-1} & if\ \tau_j[\theta(k)] = 0 \end{cases} \tag{2.13}$$

where

$$f_{t_j}\left[z_j(k)\right] = N[z_j(k); \hat{z}^{t_j}(k|k-1), S^{t_j}(k)] \tag{2.14}$$

and $\hat{z}^{t_j}(k|k-1)$ is the predicted measurement for target t_j, with associated innovation covariance $S^{t_j}(k)$.

Measurements not associated with a target are assumed uniformly distributed in the surveillance region of volume V.

Using (2.13), the pdf (2.12) can be written as follows

$$p[Z(k)|\theta(k), m(k), Z^{k-1}] = V^{-\emptyset} \prod_{j=1}^{m(k)} \{f_{t_j}[z_j(k)]\}^{\tau_j} \tag{2.15}$$

In the above V^{-1} is raised to power $\emptyset(\theta)$, the total number of false measurements in event $\theta(k)$ and the indicators $\tau_j(\theta)$ select the single measurement densities according to their associations in event $\theta(k)$.

The Prior Probability of a Joint Association Event

$P\{\theta(k)|m(k)\}$ is obtained next. The last term of Equation (2.11) can be defined as

$$P\{\theta(k)|m(k)\} = P\{\theta(k), \delta(\theta), \emptyset(\theta)|\mathrm{m(k)}\} \tag{2.16}$$

The above Equation (2.16) can be rewritten as

$$P\{\theta(k)|m(k)\} = P\{\theta(k)|\delta(\theta), \emptyset(\theta), m(k)\}P\{\delta(\theta), \emptyset(\theta)|\mathrm{m(k)}\} \tag{2.17}$$

We replace the first term on the right hand side of the above with (2.18)

$$P\{\theta(k)|\delta(\theta), \emptyset(\theta), m(k)\} = \left(\frac{m(k)!}{\emptyset!}\right)^{-1} \tag{2.18}$$

and after some manipulations and assuming δ and $\emptyset$ independent, we replace the last term in (2.17) with (2.19)

$$P\{\delta(\theta), \emptyset(\theta)|\mathrm{m(k)}\} = \left[\prod_t ({P_D}^t)^{\delta_t}(1-{P_D}^t)^{1-\delta_t}\right]\mu_F(\emptyset) \tag{2.19}$$

where ${P_D}^t$ is the detection probability of target t and $\mu_F(\emptyset)$ is the probability mass function of the clutter model. Combining (2.18) and (2.19) into (2.17) yields the ***prior probability of a joint association event*** $\theta(k)$ as

$$P\{\theta(k)|m(k)\} = \frac{\mu_F(\emptyset)\emptyset!}{m(k)!}\left[\prod_t ({P_D}^t)^{\delta_t}(1-{P_D}^t)^{1-\delta_t}\right] \tag{2.20}$$

To give an example, suppose that at a specific time $\delta(\theta_i) = [1 \quad 1]$ and there exists 3 measurements. Given this information one can conclude that there exists 2 targets and one measurement is associated with Target-1 and one measurement is associated with Target-2. The left measurement is always a false alarm.

$$m(k) = 3, \ \emptyset = 1, \quad \frac{m(k)!}{\emptyset!} = 6$$

So 6 different joint association events $\theta(k)$, may cause this $\delta(\theta_i) = [1 \quad 1]$.

$$\theta_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \theta_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \theta_3 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$
$$\theta_4 = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \theta_5 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}, \theta_6 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

Assuming each such event is a priori equally likely, one has

$$P\{\theta(k)|\delta(\theta), \emptyset(\theta), m(k)\} = \left(\frac{m(k)!}{\emptyset!}\right)^{-1} = \frac{1}{6}$$

$$P\{\theta(k)|m(k)\} = \frac{1}{6}\mu_F(1){P_D}^1{P_D}^2$$

The Posterior Probability of a Joint Association Event

Combining (2.15) and (2.20) into (2.11) yields the posterior ***probability of a joint association event*** $\theta(k)$ as

$$P\{\theta(k)\big|Z^k\} = \frac{1}{c}\frac{\emptyset!}{m(k)!}\mu_F(\emptyset)V^{-\emptyset}\prod_j \{f_{t_j}[z_j(k)]\}^{\tau_j}\prod_t ({P_D}^t)^{\delta_t}(1 - {P_D}^t)^{1-\delta_t} \tag{2.21}$$

where $\emptyset$, δ_t and τ_j are all functions of the event $\theta(k)$ under consideration.

To define Equation (2.21) completely we have to define the probability mass function (pmf) of the clutter model $\mu_F(\emptyset)$. We use Poisson pmf for this model,

$$\mu_F(\emptyset) = e^{-\lambda V}\frac{(\lambda V)^{\emptyset}}{\emptyset!} \tag{2.22}$$

which requires the spatial density λ for the false measurements.

Using (2.22) in (2.21) leads to the cancellation of $V^{-\emptyset}$ and $\emptyset!$. Thus Equation (2.21) becomes;

$$P\{\theta(k)|Z^k\} = \frac{\lambda^{\emptyset} e^{-\lambda V}}{c\, m(k)!} \prod_{j} \{f_{t_j}\left[z_j(k)\right]\}^{\tau_j} \prod_{t} ({P_D}^t)^{\delta_t} (1 - {P_D}^t)^{1-\delta_t} \tag{2.23}$$

2.2.4 The State Estimation

The state estimation (filtering) algorithm can be carried out according to the assumption that the states of the targets conditioned on the past observations are mutually *independent.* The product term written on the right hand side of the (2.12) also follows from this assumption. In this case we need the ***marginal association probabilities***, which are obtained from the joint probabilities by summing over all the joint events in which the marginal event of interest occurs [23]. Using definition (2.4) this summation can be written as follows

$$\begin{aligned} \beta_{jt}^{i} &\triangleq P\{\theta_{jt}|Z^k\} \quad i = 0 \ldots m \\ &= \sum_{\theta} P\{\theta|Z^k\} \hat{\omega}_{jt}(\theta) \\ &= \sum_{\theta : \theta_{jt} \in \theta} P\{\theta|Z^k\} \end{aligned} \tag{2.24}$$

Updates:

Consider the stochastic system

$$x_{k+1} = Ax_k + Gw_k$$

$$z_k = Cx_k + Hv_k$$

where $x_k \in R^n, z_k \in R^p, w_k \in R^g, v_k \in R^h$; A, G, C and H are time-invariant, known matrices of appropriate dimension. The basic random variables $\{x_0, w_0, \ldots, v_0, \ldots\}$ are all independent and Gaussian, with $x_0 \sim N(0, P_0)$, $w_k \sim N(0, Q)$, $v_k \sim N(0, R)$.

The covariance matrices P_0, Q, R are all known.

According to these system equations and by the help of Equation (2.24) we apply Kalman filter for all measurements. We define $x_{k|k}$ as the conditional mean of x_k given the available information z^k and define $P_{k|k}$ as the conditional covariance. So the conditional density $p_{k|k} \sim N(x_{k|k}, P_{k|k})$ can be obtained from the following recursion relations:

$$x_{k|k}^{i} = x_{k|k-1} + W_k v_k^i \qquad i = 1 \dots m \tag{2.25}$$

$$v_k^i = z_k^i - z_{k|k-1} \tag{2.26}$$

$$W_k = P_{k|k-1} C_k^T S_k^{-1} \tag{2.27}$$

Here S_k is defined as:

$$S_k = C_k P_{k|k-1} C_k^T + \; H_k R H_k^T \tag{2.29}$$

If there exists no measurement, i.e., $i = 0$

$$x_{k|k} = x_{k|k-1} \tag{2.29}$$

By the help of Equation (2.24), we obtain state vector update formulas as follows:

$$x_{k|k} = x_{k|k-1} + \; W_k v_k \tag{2.30}$$

$$v_k = \sum_{i=1}^{m} \beta_{jt}^i v_k^i \tag{2.31}$$

and we obtain covariance matrix update formulas as follows:

$$P_{k|k} = \beta_{jt}^0 P_{k|k-1} + \; \left(1 - \beta_{jt}^0\right) P_{k|k}^c + \bar{P}_k \tag{2.32}$$

$$P_{k|k}^c = P_{k|k-1} - W_k S_k W_k^T \tag{2.33}$$

$$\bar{P}_k = W_k \left[\sum_{i=1}^{m} \beta_{jt}^i v_k^i v_k^{iT} - v_k v_k^T \right] W_k^T \tag{2.34}$$

SUMMARY:

The “pseudo code” for the algorithm explained in this Chapter is given in Table 2.1.

Table 2.1. Pseudo Code For the JPDAF algorithm

```
Step-1: Generate validation matrix that indicates all the possible sources of each measurement

Step-2: Obtain all feasible joint association events (hypotheses) from the validation matrix according to rules
    — one source for each measurement
    — one measurement (or none) from each target

Step-3: Obtain the prior probabilities of these joint events according to following assumptions and formula
    — False measurements are uniformly distributed in the surveillance region
    — The number of false measurements is distributed according to Poisson rule

    $P\{\theta(k)|m(k)\} = \frac{\mu_F(\emptyset)\emptyset!}{m(k)!}\left[\prod_t ({P_D}^t)^{\delta_t}(1-{P_D}^t)^{1-\delta_t}\right]$

Step-4: Obtain the posterior probabilities of these joint events according to formula

    $P\{\theta(k)|Z^k\} = \frac{\lambda^{\emptyset}e^{-\lambda V}}{c\, m(k)!}\prod_j \{f_{t_j}[z_j(k)]\}^{\tau_j}\prod_t ({P_D}^t)^{\delta_t}(1-{P_D}^t)^{1-\delta_t}$

Step-5: Obtain the marginal association probabilities according to formula

    $\beta_{jt}^i \triangleq P\{\theta_{jt}|Z^k\} \quad i = 0 \ldots m$
    $= \sum_\theta P\{\theta|Z^k\}\hat{\omega}_{jt}(\theta)$
    $= \sum_{\theta:\theta_{jt}\in\theta} P\{\theta|Z^k\}$

Step-6: Obtain the state estimates [{$x_{k|k}$, $P_{k|k}$}]=KF [{$x_{k-1|k-1}$, $P_{k-1|k-1}$},$z^k$]
```

Table 2.2. Pseudo Code For the JPDAF algorithm (cont'd)

```
State Vector Update Formulas:
    $x_{k|k} = x_{k|k-1} + W_k v_k$
    $v_k = \sum_{i=1}^{m} \beta_{jt}^{i} v_k^i$
Covariance Matrix Update Formulas:
    $P_{k|k} = \beta_{jt}^{0} P_{k|k-1} + \left(1 - \beta_{jt}^{0}\right) P_{k|k}^{c} + \bar{P}_k$
    $P_{k|k}^{c} = P_{k|k-1} - W_k S_k W_k^T$
    $\bar{P}_k = W_k \left[ \sum_{i=1}^{m} \beta_{jt}^{i} v_k^i v_k^{iT} - v_k v_k^T \right] W_k^T$
```

2.3 Simulations

The aim of this section is to analyze the performance and the behavior of the JPDA filter on some examples. We have generated two different scenarios for this purpose. The first scenario investigates whether our implementation works properly or not. To do this we analyze the generated hypotheses with their probabilities and try to understand whether there exists a correct matching of the received measurements at some specific time instant. For the second scenario, the starting points of motions of the targets and their velocities are chosen as specific values to ensure at least one crossing time instant. Later, we plot the error graphs of the targets.

2.3.1 The First Scenario

In the first scenario there exist 4 targets in the surveillance region.

- The targets move according to constant velocity model.

- The starting points of motions of the targets are different and their velocities are chosen randomly.
- Unless otherwise stated, target detection probability is 0.9 for all targets. This means that at a specific time instant it is possible not to receive a measurement from a specific target with a probability of 0.1.
- Targets may have different gate size. But for all time instants gate size of the target is assumed to be stationary. The gate shapes are always ellipsoidal for all time instants and for all targets.
- Target tracks may cross each other.
- There always exists a heavy clutter in the system. Clutter is uniformly distributed over the surveillance region with a density determined from Poisson distribution.

The algorithm first generates the validation matrix at every time instant. From this validation matrix, all feasible hypotheses are generated and their scores are computed and summarized. Finally by using these scores and received measurements, algorithm tries to make the best measurement update according to the formulas given in Section 2.2.4.

This algorithm gives satisfactory results for clusters containing up to 4 targets. When the number of targets in a cluster exceeds this limit, however, the total number of feasible hypotheses increases exponentially. So JPDAF requires prohibitive amounts of processor time to compute all the joint probabilities.

2.3.1.1. Experiments

We have made three experiments to investigate the computational load of JPDA. For the first experiment the gate sizes of the targets are chosen as their nominal values and clutter is uniformly distributed over the surveillance region with a density determined from Poisson distribution of parameter $\lambda = 100$. These parameters describe a nominal system. Figure 2.2 depicts the true tracks, measured and filtered positions of all targets with the clutter in the surveillance region.

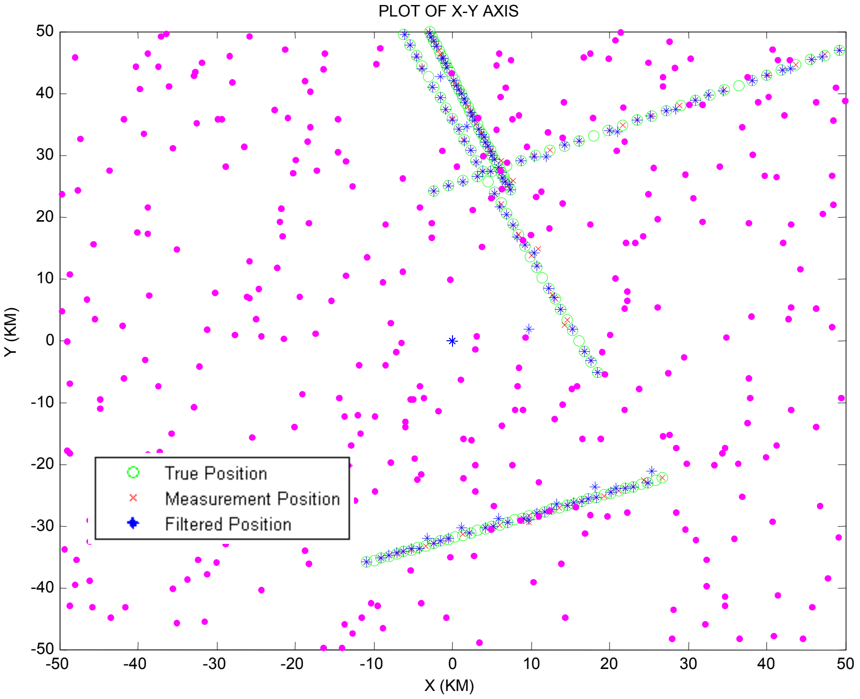


Figure 2.2. Plot of the Targets' X and Y positions

The average position error values of the targets are given in Table 2.3.

Table 2.3. Average Position Error Values of the Targets

Index of Target	Measurement Error (m)	Prediction Error (m)	Filtering Error (m)
Target-1	18.6	13.4	4.9
Target-2	22.5	15.8	5.3
Target-3	19.3	12.1	4.7
Target-4	17.0	11.7	4.2

In Experiment 2 the parameters of the scenario is same as the first experiment except the clutter density which is increased from $\lambda = 100$ to $\lambda = 200$. Since the clutter density is doubled, the possibility of receiving more measurements at a specific time instant is also increased. By use of these system parameters, Figure 2.3 depicts the true, the measured and the filtered positions of all targets with the clutter in the surveillance region.

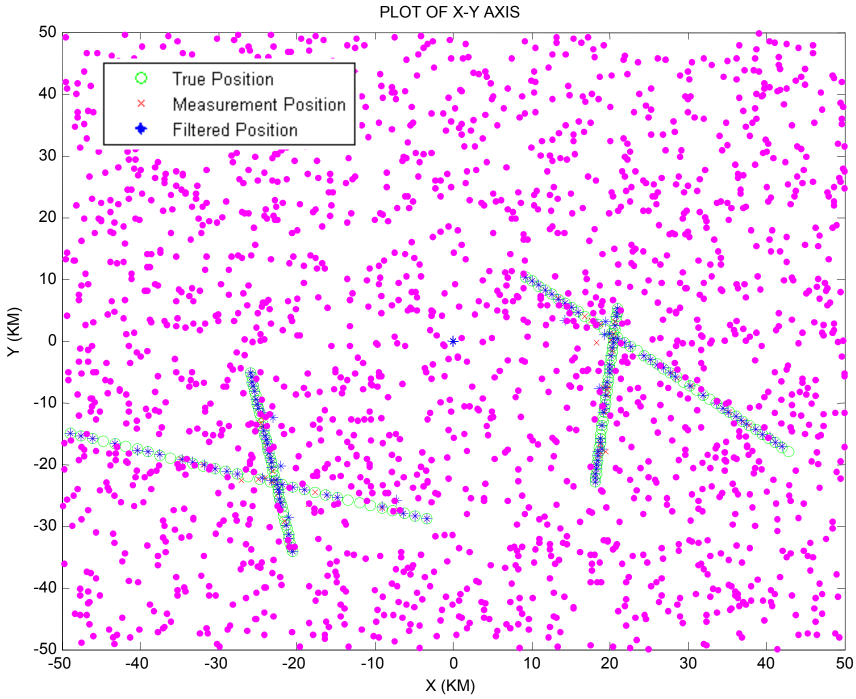


Figure 2.3. Plot of the Targets' X and Y positions in a heavy clutter

The average position error values of the targets are given in Table 2.4.

Table 2.4. Average Position Error Values of the Targets

Index of Target	Measurement Error (m)	Prediction Error (m)	Filtering Error (m)
Target-1	32.4	11.7	7.8
Target-2	28.5	10.2	5.4
Target-3	32.6	15.6	7.3
Target-4	30.2	11.3	6.8

In Experiment 3 the gate sizes are increased. The clutter density is the same as the first run. But since the gate sizes of the targets are increased, the possibility of receiving more measurements at a specific time instant is also increased.

The average position error values of the targets are given in Table 2.5.

Table 2.5. Average Position Error Values of the Targets

Index of Target	Measurement Error (m)	Prediction Error (m)	Filtering Error (m)
Target-1	41.3	23.2	14.4
Target-2	43.4	24.7	15.7
Target-3	46.6	24.2	15.2
Target-4	47.1	25.6	15.9

The statistics related with the computational load of the algorithm for this scenario is given in Table 2.6.

Table 2.6. The Statistics Related with the Computational Load of the Algorithm

Index Of Experiment	Average Number of Hypothesis Generated	Max. Number of Hypothesis Generated	Standard Variation of the Number of Hypothesis
Experiment 1	9.975	16	7.133
Experiment 2	12.05	16	6.662
Experiment 3	16.845	24	8.201

2.3.2 The Second Scenario

- The targets move according to constant velocity model.
- The starting points of motions of the targets and their velocities are chosen as specific values to ensure at least one crossing time instant at $k = 30$.

$$x_t = [x \quad y \quad z \quad \dot{x} \quad \dot{y} \quad \dot{z}]^T \qquad t = 1..N$$

$$x_1 = [0 \quad 30 \quad 0 \quad 1 \quad -1 \quad 0]^T$$

$$x_2 = [0 \quad -30 \quad 0 \quad 1 \quad 1 \quad 0]^T$$

$$x_3 = [0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0]^T$$

$$x_4 = [30 \quad 30 \quad 0 \quad 0 \quad -1 \quad 0]^T$$

- Unless otherwise stated, target detection probability is again 0.9 for all targets. This means that at a specific time instant it is possible not to receive a measurement from a specific target with a probability of 0.1.
- Targets may have different gate size. But for all time instants gate size of the target is assumed to be stationary. The gate shapes are always ellipsoidal for all time instants and for all targets.
- There always exists a heavy clutter in the system. Clutter is uniformly distributed over the surveillance region with a density determined from Poisson distribution.

2.3.1.2. Experiments

We have made four experiments in this section. In the first experiment, the gate sizes of the targets are chosen as their nominal values and clutter is uniformly distributed over the surveillance region with a density determined from Poisson distribution of parameter $\lambda = 100$. These parameters describe a nominal system. Figure 2.4 depicts true, measured and filtered positions of all targets with the clutter in the surveillance region. Figure 2.5 depicts the indexes and probabilities of hypotheses generated at time $k = 30$. Figure 2.6, 2.7, 2.8 and 2.9 depict rms position errors evaluated after we did 100 Monte Carlo runs for Target-1, Target-2, Target-3 and Taget-4 respectively.

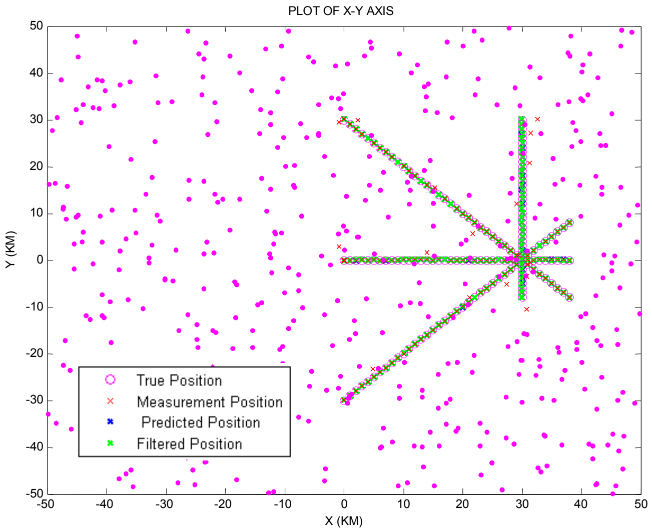


Figure 2.4. Plot of the Targets' X and Y positions

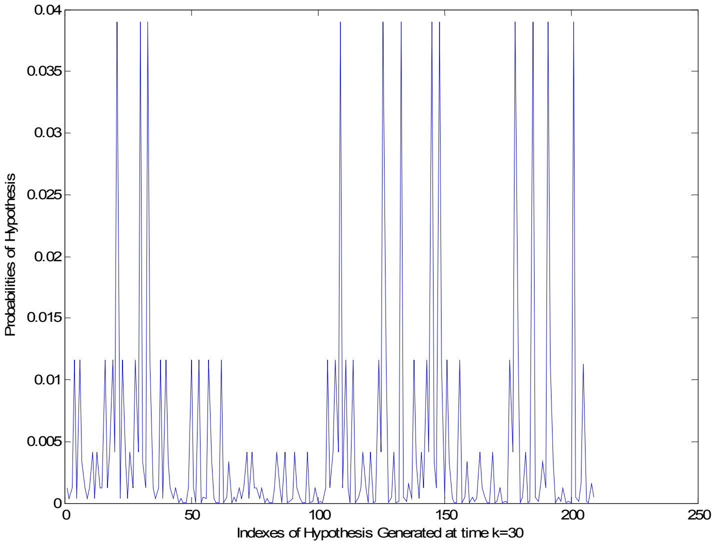


Figure 2.5. Indexes and Probabilities of Hypotheses Generated at time $k = 30$

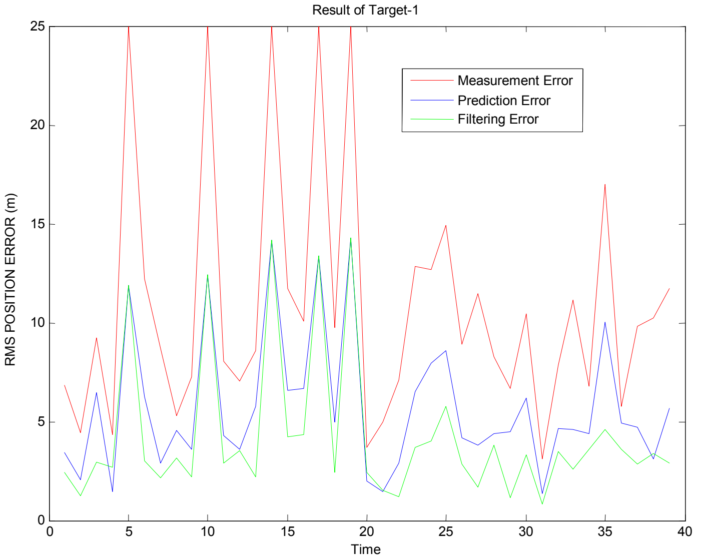


Figure 2.6. Plot of Measurement, Prediction and Filtering Position Errors of Target-1

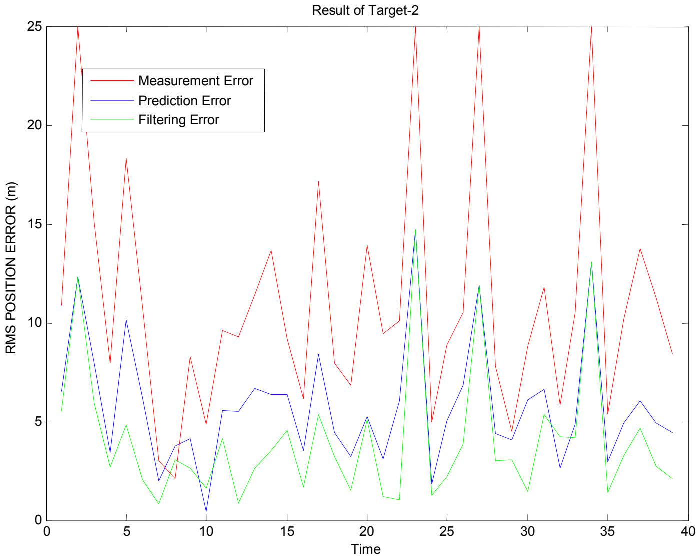


Figure 2.7. Plot of Measurement, Prediction and Filtering Position Errors of Target-2

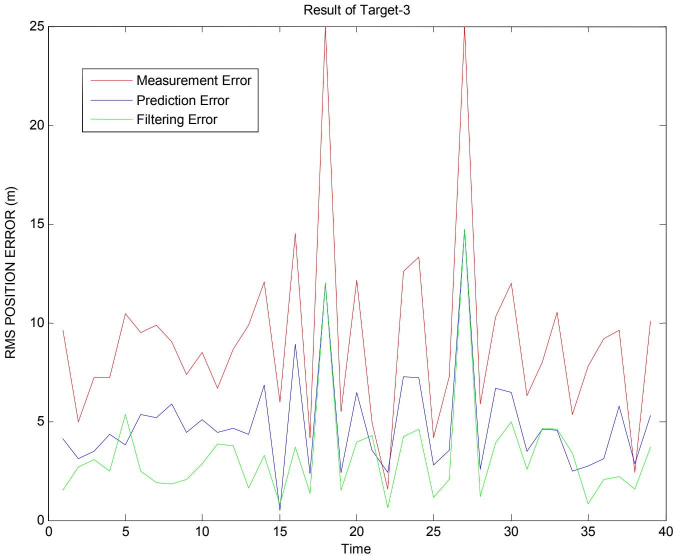


Figure 2.8. Plot of Measurement, Prediction and Filtering Position Errors of Target-3

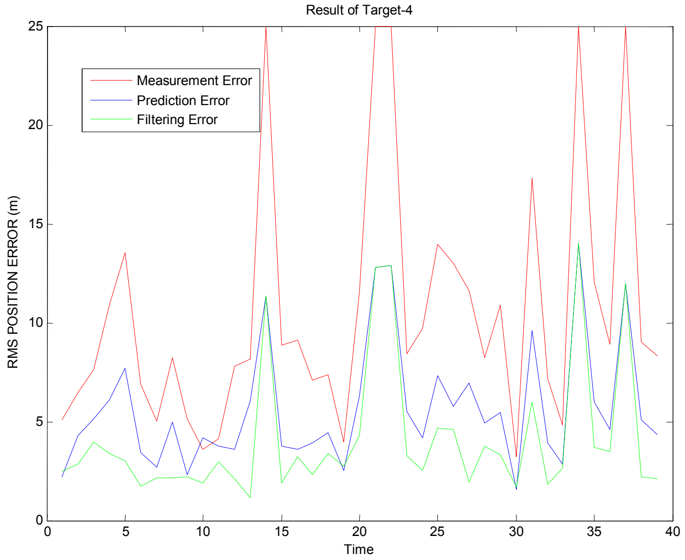


Figure 2.9. Plot of Measurement, Prediction and Filtering Position Errors of Target-4

In Experiment 2, the gate sizes are left unchanged but the clutter density is increased from $\lambda = 100$ to $\lambda = 200$. Since the clutter density is doubled, the possibility of receiving more measurements at a specific time instant is also increased. Figure 2.10 depicts the true, the measured and the filtered positions of all targets with the clutter in the surveillance region.

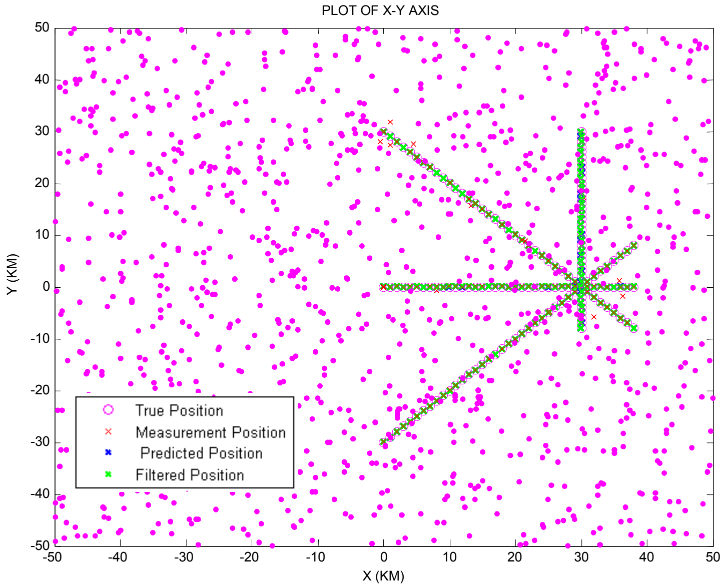


Figure 2.10. Plot of the Targets' X and Y positions in a heavy clutter

Table 2.7. Average Position Error Values of the Targets

Index of Target	Measurement Error (m)	Prediction Error (m)	Filtering Error (m)
Target-1	33.6	12.7	8.4
Target-2	27.8	11.4	7.8
Target-3	34.5	14.3	8.7
Target-4	31.2	12.3	6.9

Table 2.8. Position Error Values of the Targets at time instant $k = 30$

Index of Target	Measurement Error (m)	Prediction Error (m)	Filtering Error (m)
Target-1	25.8	8.4	4.9
Target-2	16.2	10.4	5.2
Target-3	21.9	9.2	5.4
Target-4	27.3	10.1	6.7

The algorithm gives satisfactory results at time instant $k = 30$.

In Experiment 3, the gate sizes are increased. The clutter density is the same as the first run. Since the gate sizes of the targets are increased, the possibility of receiving more measurements at a specific time instant is also increased. This means that the average position error values will be increased also.

Table 2.9. Average Position Error Values of the Targets

Index of Target	Measurement Error (m)	Prediction Error (m)	Filtering Error (m)
Target-1	42.3	25.4	18.4
Target-2	46.7	18.3	13.2
Target-3	39.8	23.9	16.9
Target-4	41.5	20.8	14.6

Table 2.10. Position Error Values of the Targets at time instant $k = 30$

Index of Target	Measurement Error (m)	Prediction Error (m)	Filtering Error (m)
Target-1	34.3	12.1	8.5
Target-2	32.4	11.5	7.4
Target-3	21.6	10.9	9.0
Target-4	28.5	19.8	10.2

The algorithm gives satisfactory results at time instant k=30. The statistics related with the computational load of the algorithm for this scenario is given in Table 2.11.

Table 2.11. The Statistics Related with the Computational Load of the Algorithm

Index Of Experiment	The number of Hypothesis Generated at time k=30	Average Number of Hypothesis Generated	Max. Number Of Hypothesis Generated	Standard Variation of the Number of Hypothesis
Experiment 1	209	16.5	209	32.396
Experiment 2	176	17.7	176	27.153
Experiment 3	258	26.85	258	43.961

In Experiment 4, we choose the gate sizes and the clutter density values are the same as the first run. But probability of target detection is decreased to 0.5 for all targets. Since the probability of target detection is decreased, the position error values are increased.

Table 2.12. Average Position Error Values of the Targets

Index of Target	Measurement Error (m)	Prediction Error (m)	Filtering Error (m)
Target-1	98.5	45.4	33.4
Target-2	96.4	43.7	25.2
Target-3	94.8	52.9	37.3
Target-4	93.2	42.4	32.6

Table 2.13. Position Error Values of the Targets at time instant $k = 30$

Index of Target	Measurement Error (m)	Prediction Error (m)	Filtering Error (m)
Target-1	64.3	22.1	15.7
Target-2	73.5	34.5	16.4
Target-3	62.4	30.3	13.2
Target-4	79.6	28.3	18.1

The algorithm gives satisfactory result at time instant k=30.

The two experiments show that JPDA filters the measurement errors for the targets moving with a constant velocity at the order of 75-50% both at prediction and filtering for the 4 target scenarios given above. These results may be considered as satisfactory for some applications. The main drawback of the algorithm is its computational complexity. The number of hypothesis may increase to a very large number when a certain number of targets are in some proximity of each other. Another important result of these experiments is that decreasing the target detection probability causes significant increase in the error values.

CHAPTER 3

A BAYESIAN APPROACH TO CLUSTER TRACKING USING RANDOM MATRICES

3.1 Introduction

Our problem is to track the trajectories formed by a fixed number of projectiles. The projectiles are fired with a specific time difference one after the other from the same weapon. There is an obvious parallelism between this problem and 'extended object-group tracking' problem. So we decided to apply a relatively new technique developed for cluster tracking [1] using random matrices to our problem. According to this technique the object extension, characterized by the projectiles, is considered as a part of the object state and is estimated jointly with the kinematical properties involved. The basic rule for the object extension is that it can be mathematically described by a symmetric and positive definite (SPD) random matrix X_k to be estimated from the sensor measurements. A simulation using the detailed nonlinear model of the ballistic motion of the group showed that they form an ellipse that changes its shape in time. This motivated the idea of using the approach of [1] to track the group of projectiles fired one after the other.

This approach, that is called 'Koch's approach' from now on, covers the following object properties:

- Object size: volume of the extension ellipsoid,
- Object shape: ratio of the corresponding semi-axes,
- Object orientation: direction of the semi-axes.

This chapter proposes a realization of the general concept given in [1] within a Bayesian framework and in Chapter 4 we give the results of our problem.

3.2 Elements of a Bayesian Solution: Koch's approach

We assume that we have a set of measurements at each time instant t_k that comes from the elements of the group that we aim to track. Let $Z_k = \{z_k^j\}_{j=1}^{n_k}$ denote the measurement set that contains n_k elements at time t_k and each individual measurement is described by the vector z_k^j. The aim of the tracking algorithm is to find an iterative updating scheme for conditional probability densities $p(x_k, X_k|Z^k)$ at each time t_k given the accumulated sensor data $Z^k = \{Z_l^{n_k}\}_{l=1}^{k}$. X_k , current object extension, in this representation denotes a positive definite random matrix that determines the 'shape' of the target cluster.

The joint density

$$p(x_k, X_k|Z^k) = p(x_k|X_k, Z^k)p(X_k|Z^k) \tag{3.1}$$

can be written as a product of two densities. The first term on the right hand side of (3.1) is a vector-variate probability density $p(x_k|X_k, Z^k)$ and it describes the kinematical object properties. The second term on the right hand side of (3.1) is a matrix-variate density $p(X_k|Z^k)$ and it describes the object extension property. Furthermore, it is simple to realize that the density $p(x_k|X_k, Z^k)$ should show an explicit dependency on the current object extension X_k.

The iterative calculation of the joint density $p(x_k|X_k, Z^k)$, basically consists of two steps: Prediction and Filtering. After applying these steps, a retrodiction step can also be applied to improve the performance of the filter.

3.2.1. Prediction Step

The iterative calculation of the joint probability density $p(x_k|X_k, Z^k)$ is preceded by a prediction step,

$$p\left(x_{k-1}, X_{k-1}\middle|Z^{k-1}\right) \xrightarrow{evolution\ models} p(x_k, X_k|Z^{k-1}) \tag{3.2}$$

based on the underlying evolution models. The prediction density $p(x_k, X_k|Z^{k-1})$ can be calculated by integration [1]:

$$p(x_k, X_k|Z^{k-1}) \\ = \int p(x_k, X_k|x_{k-1}, X_{k-1}, Z^{k-1}) p(x_{k-1}, X_{k-1}|Z^{k-1})\, dx_{k-1} dX_{k-1} \tag{3.3}$$

We try to simply the Equation (3.3). To do this we interpret the transition density which is the first term on the right hand side of the Equation (3.3) as

$$p\big(x_k, X_k\big|x_{k-1}, X_{k-1}, Z^{k-1}\big) \\ = p(x_k|X_k, x_{k-1}, X_{k-1}, Z^{k-1}) p(X_k|x_{k-1}, X_{k-1}, Z^{k-1}) \tag{3.4}$$

By using the basic Markov-type assumptions for its kinematical part we write:

$$p\big(x_k\big|X_k, x_{k-1}, X_{k-1}, Z^{k-1}\big) = p(x_k|X_k, x_{k-1}) \tag{3.5}$$

and assuming that the object's kinematical properties have no impact on the temporal evolution of the object extension and the previous measurements if X_{k-1} is given, we can write

$$p\big(X_k\big|x_{k-1}, X_{k-1}, Z^{k-1}\big) = p(X_k|X_{k-1}) \tag{3.6}$$

Using (3.5) and (3.6) we can rewrite the Equation (3.4) as

$$p\big(x_k, X_k\big|x_{k-1}, X_{k-1}, Z^{k-1}\big) = p(x_k|X_k, x_{k-1}) p(X_k|X_{k-1}) \tag{3.7}$$

Equation (3.7) clearly indicates that $p(x_k|X_k, x_{k-1})$ is affected by the current object extension X_k as well.

We interpret the second term on the right hand side of the Equation (3.3) as

$$p\big(x_{k-1}, X_{k-1}\big|Z^{k-1}\big) = p\big(x_{k-1}\big|X_{k-1}, Z^{k-1}\big) p\big(X_{k-1}\big|Z^{k-1}\big) \tag{3.8}$$

and together with the Equation (3.7), we obtain the following prediction formula:

$$p\big(x_k, X_k\big|Z^{k-1}\big) = \\ \int p(x_k|X_k, x_{k-1}) p(X_k|X_{k-1}) p\big(x_{k-1}\big|X_{k-1}, Z^{k-1}\big) p\big(X_{k-1}\big|Z^{k-1}\big)\; dx_{k-1} dX_{k-1} \tag{3.9}$$

To simplify the filter derivations an additional assumption is done at this point as follows. We assume that $p(x_{k-1}|X_{k-1}, Z^{k-1}) \approx p(x_{k-1}|X_k, Z^{k-1})$ or, in other words, we replace X_{k-1} by X_k. By this approximation, the predicted density

$$p(x_k, X_k|Z^{k-1}) = p(x_k|X_k, Z^{k-1})p(X_k|Z^{k-1}) \tag{3.10}$$

is given by two factors to be obtained by independent integrations [1]:

$$p(x_k|X_k, Z^{k-1}) = \int p(x_k|X_k, x_{k-1})p(x_{k-1}|X_k, Z^{k-1})\, dx_{k-1} \tag{3.11}$$

$$p(X_k|Z^{k-1}) = \int p(X_k|X_{k-1})p(X_{k-1}|Z^{k-1})\; dX_{k-1} \tag{3.12}$$

3.2.2. Filtering Step

The prediction is followed by a filtering step as follows:

$$p(x_k, X_k|Z^{k-1}) \xrightarrow{sensor\ model} p(x_k, X_k|Z^k) \tag{3.13}$$

The current data together with the underlying sensor model forms the sensor specific likelihood function $p(Z_k, n_k|x_k, X_k)$. This function is combined with the predicted density and by the help of Bayes' formula we obtain the Equation (3.14).

$$p(x_k, X_k|Z^k) = \frac{p(Z_k, n_k|x_k, X_k)p(x_k, X_k|Z^{k-1})}{\int p(Z_k, n_k|x_k, X_k)p(x_k, X_k|Z^{k-1})dx_k dX_k} \tag{3.14}$$

3.3 Retrodiction Step

This step is a backward-directed iterative calculation of the probability densities $p\{x_l\,, X_l|Z^k\,\}$ from $p\{x_{l+1}\,, X_{l+1}|Z^k\,\}$ where $l < k$. The detailed formulas will be given later in this chapter.

3.4 Extended Object Tracking

The kinematical state variable x_k at time t_k is given by $x_k = (r_k^T, \dot{r}_k^T, \ddot{r}_k^T)^T$. Here r_k is the spatial state component and $\dot{r}_k, \ddot{r}_k$ denote the corresponding velocity and acceleration components. Let the dimension d of the vector r_k be also the dimension of the dxd SPD matrix X_k describing the current ellipsoidal object extension. The dimension of the kinematical state vector x_k is thus $s\ x\ d$, where $s - 1$ describes up to which derivative the object kinematics is modeled [1]. Here we have $s = 3$ and the dimension of the vector r_k is $d = 3$.

3.4.1. Object Dynamics Model

The temporal evolution of an extended or collective object is modeled as a linear stochastic system:

$$x_k = E_{k|k-1} x_{k-1} + v_k, \quad p(v_k) = N(v_k; 0, G_{k|k-1}) \tag{3.15}$$

Using the Kronecker product[1] the evolution matrix $E_{k|k-1}$ can be written as

$$E_{k|k-1} = F_{k|k-1} \otimes I_d \tag{3.16}$$

where the $s\ x\ s$ matrix $F_{k|k-1}$ is given by

$$F_{k|k-1} = \begin{pmatrix} 1 & t_k - t_{k-1} & \frac{1}{2}(t_k - t_{k-1})^2 \\ 0 & 1 & t_k - t_{k-1} \\ 0 & 0 & e^{-(t_k - t_{k-1})/\theta} \end{pmatrix} \tag{3.17}$$

For the dynamics noise covariance $G_{k|k-1}$ we define the following structure:

$$G_{k|k-1} = D_{k|k-1} \otimes X_k \tag{3.18}$$

[1]: Kronecker product is defined in the Appendix A.

where the $s\ x\ s$ matrix $D_{k|k-1}$ is given by Equation (3.19)

$$D_{k|k-1} = \mu^2(1 - e^{-2(t_k - t_{k-1})/\theta})\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{3.19}$$

with the scalar acceleration rms value μ and the maneuver correlation time constant θ.

The model described here and used in the simulations is called the 'Singer model' in the tracking literature.

3.4.2. Prediction

Kinematical Part:

Let us remember the prediction formulae derived in Section 3.2.1

$$p(x_k|X_k, Z^{k-1}) = \int p(x_k|X_k, x_{k-1})p(x_{k-1}|X_k, Z^{k-1})\, dx_{k-1} \tag{3.20}$$

The second term written in the right hand side of the integration is assumed to be Gaussian with the following special structure [1]:

$$p(x_{k-1}|X_k, Z^{k-1}) = N(x_{k-1}; x_{k-1|k-1}, P_{k-1|k-1} \otimes X_k) \tag{3.21}$$

By using the previous evolution model, Equation (3.20) can be rewritten as

$$\begin{aligned} &p(x_k|X_k, Z^{k-1}) \\ &= \int N(x_k; (F_{k|k-1} \otimes I_d)x_{k-1}, D_{k|k-1} \otimes X_k)N(x_{k-1}; x_{k-1|k-1}, P_{k-1|k-1} \otimes X_k)dx_{k-1} \\ &= N(x_k; x_{k|k-1}, P_{k|k-1} \otimes X_k) \end{aligned} \tag{3.22}$$

where $x_{k|k-1}$ and $P_{k|k-1}$ are given by

$$x_{k|k-1} = (F_{k|k-1} \otimes I_d)x_{k-1|k-1} \tag{3.23}$$

$$P_{k|k-1} = F_{k|k-1}P_{k-1|k-1}F_{k|k-1}^T + D_{k|k-1} \tag{3.24}$$

which is very close analogy to standard Kalman filtering [1].

Object Extension Part:

Let us remember the prediction formulae derived in Section 3.2.1

$$p(X_k|Z^{k-1}) = \int p(X_k|X_{k-1})p(X_{k-1}|Z^{k-1})\ dX_{k-1} \tag{3.25}$$

The second term written in the right hand side of the integration is the density of the object extension state variable and is given by the inverted Wishart density[2]:

$$\begin{aligned} p(X_{k-1}|Z^{k-1}) &= IW(X_{k-1}; v_{k-1|k-1}, X_{k-1|k-1}) \\ &\propto |X_{k-1}|^{-(1/2)v_{k-1|k-1}} etr[-\tfrac{1}{2}X_{k-1|k-1}X_{k-1}^{-1}] \end{aligned} \tag{3.26}$$

where etr[A] is an abbreviation for exp[tr A] with tr A denoting the trace of the matrix A. Here $v_{k-1|k-1}$ is a scalar parameter and $X_{k-1|k-1}$ is a dxd matrix. These parameters determine the shape of the object extension part. The expectation of $X_{k-1}$is given by

$$E[X_{k-1}] = \frac{X_{k-1|k-1}}{v_{k-1|k-1} - 2d - 2} \tag{3.27}$$

In the prediction step the parameters $v_{k|k-1}, X_{k|k-1}$ defining $p(X_k|Z^{k-1})$ have to be calculated from $v_{k-1|k-1}, X_{k-1|k-1}$. It is not easy to obtain the update formulas for the estimated parameters $X_{k|k-1}$ and $v_{k|k-1}$. A heuristic approach [1] gives the following prediction update equations with a temporal decay constant τ as an additional modeling parameter:

$$v_{k|k-1} = e^{-\Delta t_k/\tau} v_{k-1|k-1} \tag{3.28}$$

$$X_{k|k-1} = \frac{e^{-\Delta t_k/\tau} v_{k-1|k-1} - d - 1}{v_{k-1|k-1} - d - 1} X_{k-1|k-1} \tag{3.29}$$

[2] : Inverted Wishart density is defined in the Appendix B.

$\tau = \infty$ represents a static object or group extension [1]. Note that in this formulation the matrices $X_{k|k-1}$ and $X_{k-1|k-1}$ are different only in the scale. This may cause a drawback in the application.

The first term written in the right hand side of the integration in (3.25) is the transition density of the object extension and is given by Wishart densities[3]. We thus have

$$p(X_k|X_{k-1}) = W\left(X_k; \delta_{k|k-1}, \frac{X_{k-1}}{\delta_{k|k-1}}\right)$$

$$\propto |X_{k-1}|^{-(\delta_{k|k-1})/2} |X_k|^{(\delta_{k|k-1}-d-1)/2} etr[-\tfrac{1}{2}\delta_{k|k-1} X_k X_{k-1}^{-1}] \tag{3.30}$$

Here the term $\delta_{k|k-1}$ is a scalar parameter and X_{k-1} is a $d\ x\ d$ matrix.

The following functional form for $\delta_{k|k-1}$ is given in [1].

$$\delta_{k|k-1} = \delta e^{-(t_k - t_{k-1})/\tau} \tag{3.31}$$

which is defined by two extension evolution parameters δ and τ. We use the product formula for Wishart and inverted Wishart densities and multiply the Equations (3.26) and (3.30) to calculate the predicted density $p(X_k|Z^{k-1})$. We end up with "Generalized Beta Type II" density[4] as follows [1]:

$$p(X_k|Z^{k-1})$$

$$= \int W\left(X_k; \delta_{k|k-1}, \frac{X_{k-1}}{\delta_{k|k-1}}\right) IW\left(X_{k-1}; v_{k-1|k-1}, X_{k-1|k-1}\right) dX_{k-1} \tag{3.32}$$

$$= B(X_k; \tfrac{1}{2}\delta_{k|k-1}; \tfrac{1}{2}(v_{k-1|k-1} - d - 1), X_{k-1|k-1}/\delta_{k|k-1})$$

[3] : Wishart density is defined in the Appendix B.

[4] : Generalized Beta Type II density is defined in the Appendix B.

Here the terms $\frac{1}{2}\delta_{k|k-1}$ and $\frac{1}{2}(v_{k-1|k-1} - d - 1)$ are scalar parameters and X_{k-1} is a dxd matrix. To complete the cycle this density is approximated as inverted Wishart density.

3.4.3. Filtering

To this end, we have explained the prediction step briefly for both kinematical part and the object extension part. In this section we are going to give the details of the filtering step. First of all let us remember the formulae derived in Section 3.2.2, which was

$$p(x_k, X_k | Z^k) = \frac{p(Z_k, n_k | x_k, X_k) p(x_k, X_k | Z^{k-1})}{\int p(Z_k, n_k | x_k, X_k) p(x_k, X_k | Z^{k-1}) dx_k dX_k} \tag{3.33}$$

We need to define two densities to reach our goal. One of them is $p(Z_k, n_k | x_k, X_k)$ and the other is $p(x_k, X_k | Z^{k-1})$. As a first approximation $p(n_k | x_k, X_k)$ is assumed to be constant, i.e., independent of x_k, X_k which means that it is sufficient to define the joint density $p(Z_k | n_k, x_k, X_k)$ to obtain $p(Z_k, n_k | x_k, X_k)$. We interpret this density as

$$\begin{aligned} p(Z_k | n_k, x_k, X_k) &= \prod_{j=1}^{n_k} N(z_k^j; (H_k \otimes I_d) x_k, X_k) \\ &\propto N\left(z_k; (H_k \otimes I_d) x_k, \frac{X_k}{n_k}\right) LW(Z_k; n_k - 1, X_k) \end{aligned} \tag{3.34}$$

where $H_k = [1 \quad 0 \quad 0]$ which means position measurements are accessible.

The centroid measurement z_k and the corresponding scattering matrix Z_k are given by

$$z_k = \frac{1}{n_k} \sum_{j=1}^{n_k} z_k^j \tag{3.35}$$

$$Z_k = \sum_{j=1}^{n_k} (z_k^j - z_k)(z_k^j - z_k)^T \tag{3.36}$$

while $LW\ (Z_k; n_k - 1, X_k)$ is proportional to a Wishart density in Z_k with $n_k - 1$ degrees of freedom [1]:

$$LW(Z_k; n_k - 1, X_k) := |X_k|^{-(1/2)(n_k-1)} etr\left[-\tfrac{1}{2} Z_k X_k^{-1}\right] \tag{3.37}$$

To obtain $p\left(x_k, X_k \middle| Z^{k-1}\right)$ we use the formula derived in Equation (3.10) and we need to multiply the Equations (3.22) and (3.32). Together with the result of (3.34) we obtain the following formula:

$$\begin{aligned} & p(Z_k|n_k, x_k, X_k)\, p\left(x_k, X_k \middle| Z^{k-1}\right) \\ & \propto N\left(z_k; (H_k \otimes I_d)x_k, \frac{X_k}{n_k}\right) N\left(x_k; x_{k|k-1}, P_{k|k-1} \otimes X_k\right) \\ & \quad .LW\ (Z_k; n_k - 1, X_k) IW(X_k; v_{k|k-1}, X_{k|k-1}) \end{aligned} \tag{3.38}$$

Kinematical part:

The product of the two Gaussians in the previous equation yields

$$\begin{aligned} & N\left(z_k; (H_k \otimes I_d)x_k, \frac{X_k}{n_k}\right) N\left(x_k; x_{k|k-1}, P_{k|k-1} \otimes X_k\right) \\ & = N\left(z_k; (H_k \otimes I_d)x_{k|k-1}, S_{k|k-1} X_k\right) N(x_k; x_{k|k}, P_{k|k} \otimes X_k) \end{aligned} \tag{3.39}$$

where the quantities $x_{k|k}$ and $P_{k|k}$ are given by

$$x_{k|k} = x_{k|k-1} + \left(W_{k|k-1} \otimes I_d\right)(z_k - (H_k \otimes I_d)x_{k|k-1}) \tag{3.40}$$

$$P_{k|k} = \ P_{k|k-1} - W_{k|k-1} S_{k|k-1} W_{k|k-1}^T \tag{3.41}$$

with a scalar innovation factor and a gain matrix defined by

$$S_{k|k-1} = H_k P_{k|k-1} H_k^T + \frac{1}{n_k} \tag{3.42}$$

$$W_{k|k-1} = P_{k|k-1} H_k^T {S_{k|k-1}}^{-1} \tag{3.43}$$

Object Extension Part:

Since the object extension, represented by X_k, is independent of the kinematical state variable x_k, $N\left(z_k; (H_k \otimes I_d)x_{k|k-1}, S_{k|k-1} X_k\right)$ can be rewritten as

$$N\big(z_k; (H_k \otimes I_d)x_{k|k-1}, S_{k|k-1}X_k\big) \propto |X_k|^{-1/2} etr[-\tfrac{1}{2}N_{k|k-1}X_k^{-1}] \tag{3.44}$$

with an innovation matrix $N_{k|k-1}$ defined

$$N_{k|k-1} = S_{k|k-1}{}^{-1}(z_k - (H_k \otimes I_d)x_{k|k-1})(z_k - (H_k \otimes I_d)x_{k|k-1})^T \tag{3.45}$$

The remaining two factors on the right side of the Equation (3.38) yield

$$\begin{aligned} LW(Z_k; n_k - 1, X_k)\ IW\ \big(X_k; v_{k|k-1}, X_{k|k-1}\big)|X_k|^{-1/2} etr\left[-\tfrac{1}{2}N_{k|k-1}X_k^{-1}\right] \\ \propto IW(X_k; v_{k|k}, X_{k|k}) \end{aligned} \tag{3.46}$$

with the update equations:

$$X_{k|k} = X_{k|k-1} + N_{k|k-1} + Z_k \tag{3.47}$$

$$v_{k|k} = v_{k|k-1} + n_k \tag{3.48}$$

Joint Density after Filtering:

The probability density function of the joint state (x_k, X_k) after processing the current sensor data Z_k at time t_k is thus given by

$$p\big(x_k, X_k\big|Z^k\big) = N\ \big(x_k; x_{k|k}, P_{k|k} \otimes X_k\big) IW(X_k; v_{k|k}, X_{k|k}) \tag{3.49}$$

3.4.4. Retrodiction

The joint retrodiction density at time t_l, $l < k$, is given by

$$p\big(x_l, X_l\big|Z^k\big) = N(x_l; x_{l|k}, P_{l|k} \otimes X_l) W(X_l; v_{l|k}, X_{l|k}) \tag{3.50}$$

with

$$x_{l|k} = x_{l|l} + \big(W_{l|l+1} \otimes I_d\big)(x_{l+1|k} - x_{l+1|l}) \tag{3.51}$$

$$P_{l|k} = P_{l|l} + W_{l|l+1}(P_{l+1|k} - P_{l+1|l})W_{l|l+1}^T \tag{3.52}$$

where

$$W_{l|l+1} = P_{k-1|k-1}F_{k|k-1}^T P_{k|k-1}^{-1} \tag{3.53}$$

These results are obtained by some approximations and assumptions. Applying the retrodiction step improves the performance of the filter.

3.4.5. Summary

The “pseudo code” of the algorithm as applied to the tracking of the subsequent projectiles is given in Table 3.1 below.

Table 3.1. Pseudo Code of the Koch Approach Algorithm

```
[{$x_{k|k}$ ,$P_{k|k}$ ,$X_{k|k}$ ,$v_{k|k}$}] =FILTER [{$x_{k-1|k-1}$ ,$P_{k-1|k-1}$ ,$X_{k-1|k-1}$ ,$v_{k-1|k-1}$}, $Z^k$]

INITIATION:
Choose proper values for system parameters such as $\Delta t_k, \tau, \theta$ and $\mu$
Initiate properly: $x_{0|0}$ , $P_{0|0}$ , $v_{0|0}$ , $X_{0|0}$

FOR $k = 1 : N$
STEP-1: PREDICTION
 Kinematical part:      $p(x_k | X_k, Z^{k-1}) = N(x_k; x_{k|k-1}, P_{k|k-1} \otimes X_k)$

                        $x_{k|k-1} = (F_{k|k-1} \otimes I_d) x_{k-1|k-1}$

                        $P_{k|k-1} = F_{k|k-1} P_{k-1|k-1} F_{k|k-1}^T + D_{k|k-1}$

 where                  $F_{k|k-1} = \begin{pmatrix} 1 & t_k - t_{k-1} & \frac{1}{2}(t_k - t_{k-1})^2 \\ 0 & 1 & t_k - t_{k-1} \\ 0 & 0 & e^{-(t_k - t_{k-1})/\theta} \end{pmatrix}$

                        $D_{k|k-1} = \mu^2 (1 - e^{-2(t_k - t_{k-1})/\theta}) \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

 Object Extension Part: $v_{k|k-1} = e^{-\Delta t_k/\tau} v_{k-1|k-1}$

                        $X_{k|k-1} = \frac{e^{-\Delta t_k/\tau} v_{k-1|k-1} - d - 1}{v_{k-1|k-1} - d - 1} X_{k-1|k-1}$
```

Table 3.2. Pseudo Code of the Koch Approach Algorithm (cont'd)

```
STEP-2: FILTERING
  Kinematical Part        $p(x_k|X_k, Z^k) = N(x_k; x_{k|k}, P_{k|k} \otimes X_k)$
      $x_{k|k} = x_{k|k-1} + (W_{k|k-1} \otimes I_d)(z_k - (H_k \otimes I_d)x_{k|k-1})$
      $P_{k|k} = P_{k|k-1} - W_{k|k-1} S_{k|k-1} W_{k|k-1}^T$
  Object Extension Part   $X_{k|k} = X_{k|k-1} + N_{k|k-1} + Z_k$
      $v_{k|k} = v_{k|k-1} + n_k$
  where
      $z_k = \frac{1}{n_k} \sum_{j=1}^{n_k} z_k^j$
      $Z_k = \sum_{j=1}^{n_k} (z_k^j - z_k)(z_k^j - z_k)^T$
      $S_{k|k-1} = H_k P_{k|k-1} H_k^T + \frac{1}{n_k}$
      $W_{k|k-1} = P_{k|k-1} H_k^T {S_{k|k-1}}^{-1}$
      $N_{k|k-1} = {S_{k|k-1}}^{-1}(z_k - (H_k \otimes I_d)x_{k|k-1})(z_k - (H_k \otimes I_d)x_{k|k-1})^T$
STEP-3:   $p(x_l, X_l|Z^k)$ = RETRODICTION  $p(x_{l+1}, X_{l+1}|Z^k)$
      $x_{l|k} = x_{l|l} + (W_{l|l+1} \otimes I_d)(x_{l+1|k} - x_{l+1|l})$
      $P_{l|k} = P_{l|l} + W_{l|l+1}(P_{l+1|k} - P_{l+1|l})W_{l|l+1}^T$
STEP-4: Use $x_{k|k}$ to find out the centroid position of the group
STEP-5: Use $X_{k|k}$ to draw the ellipsoidal shape of the cluster
END FOR
```

CHAPTER 4

RESULTS

The aim of this chapter is to analyze the performance and the behavior of the Koch Approach [1] on some examples and to make a comparison between the two approaches, JPDA and Koch. We have generated two different scenarios for this purpose. The first scenario investigates whether the Koch approach works properly or not for our problem. To do this we analyze the center position and the shape of the motion of the group. For the second scenario, we try to make a comparison between the two approaches Koch and JPDA.

4.1 First Scenario

In this section we used the data generated according to the nonlinear model of a ballistic target explained in Chapter 1. We consider the trajectory data of ten (10) projectiles. The projectiles are fired with a specific time difference (400 fire per minute) one after the other from the same weapon. Trajectories were obtained with a T = 0.01 s update interval. Appropriate measurement noise sequences for the range, azimuth and elevation angles were added to generate realistic observations of the trajectories of the projectiles. Each projectile is launched from the ground (x(0) = 0, y(0) = 0, z(0) = 0) with a certain muzzle velocity. Figure 4.1 shows a sample mortar trajectory with the impact point.

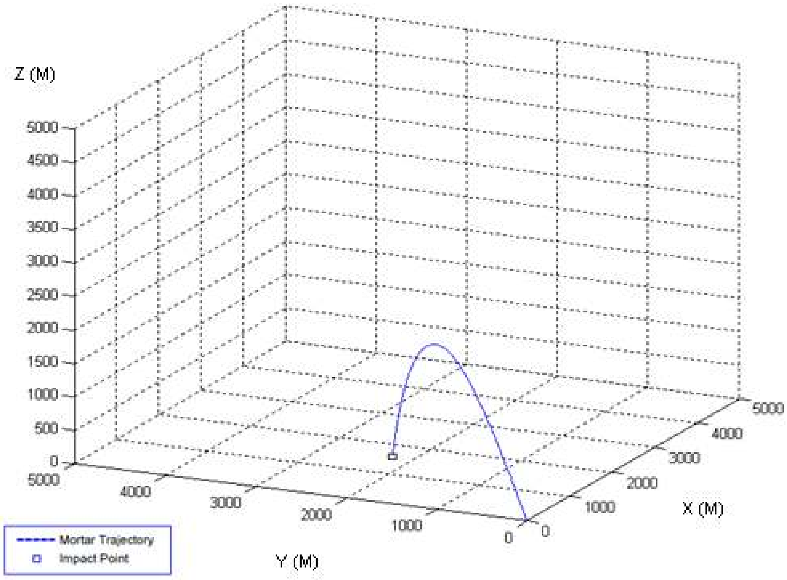


Figure 4.1. Sample Mortal Trajectory with the Impact Point

The object extension in the Koch's approach is modeled by symmetric, positive-definite random matrices. The choice of proper initial conditions for the random matrix related with the object extension is very critical for this algorithm. If not selected properly, the object extension part will always create problems at the beginning of the motion. To obtain satisfactory results it is not sufficient to choose proper initial values but we also have to choose proper values for the system parameters such as $\Delta t_k, \tau, \theta$ and μ. To obtain satisfactory results the values of these parameters, for the data set explained in Chapter 1, are selected as $\Delta t_k = 0.01, \tau = 10, \mu = 5, \theta = 10$ after some experiments. The total flying time is 40 seconds. The first scenario is for the 'proof of the concept' so we have done some unrealistic assumptions as stated below.

Assumptions:

- n_k is constant which means that the probability of detection of the targets is always 1. Since we have 10 projectiles, $n_k = 10$ for all time instants.
- As discussed before trajectories were obtained with a T = 0.01 s update interval. Since the update interval is too short, it becomes impossible to realize the ellipsoidal shapes of the group for all time instants just in one figure. To overcome this

problem we plot the corresponding ellipses by a 2.5 second time difference. Total flying time is 40 second, so we expect to see 16 samples in total.

4.1.1. Results

The results of the application of the algorithm to a group of 10 projectiles are given in the following figures. The initial velocities of the projectiles are 250 m/s, firing rate is 400 fire per minute and the mass of the each projectile is 1 kg. The motion model used in the trajectories and all of its details are defined in Chapter 1, Section 1.3.

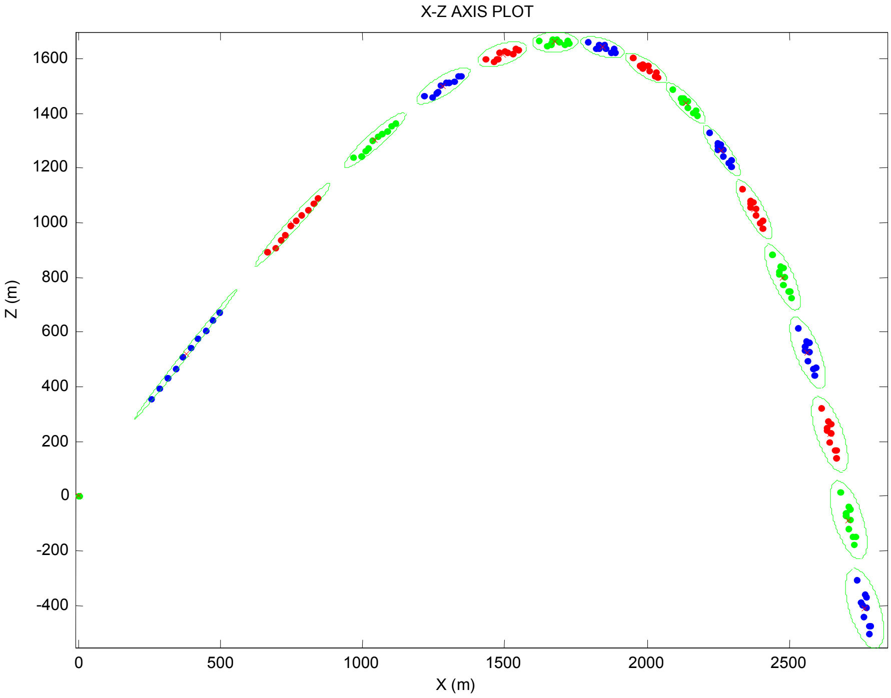
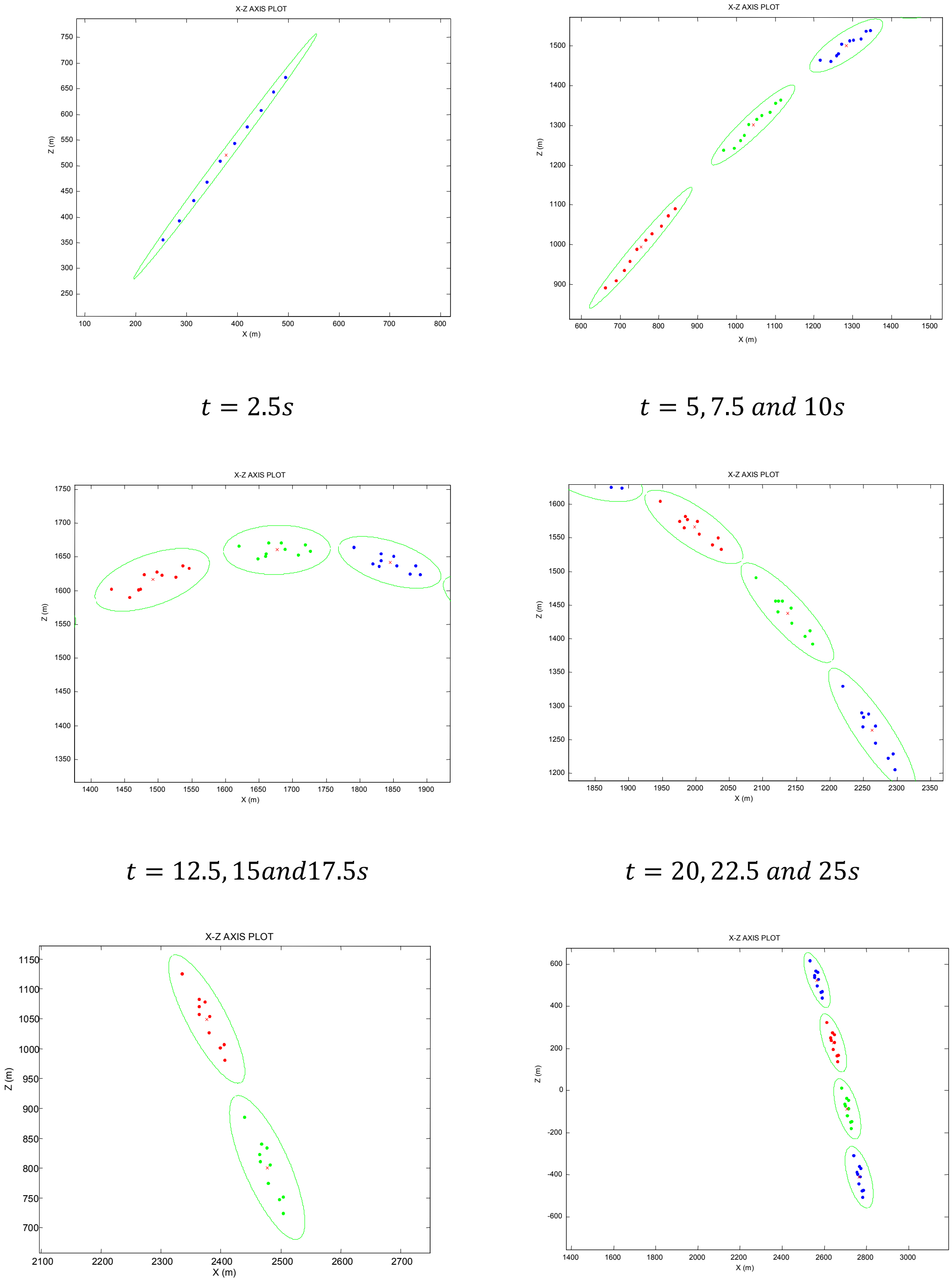


Figure 4.2. Centre Positions and Ellipsoidal Shapes of the Clusters in the X-Z Plane

Figure 4.2 shows the entire trajectory of the group on the X-Z plane and its tracking by ellipsoids in the time interval [0sec-40sec]. Not all samples are shown in this figure because of the reason explained before. The details related with the tracking performance are shown in the Figures 4.3. The details show the formation and its tracking at different times so at different locations of the group trajectory.

Figures 4.4 and Figures 4.5 show the entire trajectory and its details on the X-Y and Y-Z planes.



$t = 2.5s$

$t = 5, 7.5\ and\ 10s$

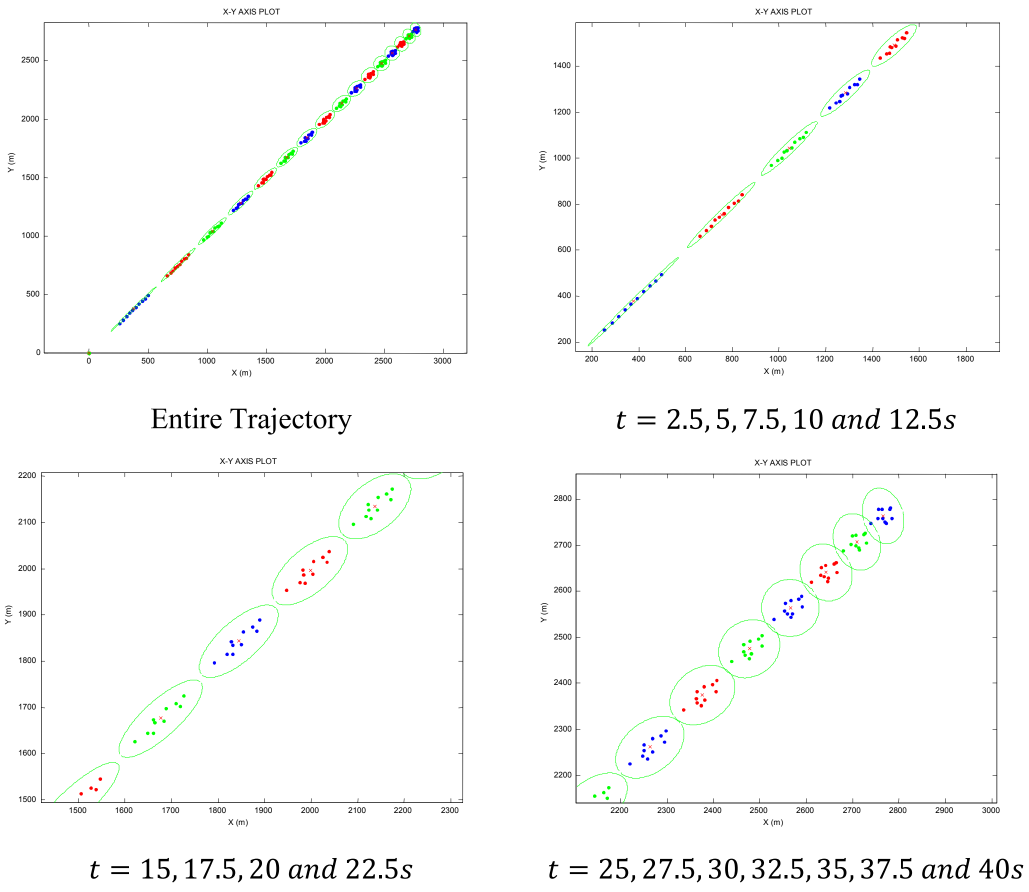
$t = 12.5, 15and17.5s$

$t = 20, 22.5\ and\ 25s$

$t = 27.5\ and\ 30s$

$t = 32.5, 35, 37.5\ and40s$

Figure 4.3. Centre Positions and Ellipsoidal Shapes of the Clusters in the X-Z plane



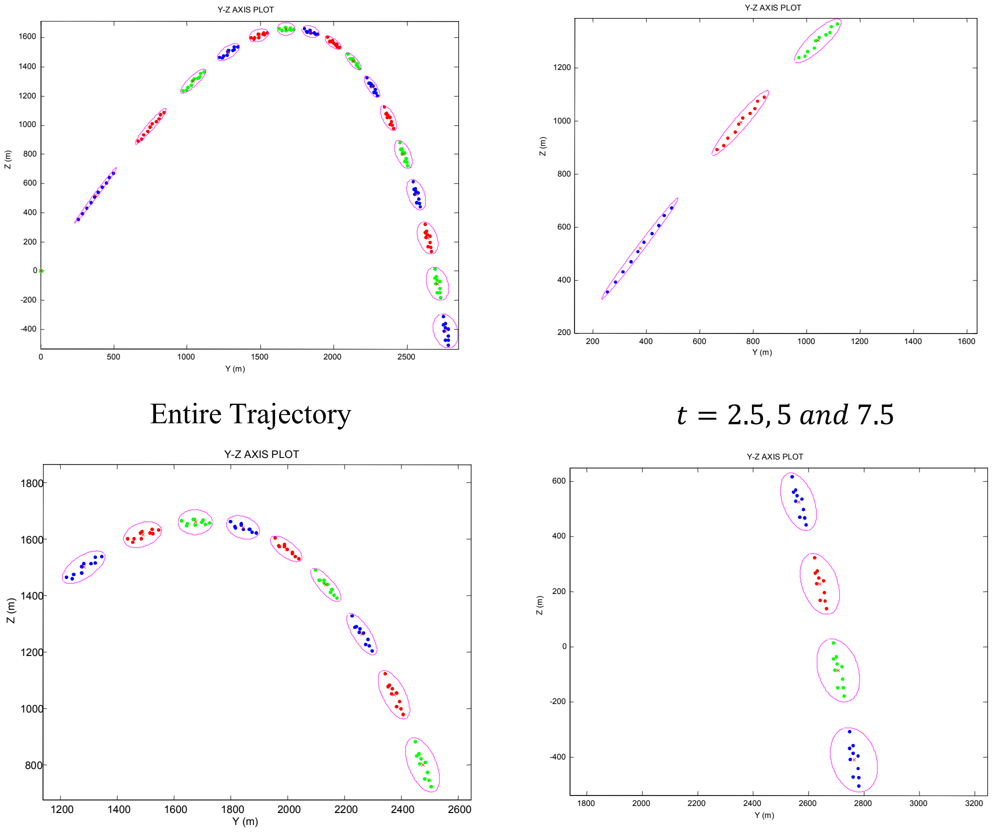
Entire Trajectory

$t = 2.5, 5, 7.5, 10\ and\ 12.5s$

$t = 15, 17.5, 20\ and\ 22.5s$

$t = 25, 27.5, 30, 32.5, 35, 37.5\ and\ 40s$

Figure 4.4. Center Positions and Ellipsoidal Shapes of the Clusters in the X-Y Plane



Entire Trajectory

$t = 2.5, 5\ and\ 7.5$

$t = 10$ through $30s$

$t = 32.5, 35, 37.5\ and\ 40s$

Figure 4.5. Center Positions and Ellipsoidal Shapes of the Clusters in the Y-Z Plane

The simulation given above shows the change in the shape of the group and its tracking by the ellipsoids. An investigation of the results shows that the ellipsoids deviate from the actual distribution of the projectiles during the final phase of their trajectory especially in the X-Y and X-Z planes. The formulation here assumes that the ellipsoid shape represented by the random matrix X is assumed to be constant in its shape during prediction. This assumption may be modified to get better results of the shape.

4.2 Second Scenario

In Chapter 2, the JPDAF algorithm is applied to two different scenarios. The algorithm proved to give good results for clusters containing up to 4 targets. When the number of targets in the cluster exceeds this limit, however, JPDA becomes infeasible for a MATLAB application since the total number of hypotheses increases exponentially and JPDAF requires prohibitive amount of processor time to compute all joint probabilities.

In Chapter 3, "*The Bayesian approach to extended object and cluster tracking using random matrices"* has been explained and this approach has been illustrated by a simulation example in Section 4.1. In this section, our aim is to compare the two approaches. For both of the algorithms we use the same trajectories generated according to the nonlinear model of a ballistic target explained in the Chapter 1 to make a fair comparison. Both approaches use the same dynamic model of the target during tracking process. Measurement models are also same for both of them.

The details related with the experiments are as follows.

- The total number of targets and the total number of time instants are decided to be the same as the values used in the JPDAF algorithm. Increasing these numbers increases the processor time to a value which makes the computation infeasible.
- We consider the trajectory data from four (4) projectiles. The projectiles are fired with a specific time difference one after the other from the same weapon. Trajectories were simulated with a T = 0.01 second interval. Total flying time is 40 seconds which means we have 4000 samples. We assume that the trajectory is sampled with a sampling period of 1 second so only 40 of them are used during tracking because of the time restriction explained in JPDAF algorithm. This assumption is also realistic for certain type of radars.
- We use the nominal values of the system parameters such as process noise and clutter density for JPDAF algorithm. The "nominal values" of these parameters are described in Chapter 2.
- Similar to JPDA, the nominal values of the system parameters such as scalar acceleration rms value μ, maneuver correlation time constant θ, update interval and temporal decay constant τ are used in the Koch approach. The nominal values of these parameters are explained in Section 4.1 of this chapter.
- In Koch's approach, the outputs of the algorithm are the center positions and ellipsoidal shape of the cluster. To compare the two algorithms, we find out the centre positions of the clusters in JPDAF algorithm. The simplest way of finding these centers is taking the average of the filtered positions of the targets. If no

measurement is associated with a specific target at a specific time instant, we use the predicted position of that specific target. To illustrate this, suppose that at a specific time instant the measurements are received from Target-2, Target-3 and Target-4 but no measurement is received from Target-1. Let;

$\hat{f}_1$ be the predicted position of Target-1

f_2 be the filtered position of Target-2

f_3 be the filtered position of Target-3

f_4 be the filtered position of Target-4

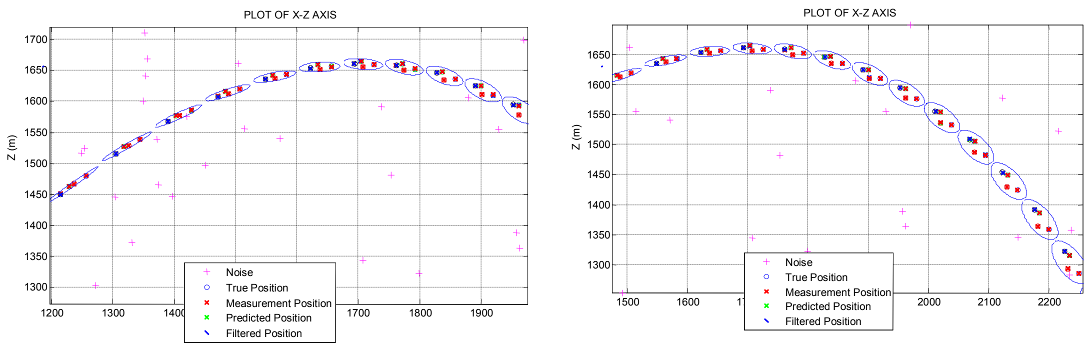
By taking the average of these positions, one can find out the centre position of the cluster at that specific time instant.

$$f = \frac{\hat{f}_1 + f_2 + f_3 + f_4}{n_k}$$

- In Section 4.1, it is assumed that n_k is constant which means probability of detection of the targets is always 1. In this section, we expect to see the effect of changing the target detection probability on the performance of the two algorithms. Decreasing P_{DE} may cause no measurement, i.e., $n_k = 0,$ at some time instants. This creates a numerical problem for the expressions containing the term $1/n_k$ (Equation 3.35 and 3.42). To deal with this problem we propose to use the predicted center position $(4 - n_k)$ times. Suppose that at a specific time instant we have received two measurements, $n_k = 2$. Let these measurements be z_k^1 and z_k^2. With this new proposed method z_k^3 and z_k^4 are exactly the same and are selected as equal to the predicted center positions. So we may give these 4 measurements as an input to the Equation (3.35).

$$z_k = \frac{1}{n_k} \sum_{j=1}^{n_k} z_k^j$$

In this section we start by setting $P_{DE} = 1$ and decrease it by 0.1 steps until we get $P_{DE} = 0.1.$ For each step we did 100 Monte Carlo runs. We plot the ellipsoidal shapes of the clusters obtained according to the Koch's Approach for one selected run for each value of P_{DE}. Additionally, when $P_{DE} = 0.9$ and $P_{DE} = 0.5$ we plot the measurement, predicted and filtered positions of the targets evaluated according to JPDA approach (Figures 4.14 and 4.24). In Figure 4.40, we show the performance result of the two methods.



First part of the Trajectory

Second part of the Trajectory

Figure 4.6. Ellipsoidal Shapes of the Clusters according to Koch approach with $P_{DE} = 1$

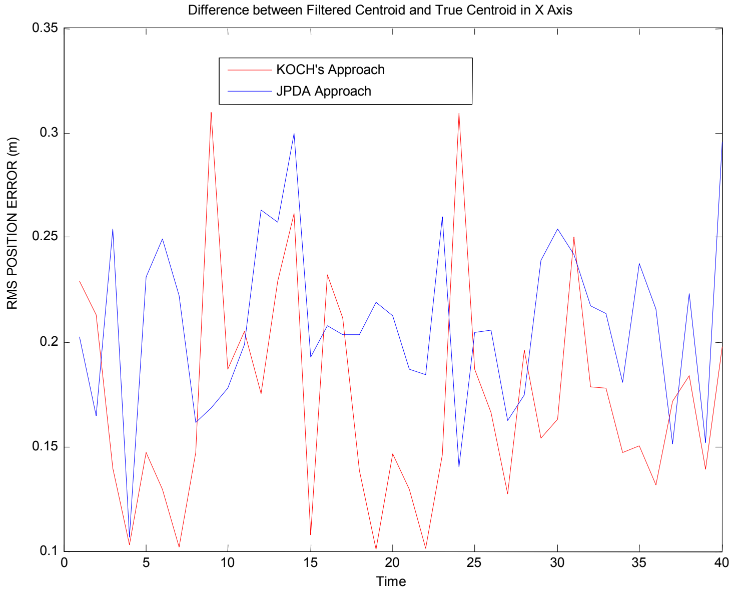


Figure 4.7. The Center Position Errors over time in the X axis with $P_{DE} = 1$

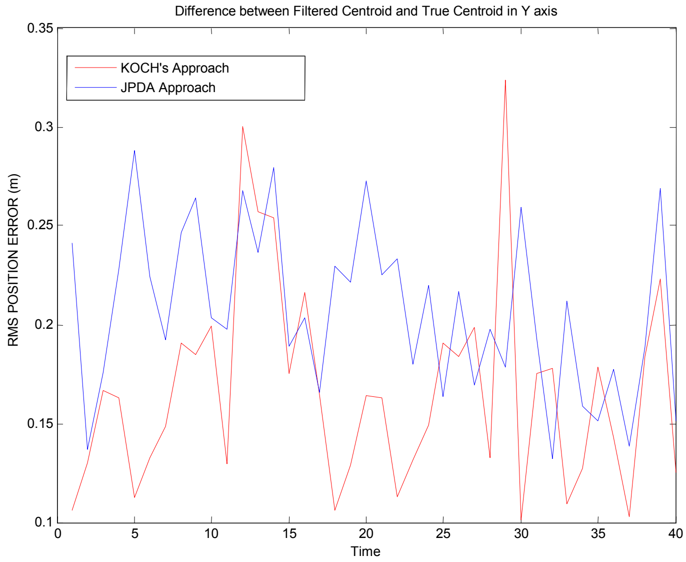


Figure 4.8. The Center Position Errors over time in the Y axis with $P_{DE} = 1$

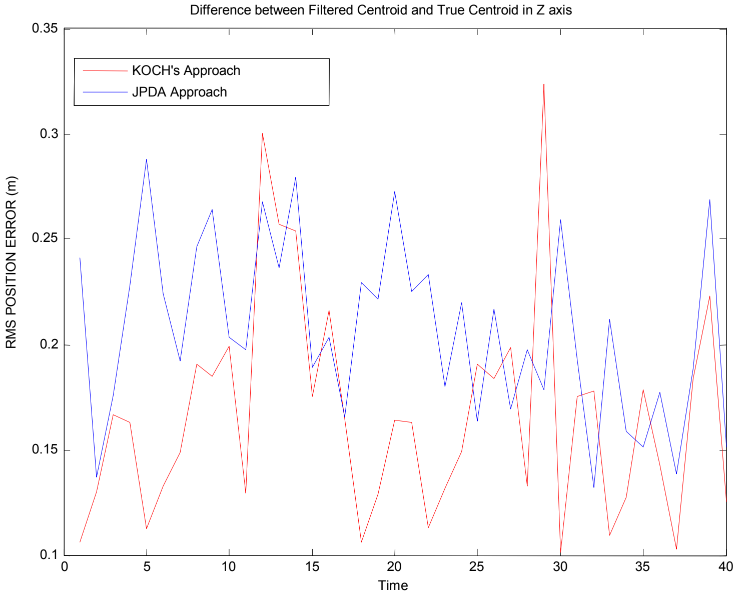
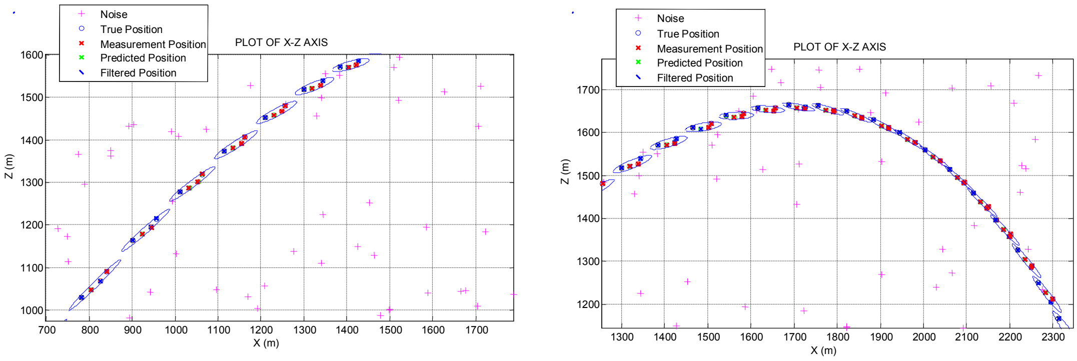


Figure 4.9. The Center Position Errors over time in the Z axis with $P_{DE} = 1$



First part of the Trajectory

Second part of the Trajectory

Figure 4.10. Ellipsoidal Shapes of the Clusters according to Koch approach with $P_{DE} = 0.9$

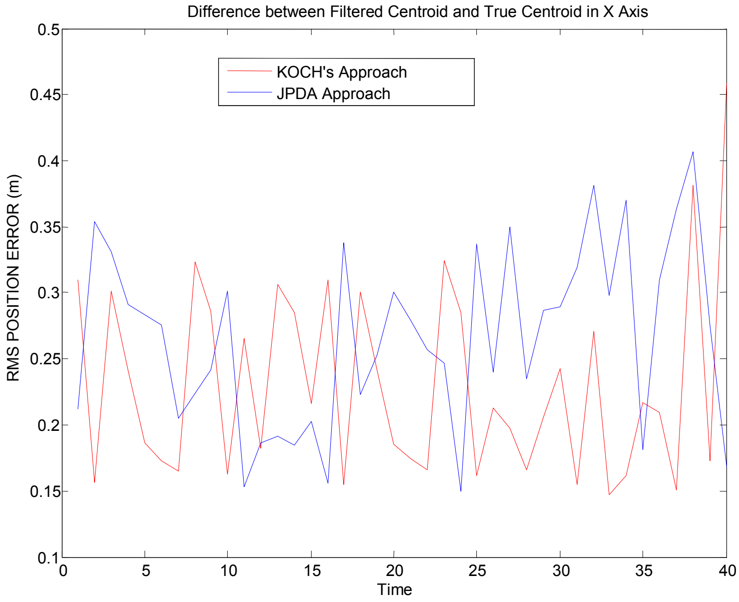


Figure 4.11. The Center Position Errors over time in the X axis with $P_{DE} = 0.9$

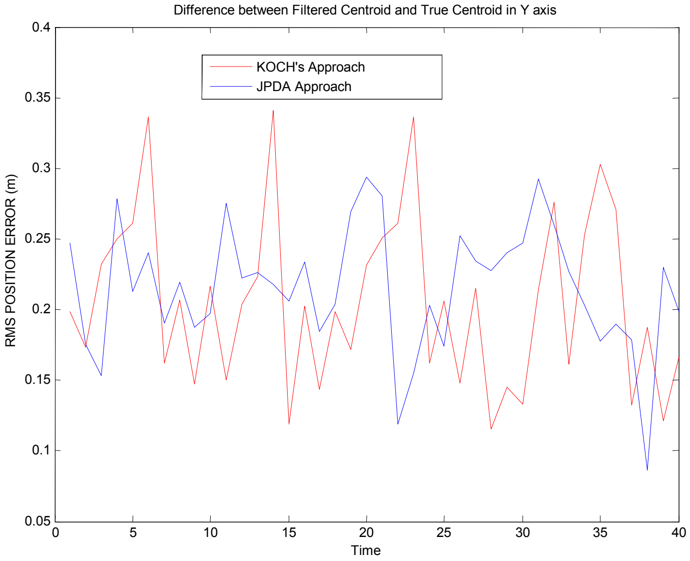


Figure 4.12. The Center Position Errors over time in the Y axis with $P_{DE} = 0.9$

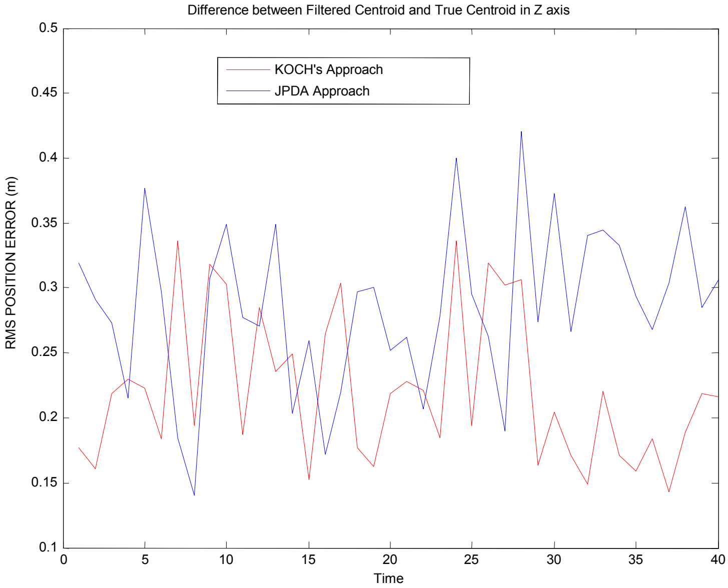
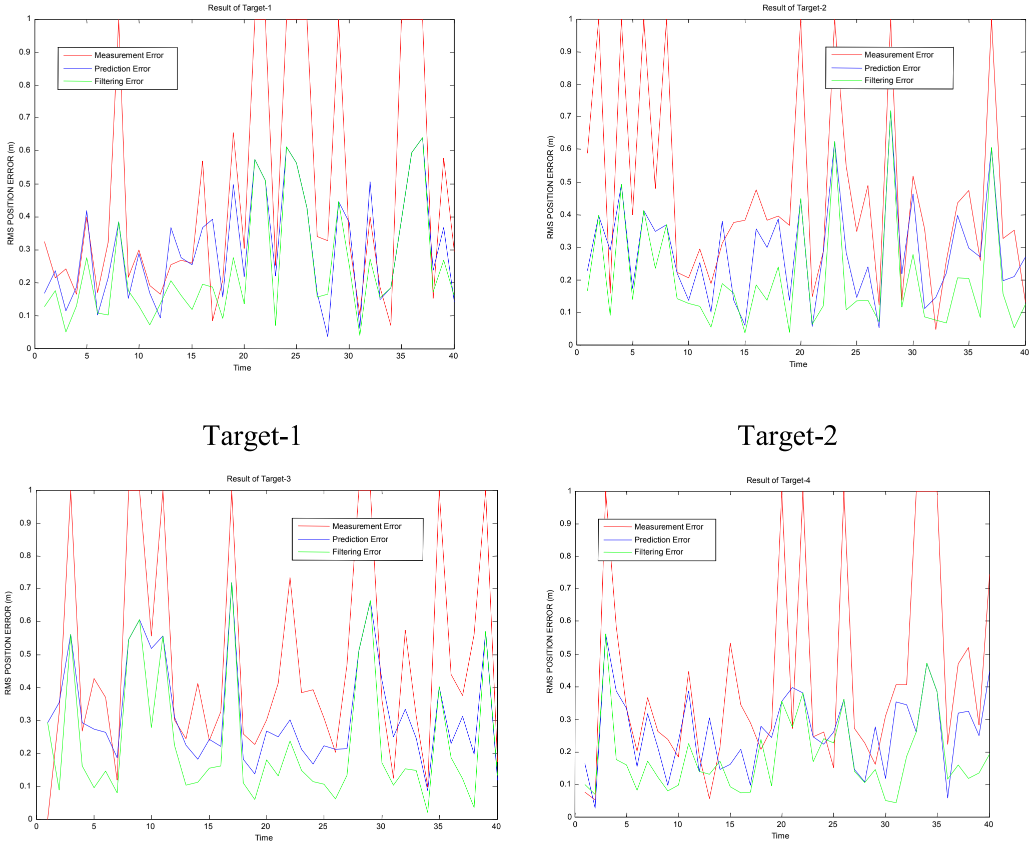


Figure 4.13. The Center Position Errors over time in the Z axis with $P_{DE} = 0.9$

These figures illustrate the performance comparison of the two methods. From the figures it can be concluded that both filters have comparable performance for $P_{DE} = 1$ and $P_{DE} = 0.9$. May be we can say that Koch's approach is slightly better compared with JPDAF. We didn't give the performance in detail since the scenario that we are working with is only illustrative and very restricted.

Below we also demonstrate the performance of the JPDA filter by comparing the measurement, prediction and filtering errors for each of the 4 targets. As can be seen from Figures 4.14 and 4.24 there is a clear performance increase between the measurements and predictions and also a clear performance increase when we go from prediction curves to filtering curves. We can say that the performance of the JPDA filter is satisfactory so the performance of the Koch approach.

Remaining experiments are done to see the effect of decreasing the probability of detection. The overall results related with the P_{DE} change are given in Table 4.1. In the following figures we demonstrate the performance for different detection probabilities as a function of time. Each figure is the result of 100 Monte Carlo runs. The performance of the JPDA is given only for $P_{DE} = 0.9$ and $P_{DE} = 0.5$ to observe the effect of the decrease in the detection probability in JPDA filter in the Figures 4.14 and 4.24. These two figures are the result of 100 Monte Carlo runs. Remaining figures show the effect of the P_{DE} decrease in the Koch approach. Only X-Z plane trajectories are shown to save space and the comparison of the two filters is done for the X and Z axis.



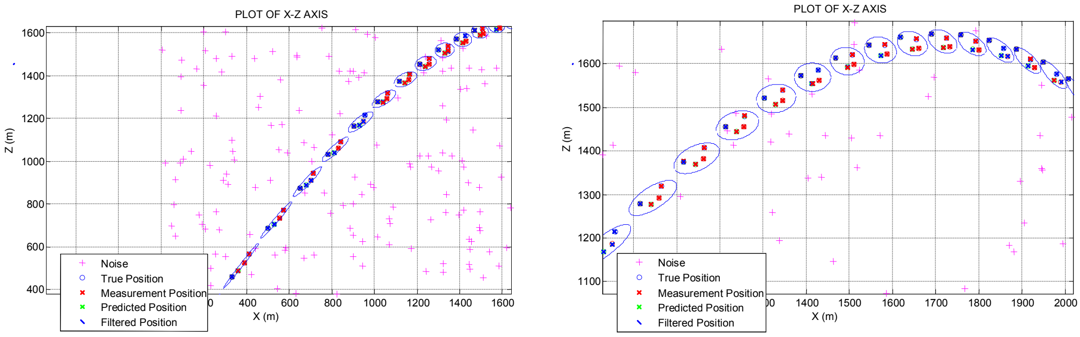
Target-1

Target-2

Target-3

Target-4

Figure 4.14. Position Errors of Targets according to JPDA with $P_{DE} = 0.9$



First part of the Trajectory

Second part of the Trajectory

Figure 4.15. Ellipsoidal Shapes of the Clusters according to Koch approach with $P_{DE} = 0.8$

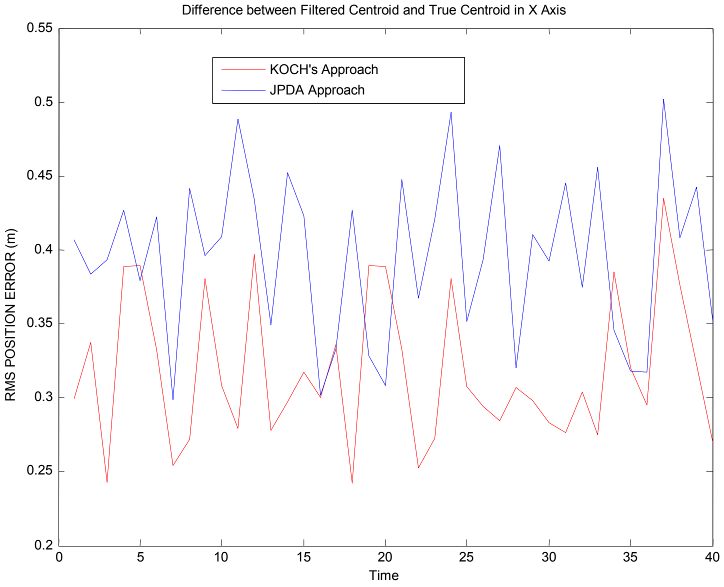


Figure 4.16. The Center Position Errors over time in the X axis with $P_{DE} = 0.8$

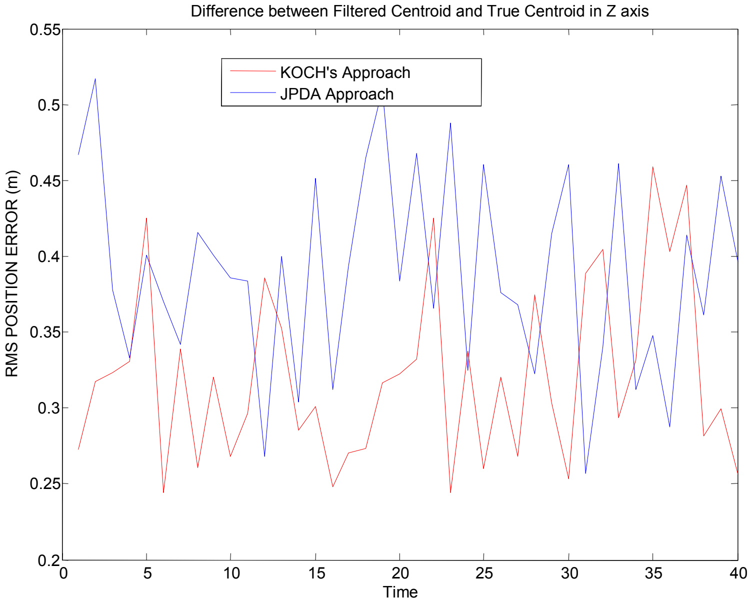
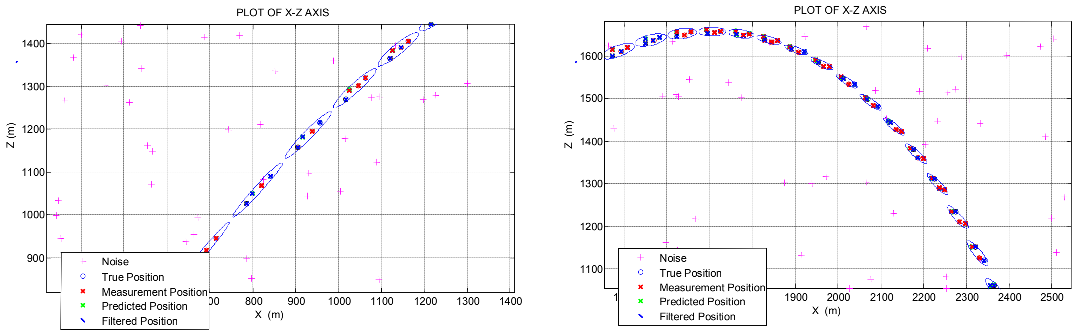


Figure 4.17. The Center Position Errors over time in the Z axis with $P_{DE} = 0.8$



First part of the Trajectory

Second part of the Trajectory

Figure 4.18. Ellipsoidal Shapes of the Clusters according to Koch approach with $P_{DE} = 0.7$

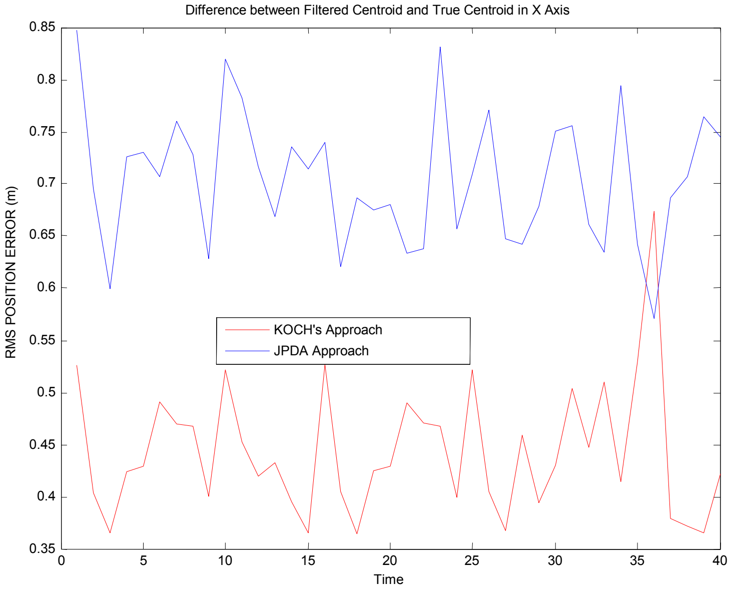


Figure 4.19. The Center Position Errors over time in the X axis with $P_{DE} = 0.7$

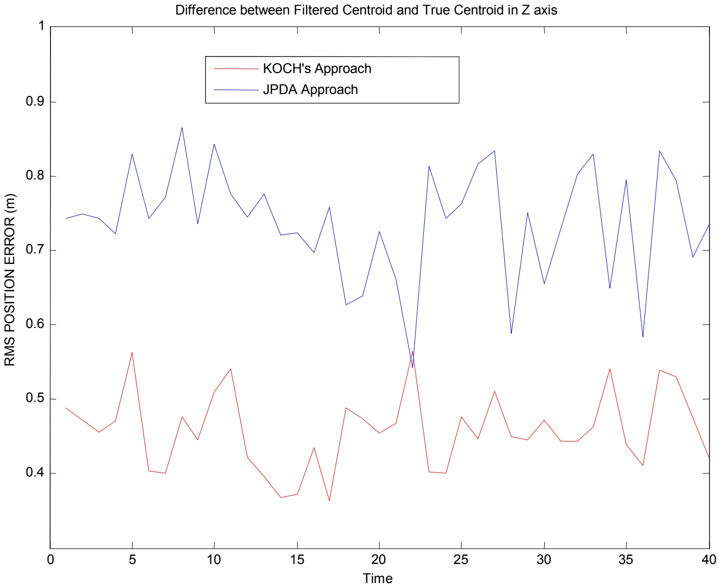
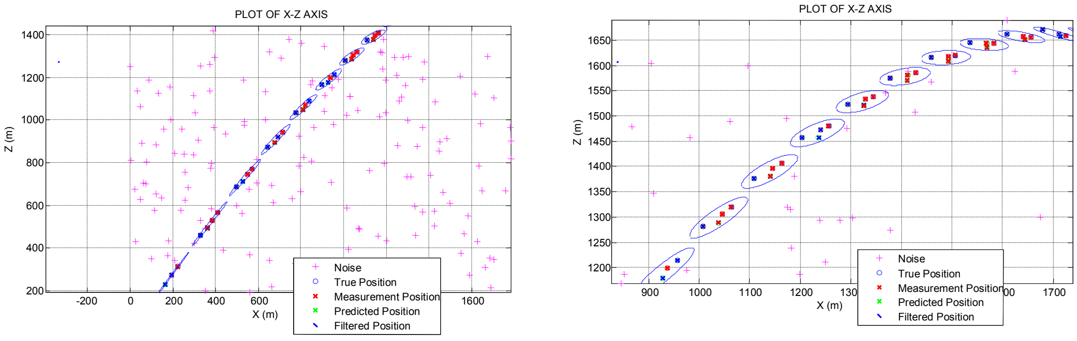


Figure 4.20. The Center Position Errors over time in the Z axis with $P_{DE} = 0.7$



First part of the Trajectory

Second part of the Trajectory

Figure 4.21. Ellipsoidal Shapes of the Clusters according to Koch approach with $P_{DE} = 0.6$

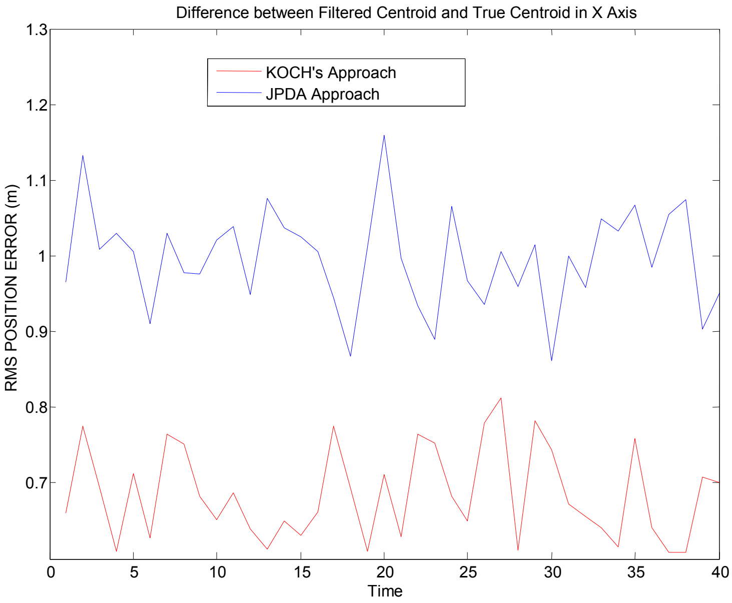


Figure 4.22. The Center Position Errors over time in the X axis with $P_{DE} = 0.6$

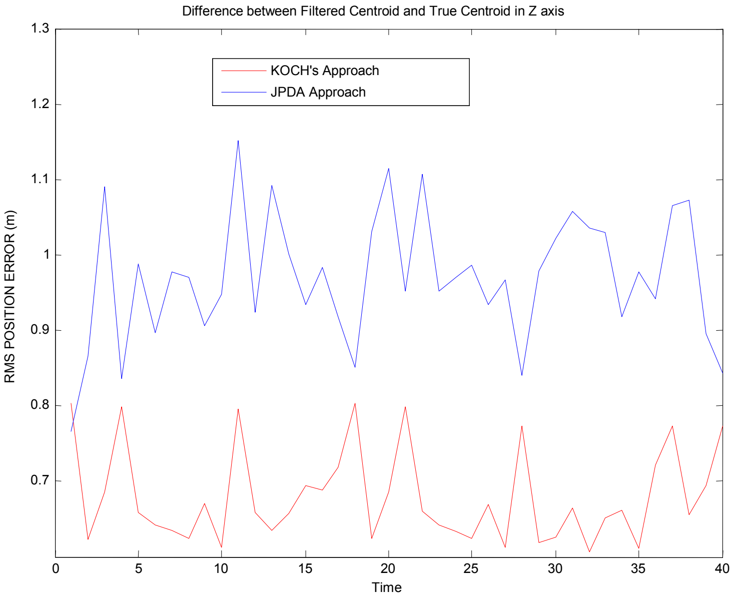
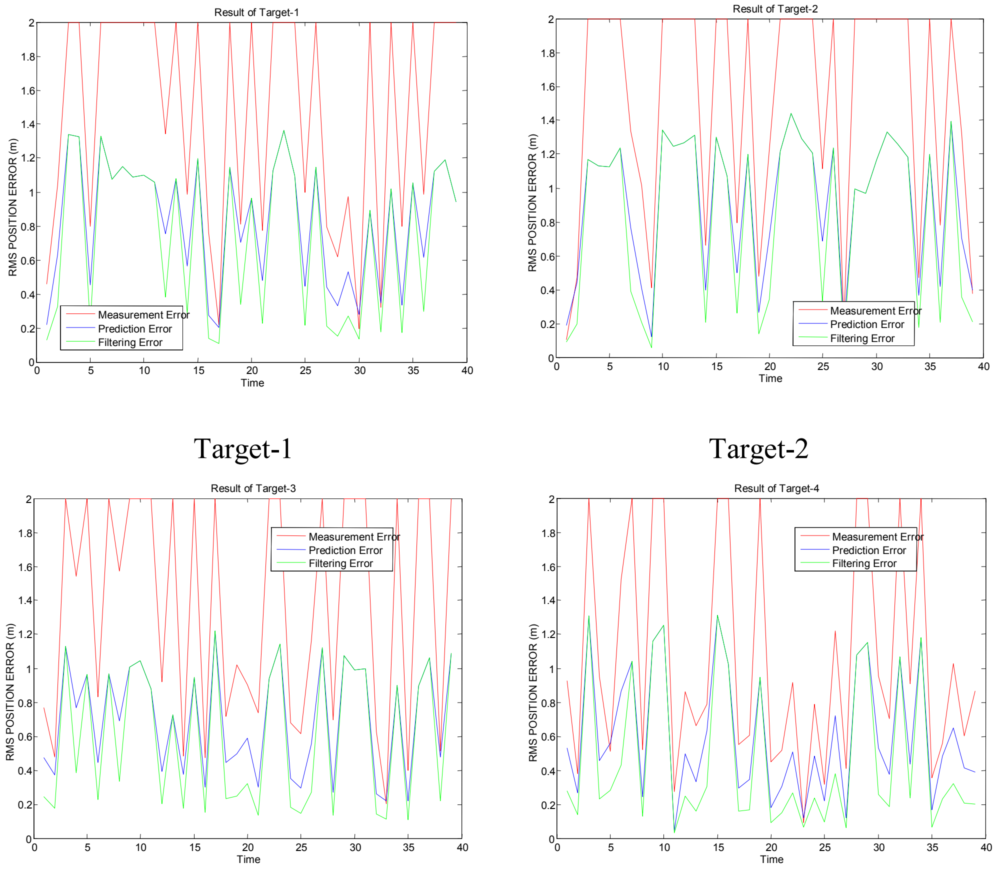


Figure 4.23. The Center Position Errors over time in the Z axis with $P_{DE} = 0.6$



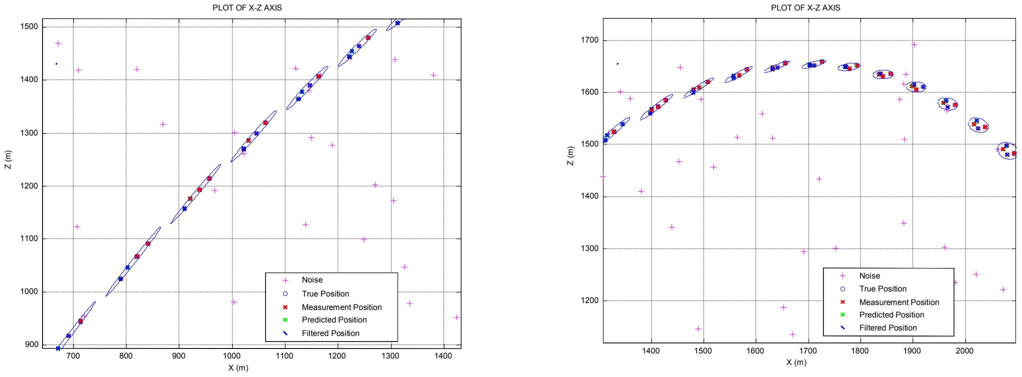
Target-1

Target-2

Target-3

Target-4

Figure 4.24. Position Errors of Targets according to JPDA with $P_{DE} = 0.5$



First part of the Trajectory

Second part of the Trajectory

Figure 4.25. Ellipsoidal Shapes of the Clusters according to Koch approach with $P_{DE} = 0.5$

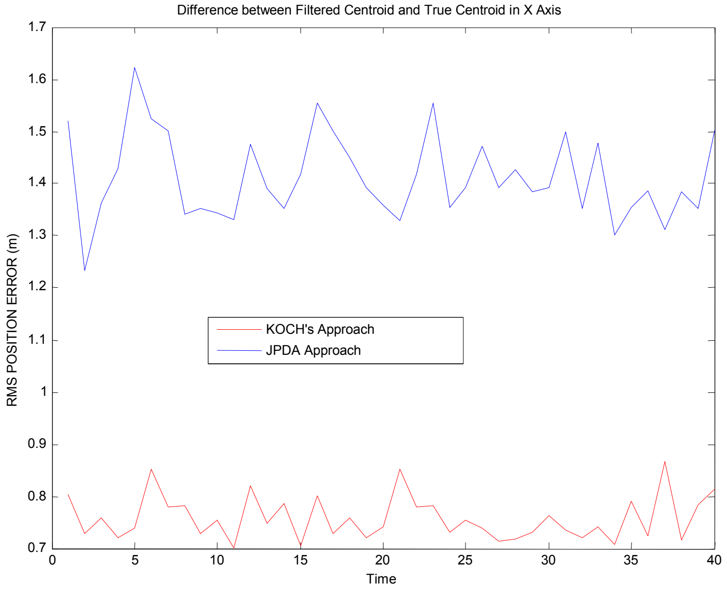


Figure 4.26. The Center Position Errors over time in the X axis with $P_{DE} = 0.5$

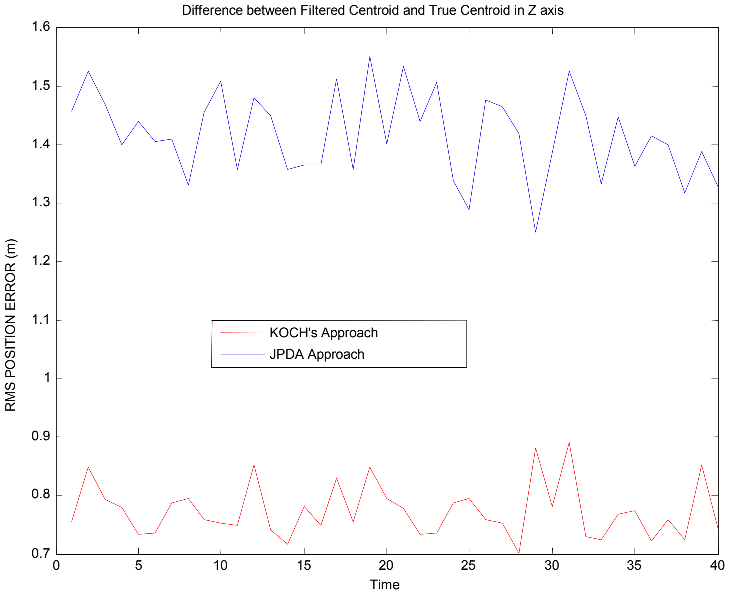
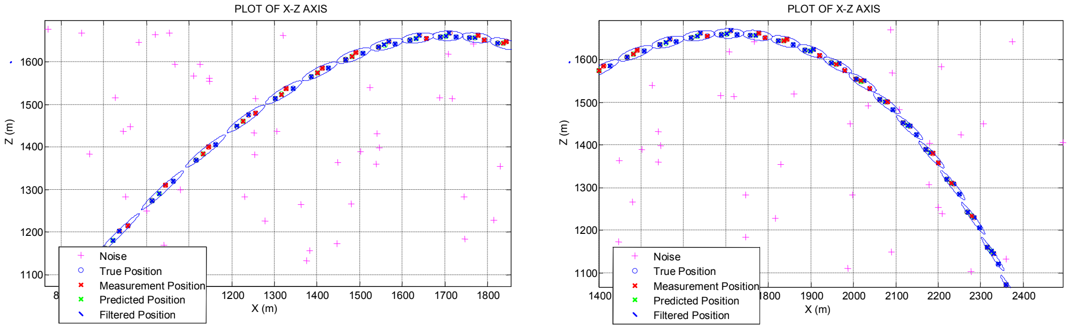


Figure 4.27. The Center Position Errors over time in the Z axis with $P_{DE} = 0.5$



First part of the Trajectory

Second part of the Trajectory

Figure 4.28. Ellipsoidal Shapes of the Clusters according to Koch approach with $P_{DE} = 0.4$

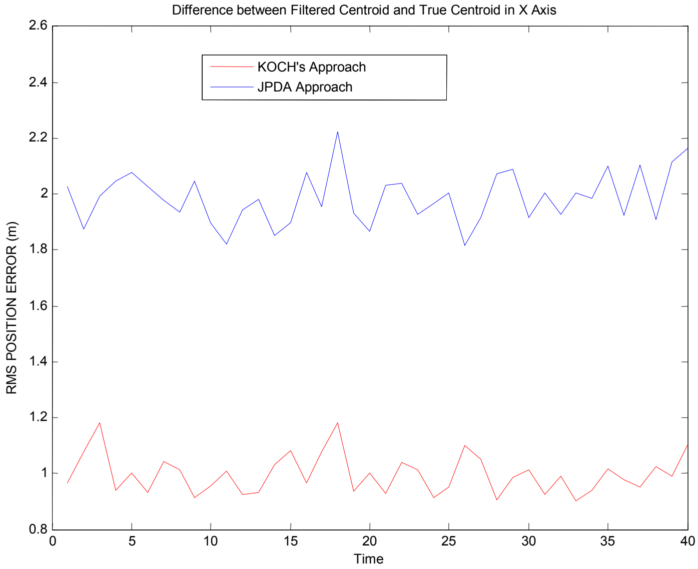


Figure 4.29. The Center Position Errors over time in the X axis with $P_{DE} = 0.4$

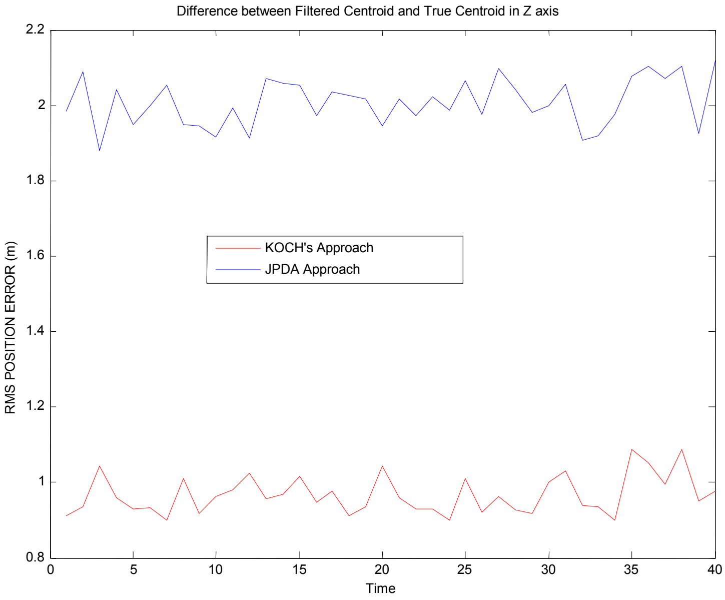
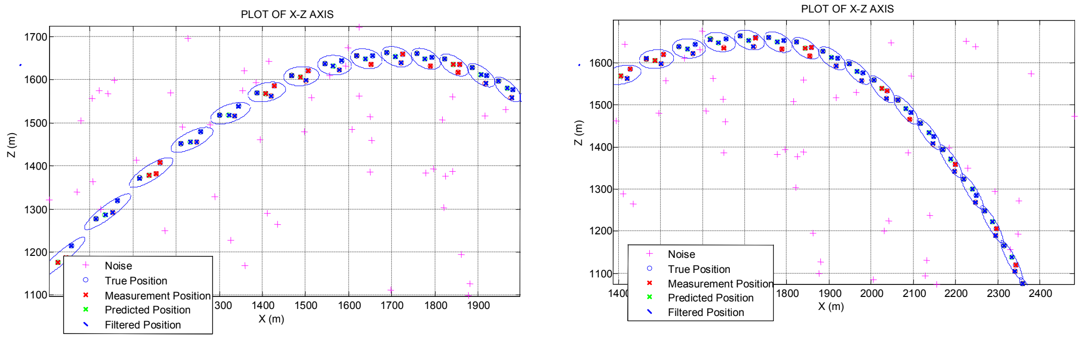


Figure 4.30. The Center Position Errors over time in the Z axis $P_{DE} = 0.4$



First part of the Trajectory

Second part of the Trajectory

Figure 4.31. Ellipsoidal Shapes of the Clusters according to Koch approach with $P_{DE} = 0.3$

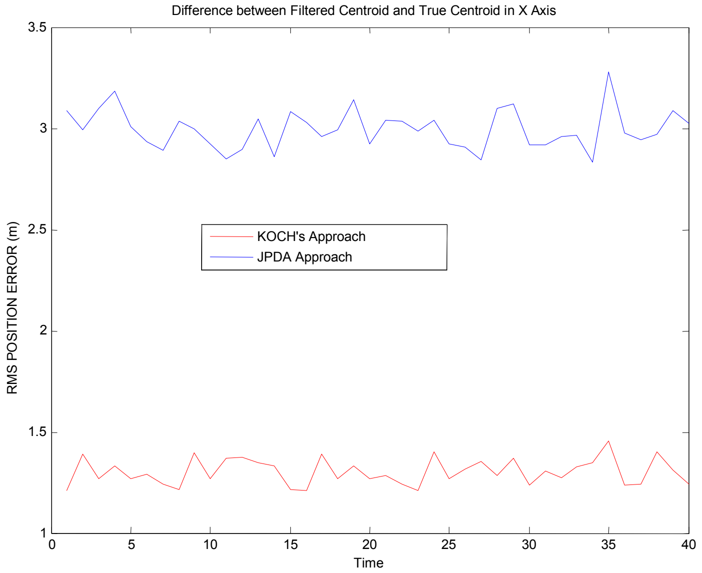


Figure 4.32. The Center Position Errors over time in the X axis with $P_{DE} = 0.3$

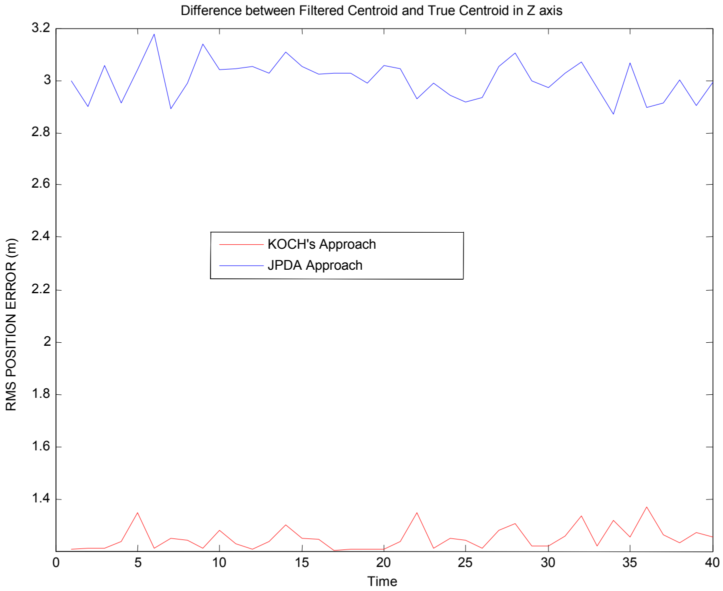
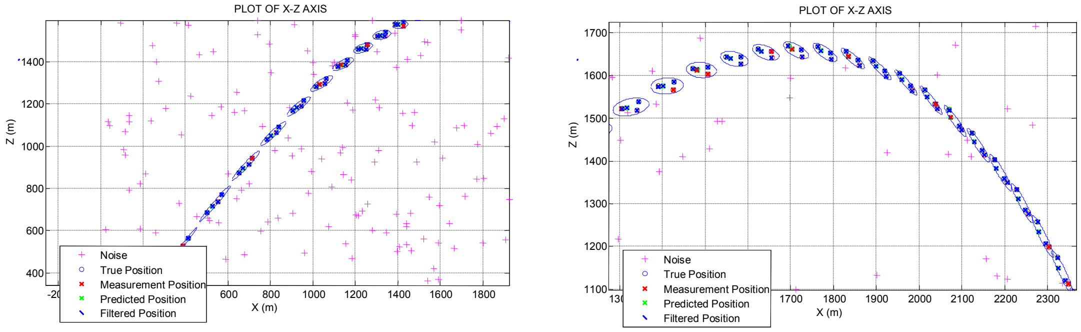


Figure 4.33. The Center Position Errors over time in the Z axis with $P_{DE} = 0.3$



First part of the Trajectory

Second part of the Trajectory

Figure 4.34. Ellipsoidal Shapes of the Clusters according to Koch approach with $P_{DE} = 0.2$

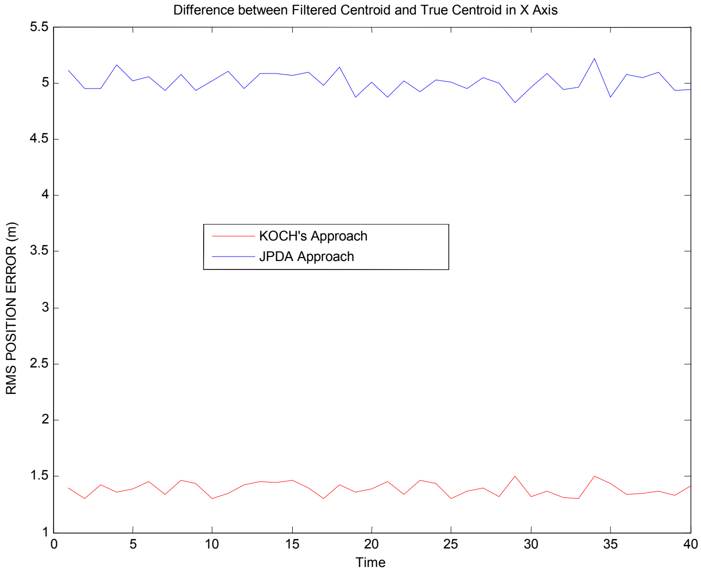


Figure 4.35. The Center Position Errors over time in the X axis with $P_{DE} = 0.2$

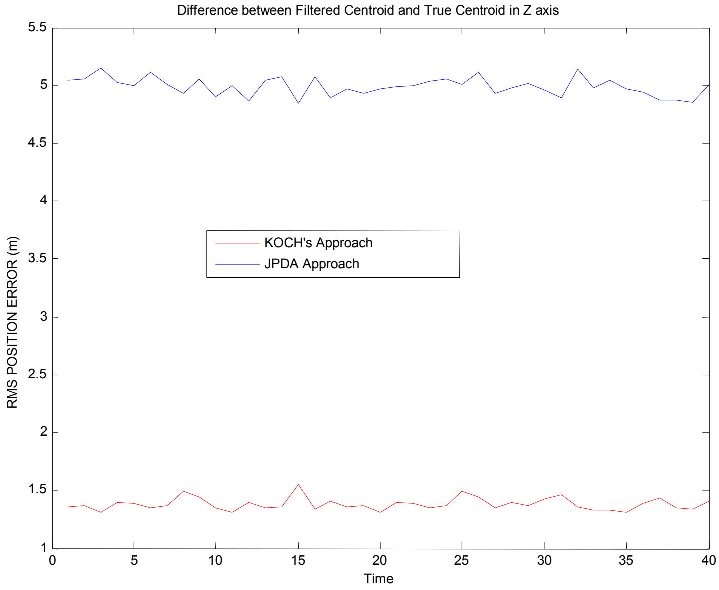
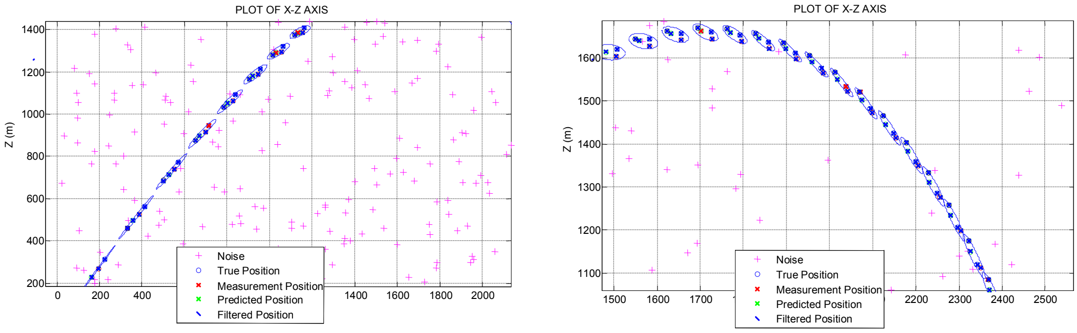


Figure 4.36. The Center Position Errors over time in the Z axis with $P_{DE} = 0.2$



First part of the Trajectory

Second part of the Trajectory

Figure 4.37. Ellipsoidal Shapes of the Clusters according to Koch approach with $P_{DE} = 0.1$

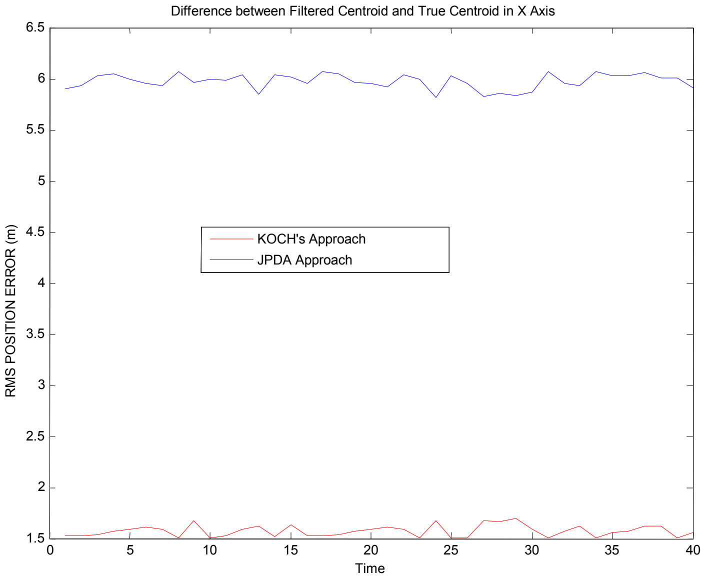


Figure 4.38. The Center Position Errors over time in the X axis with $P_{DE} = 0.1$

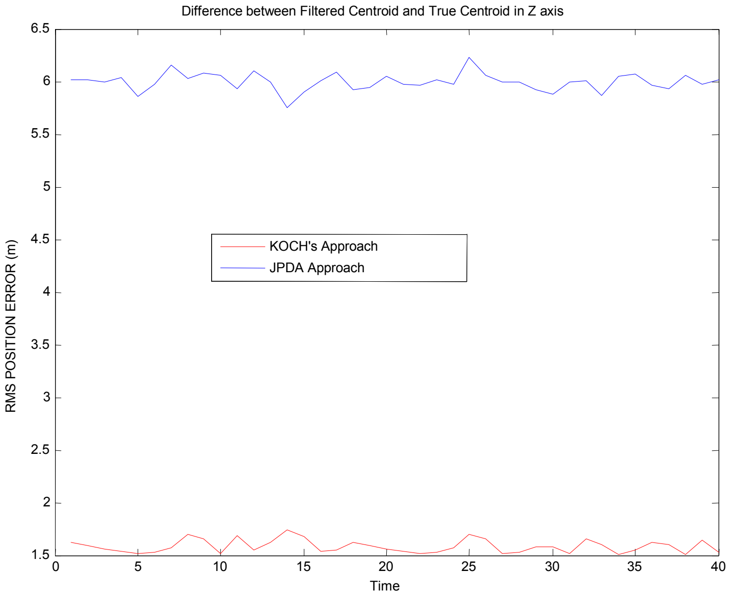


Figure 4.39. The Center Position Errors over time in the Z axis with $P_{DE} = 0.1$

Table 4.1. Average Center Position Errors evaluated after we did 100 Monte Carlo runs

P_{DE}	JPDA Approach Error (Meter)	Koch Approach Error (Meter)
$P_{DE} = 1.0$	0.2764	0.2496
$P_{DE} = 0.9$	0.3358	0.3031
$P_{DE} = 0.8$	0.4769	0.3571
$P_{DE} = 0.7$	0.7248	0.4628
$P_{DE} = 0.6$	1.1025	0.6287
$P_{DE} = 0.5$	1.3949	0.7792
$P_{DE} = 0.4$	2.0981	0.9864
$P_{DE} = 0.3$	3.0862	1.2904
$P_{DE} = 0.2$	5.0667	1.3568
$P_{DE} = 0.1$	6.0482	1.5824

A graphical representation of the table is given in Figure 4.40 below.

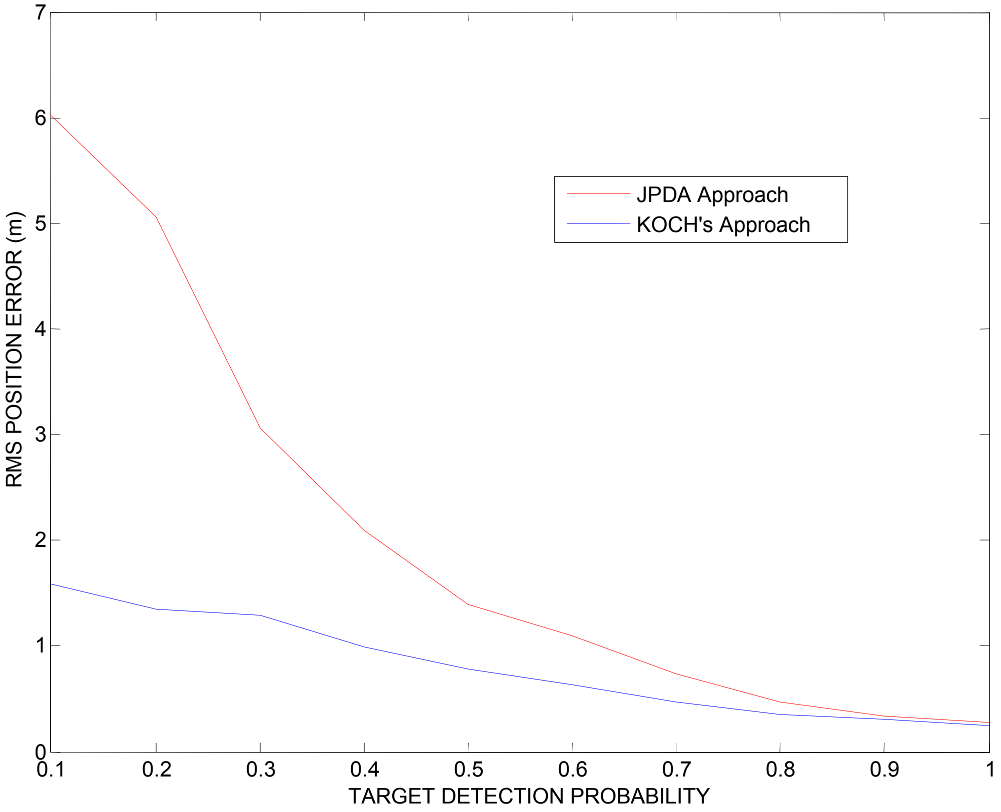


Figure 4.40. Performance Result of the Two Methods

From Figure 4.40, we conclude that the error performance of the Koch's approach is always better than the JPDA filter. The difference in the performances increases as the probability of detection decreases.

CHAPTER 5

SUMMARY AND CONCLUSIONS

Estimating the states of an object extension characterized by a collectively moving ballistic object group (cluster) is a very critical problem. In this thesis, we proposed to use two different methods for this problem. One of them is the Joint Probabilistic Data Association Filter (JPDAF) and the other is a new Bayesian approach to cluster tracking using random matrices. By using the motion model explained briefly in Chapter 1, we generated a set of data and used it in both approaches to evaluate the tracking performances of them.

In Chapter 2, the JPDAF algorithm has been illustrated by two different scenarios. By changing some of the system parameters, different runs are obtained for both of the scenarios. In Section 2.3.1, we try to show that the algorithm itself is working properly. In Section 2.3.2, the starting points of motions of the targets and their velocities are chosen as specific values to ensure at least one crossing time instant. For all runs, the measurement, prediction and filtering errors are drawn. It is shown that the errors generated by the algorithm are in an acceptable range, even at some specific time instants which we have ensured the crossing of the targets. The JPDA algorithm proved to give good results for clusters containing up to 4 targets. However, if the number of targets in the cluster exceeds this limit we meet the basic drawbacks of the JPDA algorithm, the total number of all feasible hypotheses increases exponentially. So JPDAF requires prohibitive amounts of processor time to compute all joint probabilities. Another drawback of this algorithm is that, if the tracks are close to each other for an extended time, then these closely spaced targets may be merged into a single track. And finally, once a false track is started, it will tend to persist, in other words, redundant track may be formed.

In Chapter 3, “the Bayesian approach to extended object and cluster tracking using random matrices” has been explained. Within this approach, ellipsoidal object extensions are modeled by random matrices and treated as additional state variables to be estimated. In the first scenario of Chapter 4, we illustrate this new approach by a simulation example. The aim of the example is to show that our algorithm is working properly and if we choose proper values for system parameters such as $\Delta t_k, \delta, \tau, \theta, \mu$ and make a right choice of initial conditions the algorithm will give satisfactory results and will track the center position and the ellipsoidal shape of the cluster without any problem. When the target detection probability is decreased, we have encountered some problems in the Koch's algorithm because of the equations containing the term $1/n_k$ where n_k is the number of measurements at time k (Equations 3.35 and 3.42). To deal with this problem we proposed to use the predicted center position (number of target $- n_k$) time(s). The use of the algorithm with this modification not only makes the method a feasible one but also gives much more accurate tracking results. Before the application of the modification even if the tracking is completed successfully the performance of the Koch's approach becomes always less than the performance of the JPDA. This change in the algorithm changed the comparison results completely as explained in Chapter 4 where we compared the tracking performance of the two approaches.

The Koch's approach as implemented here is modified to give more accurate results in [1], [5]. The improvements can be reflected to the work that we have presented in this thesis. Only a single-object tracking problem is handled with Koch's approach so far. In future work, the multi-object extension of this approach can be presented. This opens an interesting field for further research.

REFERENCES

[1] Koch, W., "Bayesian Approach to Extended Object and Cluster Tracking using Random Matrices." In: *IEEE Transactions Aerospace and Electronic Systems*, Vol. 44, No. 3, p. 1042-1059, 2008.

[2] Wieneke, M., Koch, W., "Probabilistic tracking of multiple extended targets using random matrices." In: Proc. SPIE, Vol. 7698, (2010); extended version under review *IEEE Transactions Aerospace and Electronic Systems.*

[3] Koch, W., Feldmann, M., "Cluster Tracking under Kinematical Constraints using Random Matrices."In: *Robotics and Autonomous Systems*, vol. 57, no. 3, pp. 296–309, 2009.

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

[5] Orguner, U., "A Variational Measurement Update for Extended Target Tracking With Random Matrices", *IEEE Transactions on Signal Processing*, Vol. 60, No. 7, July 2012, pp.3827-3834.

[6] Feldmann, M., Fränken, D., "Tracking of Extended Objects and Group Targets using Random Matrices – A New Approach," in Proc. Intern. Conf. on Information Fusion, 2008.

[7] Salmond, D., Gordon N., "Group and extended object tracking." *IEE Colloquium on Target Tracking*: Algorithms and Applications, 1999.

[8] Gupta, A. K., Nagar, D. K. "Matrix Variate Distributions". London: Chapman & Hall/CRC, 1999.

[9] Harville, D. A. "Matrix Algebra from a Statistician's Perspective". New York: Springer.

[10] Magnus, J. R., Neudecker, H., "Matrix Differential Calculus with Applications in Statistics and Econometrics". New York: Wiley, 1999.

[11] Ravindra, V.C., Bar-Shalom Y., Willet*,* P.,"Impact Point Prediction and Projectile Identification", Proc. of SPIE Vol. 6699, 66991A-1 through 66991A-10.

[12] Blackman, S.,"Multiple-Target Tracking with Radar Applications", Artech House, 1986.

[13] Bar-Shalom, Y., Fortmann, T.E., "Tracking and Data Association", Academic Press, 1988.

[14] Chang, K.C., Bar-Shalom, Y., “Joint Probabilistic Data Association for Multitarget Tracking with Possibly Unresolved Measurements and Maneuvers,” *IEEE Transactions on Automatic Control,* Vol. 29, No. 7, July 1984, pp. 585-594.

[15] Matlab codes written by Sabuncu, M., for the solution of the nonlinear ballistic equations.

[16] Bar-Shalom, Y., Tse, E., “Tracking in a Cluttered Environment With Probabilistic Data Association,” Automatica, Vol. 11, No. 5, Sept. 1975, pp. 451-460.

[17] Reid, D.B., “An Algorithm for Tracking Multiple Targets,” *Proceedings of the 17th IEEE Conference on Decision and Control*, 1978, pp. 1202-1211.

[18] McCoy, R.L., "Modern Exterior Ballistics", Schiffer Publishing Ltd, 1999.

[19] Fortmann, T.E., Bar-Shalom, Y., Scheffe, M., “Multi-Target Tracking Using Joint Probabilistic Data Association,” *Proceedings of the 19th IEEE Conference on Decision and Control*, 1980, pp. 807-812.

[20] Gilholm, K., Salmond D., "Spatial distribution model for tracking extented objects" *Proc. Inst. Electr. Eng.-Radar, Sonar, Navigation*, vol.152, no.5, pp.364-371, Oct. 2005.

[21] Mahler R.,"Statistical Multisource- Multitarget Information Fusion". Norwood, MA: Artech House, 2007.

[22] Angelova, D., Mihaylova, L., "Extended object tracking using Monte Carlo methods", *IEEE Transactions on Signal Processing*, Vol. 56, No. 2, pp. 825-832, Feb. 2008.

[23] Bar-Shalom, Y., Willet, P.K., Tian X., “Tracking and Data Fusion, A handbook of algorithms”, YBS Publishing 2011.

APPENDIX A

KRONECKER PRODUCT

Appendix A contains information about the Kronecker product and some matrix identities.

The Kronecker product $A \otimes B$ of two matrices $A = (a_{ij})_{i=1,j=1}^{m,n}$, B is any matrix is defined as

$$A \otimes B = \begin{pmatrix} a_{11}B & a_{12}B & \cdots & a_{1n}B \\ a_{21}B & a_{22}B & \cdots & a_{2n}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}B & a_{m2}B & \cdots & a_{mn}B \end{pmatrix} \quad \text{A-1}$$

For matrices A, B, C and a scalar α:

$$(A \otimes B) \otimes C = A \otimes (B \otimes C) \quad \text{A-2}$$

$$(A \otimes B)^T = A^T \otimes B^T \quad \text{A-3}$$

$$(A \otimes B)^{-1} = A^{-1} \otimes B^{-1} \quad \text{A-4}$$

For square matrices A, B we obtain: $tr[A \otimes B] = (trA)(trB)$. The determinant of $A \otimes B$ is given by the determinants of A, B with $m = \dim(A)$, $n = \dim(B) : |A \otimes B| = |A|^n |B|^m$.

For column vectors x, y of equal dimension, the following identities are valid:

$$x^T y = tr[xy^T] \quad \text{A-5}$$

$$x^T A^{-1} x = tr[xx^T A^{-1}] = tr[A^{-1} xx^T] \quad \text{A-6}$$

$$\exp\left[-\tfrac{1}{2} x^T A^{-1} x\right] = etr[-\tfrac{1}{2} xx^T A^{-1}] \quad \text{A-7}$$

$etr[A]$ is an abbreviation for $\exp[trA]$.

APPENDIX B

SOME PROBABILITY DENSITIES

Appendix B contains information about some densities such as Inverted Wishart and Wishart which are used in the computation of the group motion.

Gaussian Density:

Gaussian densities of a random vector x with expectation a and covariance matrix A can be written as [1]

$$N(x; a, A) = |2\pi A|^{-1/2} etr[-\tfrac{1}{2}(x-a)(x-a)^T A^{-1}] \qquad \text{B-1}$$

Note that $etr\left[-\frac{1}{2}(x-a)(x-a)^T A^{-1}\right] = e^{-\frac{1}{2}(x-a)^T A^{-1}(x-a)}$

Wishart Density:

In statistics, the Wishart distribution is any of a family of probability distributions defined over SPD matrix-valued random variables ("random matrices"). The distribution was first formulated by John Wishart in 1928 and it is of great importance in the estimation of covariance matrices.

Suppose E is a $n\,x\,p$ matrix. Each row of this matrix is independently drawn from a p-variate normal distribution with zero mean:

$$E_{(i)} = (e_i^1, \dots, e_i^p) \sim N_p(0, V) \qquad \text{B-2}$$

where V is a (fixed) $p\,x\,p$ positive definite symmetric (SPD).

If we define a scatter matrix S as:

$$S = E^T E \tag{B-3}$$

Then the probability distribution of the $p\ x\ p$ random matrix S has a Wishart distribution with n degrees of freedom where n is a positive integer. We generalize this as follows:

A $p\ x\ p$ positive definite symmetric (SPD) random matrix X is "Wishart-distributed," if its density is given by [1]

$$W(X; n, V) = \frac{1}{Z} |V|^n |X|^{(1/2)(n-p-1)} etr\left[-\tfrac{1}{2} V^{-1} X\right], \qquad n \geq p \tag{B-4}$$

where n is a scalar parameter (degrees of freedom), V is a $p\ x\ p$ SPD matrix , and Z is a normalizing constant.

Its first and second moments are given by

$$E[X] = nV \tag{B-5}$$

$$C[x_{ij}, x_{kl}] = n(n_{ik} n_{jl} + n_{il} n_{jk}) \tag{B-6}$$

Inverted Wishart Density:

A $d\ x\ d$ SPD random matrix X is "Inverted-Wishart-distributed" if its density is given by[1]

$$IW(X; a, A) = \frac{1}{Z} |A|^{(1/2)(a-d-1)} |X|^{-(1/2)a} etr\left[-\tfrac{1}{2} X^{-1} A\right], \qquad a > 2d \tag{B-7}$$

where a is a scalar parameter, A is a $d\ x\ d$ SPD matrix and Z is a normalizing constant.

Its first and second moments are given by

$$E[X] = \frac{A}{a - 2d - 2}, \qquad a - 2d - 2 > 0 \tag{B-8}$$

$$C[x_{ij}, x_{kl}] = \frac{2(a-2d-2)^{-1} a_{ij} a_{kl} + a_{ik} a_{jl} + a_{il} a_{kj}}{(a-2d-1)(a-2d-2)(a-2d-4)} \qquad a - 2d - 4 > 0 \tag{B-9}$$

The following fact is important:

Fact: If X is an inverted Wishart distributed random matrix with the density $p(X) \propto |X|^{-(1/2)a} etr\left[-\frac{1}{2}X^{-1}A\right]$, then the inverse matrix $Y = X^{-1}$ is Wishart-distributed with $p(Y) \propto |Y|^{-(1/2)(a-d-1)} etr\left[-\frac{1}{2}YA\right]$.

Beta Density:

A $d\, x\, d$ SPD random matrix X is "generalized-beta-type-II-distributed," if its density is given by [1]

$$GB_d^{II}(X; a, b, A, B = O) := B(X; a, b, A) := \frac{1}{Z}|A|^b |X|^{a-(d+1)/2} |A + X|^{-(a+b)} \qquad \text{B-10}$$

Where a and b are scalar parameters, A is a $d\, x\, d$ SPD matrix, and Z is a normalizing constant.

Its first and second moments are given by

$$E[X] = \frac{2a}{2b - d - 1}A, \qquad 2b - d - 1 > 0 \qquad \text{B-11}$$

$$C[x_{ij}, x_{kl}] = \frac{2a(2(a + b) - d - 1)}{(2b - d)(2b - d - 1)(2b - d - 3)} * \left(\frac{2}{(2b - d - 2)} c_{ij}c_{kl} + c_{jl}c_{ik} + c_{il}c_{kj}\right) \qquad \text{B-12}$$

with $2b - d - 3 > 0$.