INTERACTING MULTIPLE MODEL PROBABILISTIC DATA ASSOCIATION FILTER USING RANDOM MATRICES FOR EXTENDED TARGET TRACKING

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

BY

EZGİ ÖZPAK

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

JANUARY 2018

Approval of the thesis:

INTERACTING MULTIPLE MODEL PROBABILISTIC DATA ASSOCIATION FILTER USING RANDOM MATRICES FOR EXTENDED TARGET TRACKING

submitted by **EZGİ ÖZPAK** 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. Gülbin Dural Ünver ____________________
Director, Graduate School of **Natural and Applied Sciences**

Prof. Dr.Tolga Çiloğlu ____________________
Head of Department, **Electrical and Electronics Eng.**

Assoc. Prof. Dr. Umut Orguner ____________________
Supervisor, **Electrical and Electronics Eng. Dept., METU**

Examining Committee Members:

Prof. Dr. Ali Özgür Yılmaz ____________________
Electrical and Electronics Engineering Eng. Dept., METU

Assoc. Prof. Dr. Umut Orguner ____________________
Electrical and Electronics Engineering Eng. Dept., METU

Prof. Dr. Çağatay Candan ____________________
Electrical and Electronics Engineering Eng. Dept., METU

Assist. Prof. Dr. Emre Özkan ____________________
Electrical and Electronics Engineering Eng.Dept., METU

Assoc. Prof. Dr. Özgül Salor-Durna ____________________
Electrical and Electronics Engineering, Gazi University

Date: 30.01.2018

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: Ezgi ÖZPAK

Signature :

ABSTRACT

INTERACTING MULTIPLE MODEL PROBABILISTIC DATA ASSOCIATION FILTER USING RANDOM MATRICES FOR EXTENDED TARGET TRACKING

Özpak, Ezgi

M.S., Department of Electrical and Electronics Engineering

Supervisor: Assoc. Prof. Dr. Umut Orguner

January 2018, 86 pages

In this thesis, an Interacting Multiple Model – Probabilistic Data Association (IMM-PDA) filter for tracking extended targets using random matrices is proposed. Unlike the extended target trackers in the literature which use multiple alternative partitionings/clusterings of the set of measurements, the algorithm proposed here considers a single partitioning/clustering of the measurement data which makes it suitable for applications with low computational resources.

When the IMM-PDA filter uses clustered measurements, a predictive likelihood function for the extent measurements is necessary for hypothesis probability calculation. Alternative predictive likelihood functions proposed in the literature for this purpose are surveyed and their shortcomings are identified in the thesis. Then an alternative predictive likelihood function is proposed and its advantage is illustrated on simulations running IMM-PDA filters with different predictive likelihood functions on a scenario involving a fighter aircraft launching a missile.

When a single partitioning/clustering is used before the tracking operation as is the case for the tracker proposed in the thesis, the clusters corresponding to close targets might be merged by the pre-clustering step, which might lead to track loss in the tracker. For overcoming this problem, a specific algorithm is proposed for handling close targets and its performance is illustrated on a simple scenario.

Keywords: Extended Target Tracking, Bayesian Approach, Object Extension, Data Association, Random Matrices, Clustering

ÖZ

BÜYÜK HEDEF TAKİBİ İÇİN RASLANTISAL MATRİSLER KULLANAN ETKİLEŞİMLİ ÇOKLU MODEL OLASILIKSAL VERİ EŞLEME FİLTRESİ

Özpak, Ezgi
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi :Doçent. Dr. Umut Orguner

Ocak 2018, 86 sayfa

Bu tezde, rastgele matrisler ile ifade edilen büyük hedefleri izlemek için Etkileşimli Çoklu Model - Olasılıksal Veri Eşleme (IMM-PDA) filtresi önerilmiştir. Literatürdeki çok sayıdaki bölümlenmiş / kümelenmiş ölçüm gruplarını kullanan büyük hedef izleyicilerin aksine burada önerilen algoritma, düşük hesaplama kaynakları olan uygulamalar için ölçüm verisini uygun hale getiren tek bir bölümleme / kümeleme işlemini ele alır.

IMM-PDA filtresi kümelenmiş ölçümleri kullanırken, hipotez olasılığının hesaplanabilmesi için yayılan ölçümlerin olabilirlik fonksiyonuna ihtiyaç duyar. Bu amaçla bu tezde, literatürde önerilen alternatif olabilirlik fonksiyonları araştırılıp ve bu fonksiyonların eksiklikleri tespit edilmiştir. Daha sonra, alternatif bir olabilirlik fonksiyonu önerilmiş ve bu fonksiyonun avantajı, füze başlatan bir savaş uçağını içeren senaryoda, farklı olabilirlik fonksiyonlarını kullanan IMM-

PDA filtrelerinin bulunduğu simülasyonlar üzerinde gösterilmiştir.

Tezde önerilen izleyici durumunda olduğu gibi, izleme işlemi öncesinde tek bir bölümleme / kümeleme kullanıldığında, yakın hedeflere karşılık gelen kümeler, ön kümeleme adımıyla birleştirilip, izleyicide izin kaybedilmesine yol açabilir. Bu sorunun üstesinden gelmek için, yakın hedeflerin izlenmesini sağlayan özel bir algoritma önerilmiş ve bu algoritmanın performansı basit bir senaryo üzerinde gösterilmiştir.

Anahtar Kelimeler: Büyük Hedef Takibi, Bayesian Yaklaşımı, Hedef Boyutu, Ölçüm İlişkilendirme, Rastgele Matrisler, Kümeleme

To My Family

ACKNOWLEDGEMENTS

I would like to extend my deepest gratitude to Assoc. Prof. Dr. Umut ORGUNER for his help, guidance, advice, criticism and encouragements throughout the study.

I owe special and deepest gratitude to my husband Yiğit ÖZPAK and my family for their friendship, love, support, patience and encouragement throughout the study.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURES

LIST OF ALGORITHMS

ALGORITHMS

LIST OF TABLES

TABLES

LIST OF SYMBOLS

$\bar{c}_j$	Normalization constants for mixing probabilities
$\tilde{X}_{k\|k-1}$	Predicted extension
$\bar{z}_k$	Mean of measurements
$\bar{Z}_k$	Measurement spread
$\hat{z}_{k\|k-1}$	Predicted measurement
B_k	Process noise gain
H_k	Measurement matrix
K_k	Kalman gain
n_k	Number of measurements in a cluster
p_{ij}	Transition probability
$P^i_{k\|k}$	Kinematic state covariance matrix for the ith IMM model
$P_{k\|k}$	Estimated state covariance matrix
$P_{k\|k-1}$	Predicted State Covariance Matrix
$S_{k\|k-1}$	Predicted measurement covariance matrix
v_k	Process noise term in state vector
w_k	Measurement noise
x_k	Kinematic state of the target
$x^i_{k\|k}$	Kinematic state estimate for the ith IMM model
$x_{k\|k}$	Estimated kinematic state

$X_{k\|k}$	Estimated extension
$x_{k\|k-1}$	Predicted State
z_k	Measurement vector
Z_k	Measurement set
μ_k^j	Mode probability
β_k^j	Association probability
F	State transition matrix
T	Sampling time
τ	Time constant

Abbreviations

ET- GM-PHD	Extended Target Gaussian Mixture Probability Hypothesis Density
ETT	Extended Target Tracking
GGIW	Gamma Gaussian Inverse Wishart
GJPDA	Generalized Jointly Probability Data Association
GM-PHD	Gaussian Mixture Probability Hypothesis Density
GPDA	Generalized Probabilistic Data Association
IMM	Interacting Multiple Model
JPDA	Joint Probabilistic Data Association
MC	Monte Carlo
MMSE	Minimum Mean Square Error

PDA	Probabilistic Data Association
PHD	Probability Hypothesis Density
PMHT	Probabilistic Multi-hypothesis Tracking
PMHT-ET	PMHT for extended objects
RHM	Random Hypersurface Model
SPD	Symmetric Positive Definite

CHAPTER 1

INTRODUCTION

Radar systems send electromagnetic signals to the environment and receive their echo from the targets. After processing the echo, measurements that may belong to the target, clutter or false alarms are obtained. Target tracking algorithm estimates the state of the target by using these measurements. Herein, there is generally the assumption that the target is a point source. In other words, a target results in at most a single measurement. One reason of this assumption is that the resolution of the radar system may not be sufficient to detect multiple observations over a big target. However, new radar systems, such as sea surveillance radar systems, have high resolution, which leads to multiple measurements from a single target in many sensor reports. In the literature, the targets which generate multiple measurements in the sensor reports are called as extended targets.

In a standard target tracking algorithm, the target's state is estimated by using the measurements which are in a predefined region around the predicted measurement of the target. This region is called the gate for the target. In the case there are many measurements in the gate, data association is necessary to select the correct measurement belonging to the target. Some well-known data association methods are Probabilistic Data Association (PDA), Joint Probabilistic Data Association (JPDA) and Multiple Hypothesis Tracking (MHT). All of these data association methods assume that the target is a point target, i.e., it is not an extended target. This thesis proposes an efficient PDA based extended target tracking algorithm.

1.1 Organization of the Thesis

In the thesis, first the previous studies about extended target tracking are reviewed in Chapter 2. In Chapter 3, the problem of target tracking is explained under the assumption of a point target. Details about the main steps of the tracking algorithm such as gating, Kalman filter, IMM and PDA can be found in this chapter. Since the extended target tracking algorithm used in the thesis uses similar versions of these methods, this chapter gives background information about the studies in the thesis in addition to a general target tracking algorithm. In Chapter 4, existing methods used for extended target tracking in the literature are reviewed. Chapter 5 gives the proposed extended target tracking algorithm in this thesis. In Chapter 6, simulation results are presented. Finally, in last Chapter 7, conclusions are drawn according to the results in Chapter 6 and possible ideas for future work are given.

CHAPTER 2

LITERATURE SURVEY

Objective of this chapter is to present a review of the studies on extended target tracking. In radar systems, point target assumption is not valid if the resolution of the radar is higher than the spatial extension of the extended target. An extended target is defined as a target resulting in more than one measurements in a scan [1]. In order to solve the extended target tracking problem, several methods have been proposed in the literature [2]. These methods are surveyed in the following sections.

2.1 Single Extended Target Tracking Algorithms

In a radar system, measurements scattered from the target are distributed over the target extent and this information can be used to extract the target shape. This idea is given by [3] saying that the spread of measurements fits a Gaussian probability density and [4] uses Kalman filter (to estimate kinematics) and a generalized version on Jointly Probabilistic Data Association (GJPDA) algorithm under this assumption. The difference of GJPDA than JPDA is to use all measurements scattered from the target instead of a single measurement. This method identifies data association hypotheses and updates the state vector for each hypothesis.

[5] aims to estimate the target shape by modelling the measurements named by Hypersurface Model (RHM). According to RHM, measurement can lie in the interior of the object boundary as it can be on the boundary. A shape equation is defined to model a curve that represents the shape. This equation is used in a

Gaussian estimator such as Unscented Kalman Filter (UKF).

[6] aims to track an extended target or a collectively a moving target group by using extension estimation. This method assumes that measurements are randomly distributed over the target extent. A Bayesian approach for extended target tracking is proposed while the object extension is represented by a Symmetric Positive Definite (SPD) random matrix under the assumption that the target shape is elliptical. Object extension is added to the state vector and is also updated recursively as target kinematics.

[7] argues that the object extension observed by the sensor depends on the sensor error as well as the target shape whereas [6] neglects the sensor error. With this observation, [7] improves the update equations of [6]. For the possibility of maneuvering target existence, Interacting Multiple Model (IMM) filter is modified by adding the object extension to the state.

There is no analytic solution for an exact Bayesian update for the random matrix model of [7]. Due to this, [8] derives an approximate measurement update using a variational Bayes approximation. [8] also suggests an approximate likelihood function.

[9] derives a prediction update using analytical approximations to improve on the heuristic prediction updates in [6] and [7]. To improve maneuvering target tracking performance, it also investigates the cases where the kinematic state depends on the object extension.

[10] performs a study based on [6] and [7]. In [10], the extension is modeled by a SPD random matrix as in [6] and [7]. [10] does not neglect the measurement noise whereas [6] includes its effect. The main difference between [10] and [7] is the usage of two random matrix based models. [10] assumes that the object extension can change over time and depends on orientation, size and shape. So [10] uses different models to describe the dependence of extension on time and orientation, shape, size. In order to deal with the maneuvering target problem, a multiple model (MM) approach is also used.

[18] models the extended targets as ellipsoidal and measures the down range and cross range extent of the ellipse. The state vector is augmented by down range and cross range extent. Here, a Rao-Blackwellised UKF is used to deal with the nonlinearity of the model. In addition, IMM filter is used to track the maneuvering target.

[19] assumes that the extended target has a time varying number of measurements. In addition to the kinematic parameters, the shape of the target is also tracked. The shape of the target is modelled by Random Hypersurface Model for a so-called star convex extended object.

[20] assumes that the shape of the extended target is rectangular and uses the so-called Box Particle Filter to track an extended target.

2.2 Multiple Extended Target Tracking Algorithms

The study [11] achieves multiple extended target tracking using the Gaussian Mixture Probability Hypothesis Density filter (GM-PHD) without estimating the target extent. The resulting filter is called extended target GM-PHD (ET-GM-PHD) filter.

[12] derives a new kind of Probabilistic Multi-Hypothesis Tracker (PMHT) which estimates the ellipsoidal shape and the kinematics of each target simultaneously and uses the method called PMHT for extended objects (PMHT-E) to track multiple extended targets. The results show that this method is unfortunately not sufficient for merging or splitting targets.

[13] aims to estimate the object extension modelled by PSD random matrix under assumption of elliptical target shape derived as in [6]. In [13], it is assumed that PHD is approximated with a Gaussian inverse Wishart mixture distribution and a likelihood function is derived to track multiple targets.

[14] proposes a Gamma Gaussian Inverse Wishart PHD filter for tracking multiple extended targets. Here, the PHD is approximated using a Gamma

Gaussian Inverse Wishart mixture and measurement rates (number of measurements obtained from a target at each scan) are estimated in addition to the target kinematic and extent variables.

[15] proposes Generalized (Jointly) Probabilistic Data Association (G(J)PDA) algorithm and uses object extension which is represented by PSD random matrix as mentioned in [7].

[16] assumes that the target has elliptical shape and computes down-range extent of ellipse and then augments the state vector by adding down-range extent to the kinematic states. This extra information makes enables the tracker maintain tracks for close targets. Here, JPDA and IMM filter with EKF are used to track the targets. Also, it is shown that particle filter also works.

[17] models the extended target in terms of the translation and the rotation of the target. Here target is modeled as a set of points. [17] augments the state vector by adding the positions of the points in the fixed reference frame to the state. A particle filter is used to track the target. In order to reduce the computation of the particle filter, a new particle weight calculation method is proposed.

[21] uses a cardinalized probability hypothesis density (CPHD) filter to track the extended targets. This method is named as Extended Target Tracking CPHD (ETT-CPHD). The PHD for the targets is represented as a gamma Gaussian inverse Wishart (GGIW) mixture.

[22] uses a Gamma Gaussian inverse-Wishart Poisson multi-Bernoulli mixture (GGIW-PMBM) filter to track the extended targets.

CHAPTER 3

TARGET TRACKING

A target tracking algorithm estimates the current state of a target by processing the measurements originated from the target. The state of the target contains kinematic components such as position, velocity, acceleration and the feature components like target type and target extension etc.

The measurements obtained from the sensor are usually not from only the targets. Because of the environmental reasons, the measurements include information about clutter or unwanted objects such as mountains, roads, buildings or other targets. It is also evident that measurements are noise corrupted.

In order to obtain the correct state of the target, trackers use filtering and data association methods. Before filtering, gating operation selects measurements close to the prediction of the target. The filtering methods estimate the current state of a target from the gated measurements. Kalman filter is a way of filtering and is usually preferred in the target tracking algorithms. Kalman filter estimates the kinematic information from the sets of measurements that have trajectory information of the target. In addition to the Kalman filter, IMM filter, which is a state estimation algorithm used for maneuvering targets, can be used. Data association methods update the target state according to generated association hypotheses.

A general flowchart of a target tracking algorithm is shown in Figure 1.

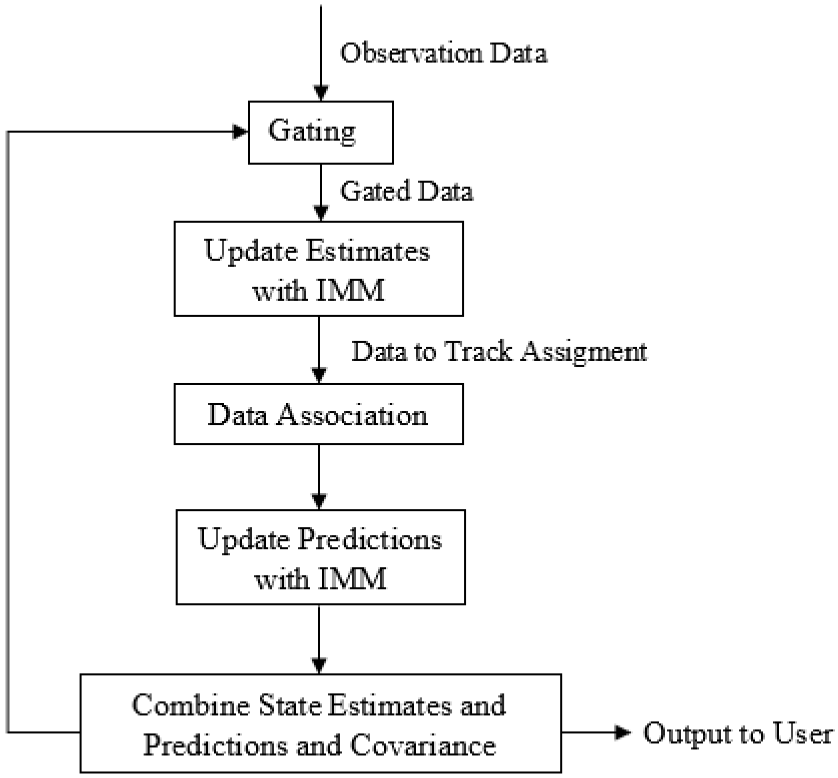

Figure 1: The general flowchart of a target tracking algorithm [23].

3.1 Gating

Gating [24] is a method to eliminate unlikely measurements. The measurements can be false or belong to other targets or unwanted objects such as buildings and mountains. The gate is the region created around the predicted measurement of a target in which measurements are allowed to be assigned to the target. All measurements in the gate are used to update the track by using data association methods. The most important benefit of the gating is to reduce later computations. In this study, we use ellipsoidal gates which can be formed as follows.

Given the measurement prediction $\hat{z}_{k|k-1}$ and innovation covariance $S_{k|k-1}$, for ellipsoidal gating, the condition in (3.1) have to be satisfied.

$$(z_k - \hat{z}_{k|k-1})^T {S_{k|k-1}}^{-1} (z_k - \hat{z}_{k|k-1}) \leq \gamma, \tag{3.1}$$

where z_k denotes a candidate measurement. γ is the gating threshold that is calculated using inverse cumulative distribution function of the Chi-Square distribution with the degrees of freedom (measurement dimension) evaluated at gating probability. The inverse chi-square cdf for a given probability p and ν degrees of freedom is given as

$$\gamma = f^{-1}(p|v) = \{x: f(x|v) = p\},$$

$$f(x|v) = p = \int_0^x \frac{t^{\frac{(v-2)}{2}} e^{-\frac{t}{2}}}{2^{\frac{v}{2}}\,\Gamma(\frac{v}{2})} dt. \tag{3.2}$$

Here, the probability p represents gating probability P_G and ν is degree of freedom which represents the dimension of the measurement.

Gating threshold γ is shown for some degrees of freedom and the gating probability $P_G = 0.9$ in Figure 2.

Figure 3 shows three predicted measurements with their gates and the measurements exceeding the threshold γ. Some gates can intersect each other and have common measurements as illustrated in Figure 3.

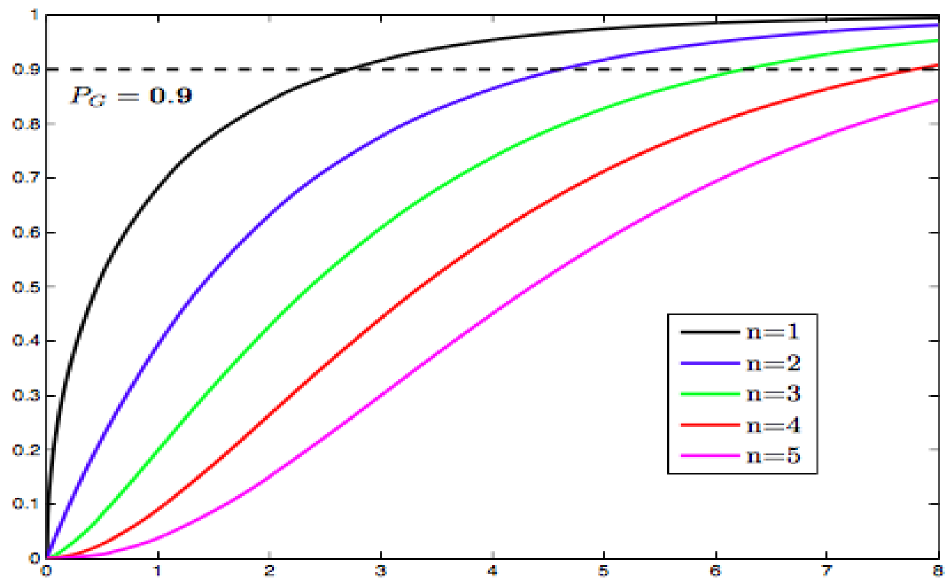

Figure 2: Gating Threshold Values [25].

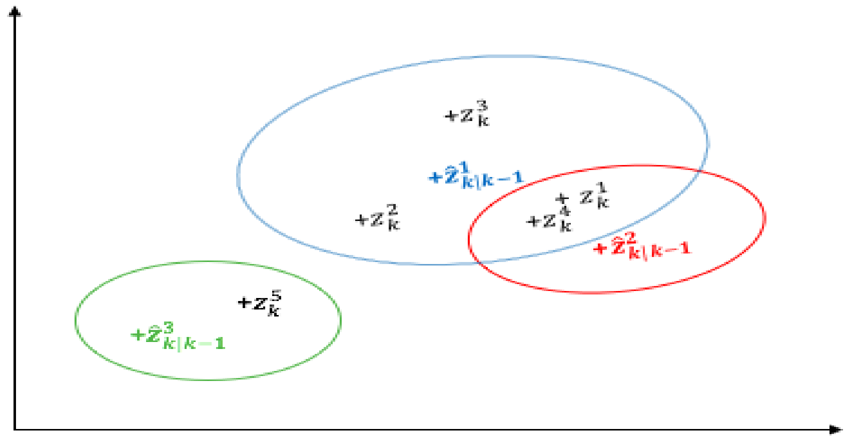

Figure 3: Example Ellipsoidal Gates.

3.2 Kalman Filter

Kalman filter is generally used in the target tracking algorithms for estimating kinematic variables such as position and velocity [23][24][26][27][29]. Kalman filter estimates the state vectors of a linear dynamic system recursively. Kalman filter assumes that measurement noise w_k is additive, zero mean and white with known covariance R_k. Kalman filter also assumes that process noise v_k is additive, zero mean and white with known covariance Q_k like the measurement noise. Kalman filter assumes that the state x_k of the target, at time k is related to the prior state x_{k-1} at time k-1, according to the equation,

$$x_k = Fx_{k-1} + B_k u_k + v_k. \tag{3.3}$$

The measurement is modeled as

$$z_k = H_k x_k + w_k. \tag{3.4}$$

Because it is not possible to observe the state x_k directly, Kalman filter determines x_k by fusing information of the system model and the noisy measurement. Kalman filter also calculates the covariance matrix P_k of the posterior state distribution which consists of the variance and covariances related to the state vector components. The diagonals of the P_k represent the variance of the elements of the state vector such as position, velocity etc. Other terms except the diagonals of P_k represent the covariance between the terms in the state vector. For linear systems with measurement and process noise modelled by zero mean Gaussian distributions, it is known that Kalman filter is the minimum mean square error (MMSE) optimum solution [28].

Kalman filter recursions are composed of two stages. The first step is the state and measurement prediction which is also called the time update or prediction update; the second step is the estimation which is also called measurement update. Kalman filter starts with the initial values of the state estimate and the state covariance matrix shown as $x_{0|0}$ and $P_{0|0}$ respectively. The equations for the two updates of the Kalman filter are given in the following.

Prediction Update

$$x_{k|k-1} = F_k x_{k-1|k-1} + B_k u_k, \tag{3.5}$$

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k. \tag{3.6}$$

Measurement Update

$$x_{k|k} = x_{k|k-1} + K_k (z_k - \hat{z}_{k|k-1}), \tag{3.7}$$

$$P_{k|k} = P_{k|k-1} - K_k S_{k|k-1} K_k^T, \tag{3.8}$$

where

$$\hat{z}_{k|k-1} = H_k x_{k|k-1}, \tag{3.9}$$

$$S_{k|k-1} = H_k P_{k|k-1} H_k^T + R, \tag{3.10}$$

$$K_k = P_{k|k-1} H_k^T S_{k|k-1}^{-1}. \tag{3.11}$$

Kalman filter algorithm is summarized in Figure 4 which is taken from [27].

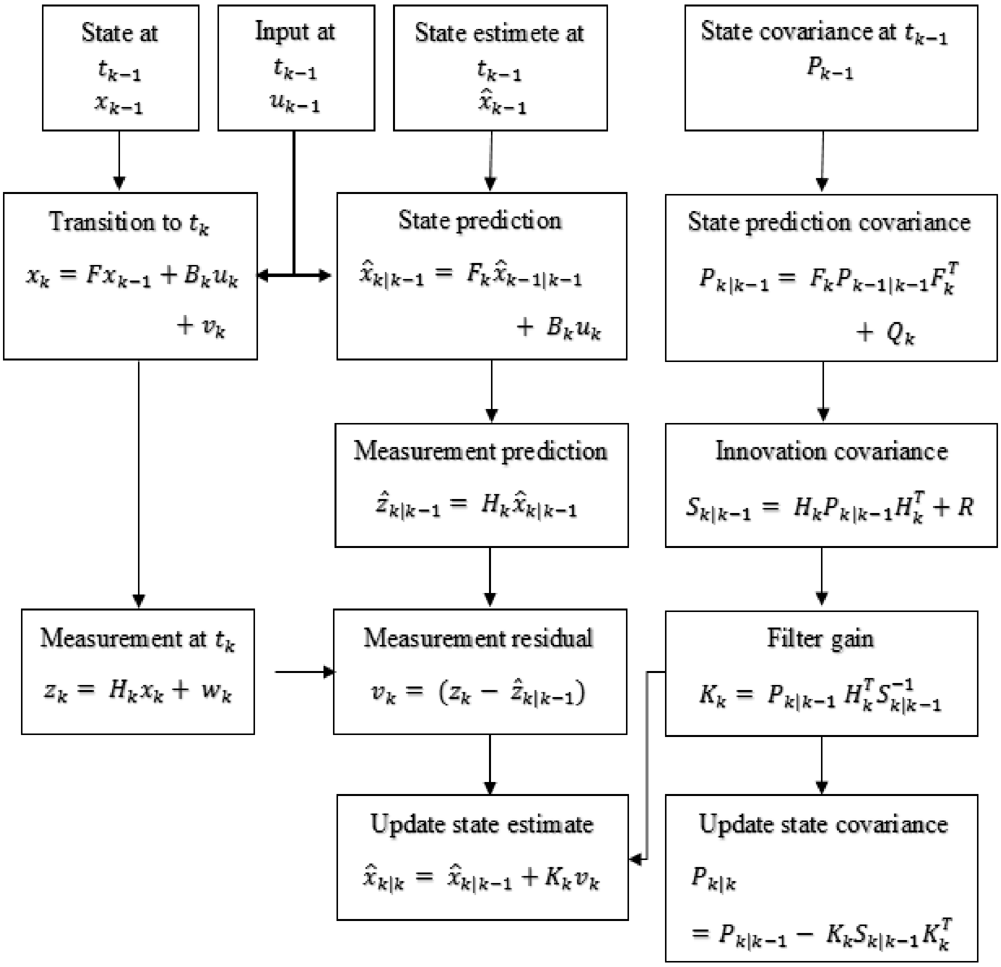

Figure 4: Kalman Filter Algorithm [27].

3.3 Interacting Multiple Model Filtering

Interacting Multiple Model (IMM) filter is a state estimation algorithm for the Jump Markov linear system given below.

$$x_{k+1} = F(r_{k+1})\, x_k + B(r_{k+1})\, u_k + v_k, \tag{3.12}$$

$$y_k = H(r_k)\, x_k + D(r_k)\, w_k, \tag{3.13}$$

where x_k is the state to be estimated. r_k is a discrete variable taking values in the set {1, 2, …, n} determining the model of the target. r_k is modelled as a Markov chain with transition probability matrix $\pi = [p_{ij}]$, $1 \leq i,j \leq n$ where n is the number of the models.

IMM filter at each time step holds the model conditioned state estimates $x^i_{k|k}$, model conditioned covariances $P^i_{k|k}$ and mode probabilities $\mu^i_{k|k}$. These quantities are defined as

$$x^i_{k|k} = E[x_k | r_k = i, z_{0:k}], \tag{3.14}$$

$$P^i_{k|k} = E[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T | r_k = i, z_{0:k}], \tag{3.15}$$

$$\mu^i_{k|k} = P[r_k = i, z_{0:k}]. \tag{3.16}$$

Given $\{x^i_{k|k}, P^i_{k|k}, \mu^i_{k|k}\}_{i=1}^n$, IMM filter calculates the new quantities $\{x^i_{k+1|k+1}, P^i_{k+1|k+1}, \mu^i_{k+1|k+1}\}_{i=1}^n$ as follows [24].

1. Calculation of mixing probabilities

   Mixing probability is conditional probability which represents that the target switches from model i to model j.

$$\mu^{i|j}_{k-1|k-1} = \frac{1}{\bar{c}_j} p_{ij}\mu^i_{k-1} \qquad i,j = 1,2,\dots,n, \tag{3.17}$$

$$\bar{c}_j = \sum_{j=1}^{n} p_{ij}\mu^i_{k-1} \qquad i,j = 1,2,\dots,n \tag{3.18}$$

where n is the number of models.

2. Mixing

   Mixing is the initiation of the state vector and the covariance matrix for each IMM model by using mixing probability,

$$x^{0j}_{k-1|k-1} = \sum_{i=1}^{n} x^i_{k-1|k-1}\,\mu^{i|j}_{k-1|k-1}, \quad j = 1,2,\dots,n, \tag{3.19}$$

$$P^{0j}_{k-1|k-1} = \sum_{i=1}^{n} \mu^{i|j}_{k-1|k-1} \Big(P^{i}_{k-1|k-1} + \left(x^{i}_{k-1|k-1} - x^{0j}_{k-1|k-1}\right)\left(x^{i}_{k-1|k-1} - x^{0j}_{k-1|k-1}\right)^{T}\Big) \quad j = 1,2,\dots,n. \tag{3.20}$$

3. <u>Mode-matched filtering</u>

Mode-matched filtering is the step which calculates the state vector and covariance matrix for each IMM model. Whereas $x^{0j}_{k-1|k-1}$ and $P^{0j}_{k-1|k-1}$are inputs of the mode matched filter, $x^{j}_{k|k}$ and $P^{j}_{k|k}$ are the outputs. In addition, normal distributed likelihood function λ^{j}_{k} is calculated in this step.

$$\lambda^{j}_{k} \triangleq \mathcal{N}(z_k; \hat{z}^{j}_{k|k-1}, S^{j}_{k|k-1}). \tag{3.21}$$

<u>Kalman Filter Equations</u>

- Prediction Update

$$x^{j}_{k|k-1} = F\, x^{0j}_{k-1|k-1}, \tag{3.22}$$

$$P^{j}_{k|k-1} = F\, P^{0j}_{k-1|k-1} F^{T} + G^{j} Q {G^{j}}^{T}. \tag{3.23}$$

- Measurement Update

$$x^{j}_{k|k} = x^{j}_{k|k-1} + K^{j}_{k}(z_k - \hat{z}^{j}_{k|k-1}), \tag{3.24}$$

$$P^{j}_{k|k} = P^{j}_{k|k-1} - K^{j}_{k} S^{j}_{k|k-1} {K^{j}_{k}}^{T}, \tag{3.25}$$

where

$$\hat{z}^{j}_{k|k-1} = H_k x^{j}_{k|k}, \tag{3.26}$$

$$S^{j}_{k|k-1} = H_k P^{j}_{k|k-1} H^{T}_{k} + R, \tag{3.27}$$

$$K^{j}_{k} = P^{j}_{k|k-1} H^{T}_{k} (S^{j}_{k|k-1})^{-1}. \tag{3.28}$$

4. Mode probability update

Mode probability is updated by using the likelihood λ_k^j.

$$\mu_k^j = \frac{1}{c}\lambda_k^j \bar{c}_j, \tag{3.29}$$

where $c = \sum_{j=1}^{n} \lambda_k^j \bar{c}_j$ is a normalization constant for the mode probabilities.

5. Combination of estimates and covariances

The state vector and the covariance matrices calculated for each model are combined to calculate the IMM filter output.

$$x_{k|k} = \sum_{j=1}^{n} x_{k|k}^j \mu_k^j, \tag{3.30}$$

$$P_{k|k} = \sum_{j=1}^{n} \mu_k^j (P_{k|k}^j + (x_{k|k}^j - x_{k|k})(x_{k|k}^j - x_{k|k})^T). \tag{3.31}$$

3.4 Probabilistic Data Association Filter

Probabilistic Data Association (PDA) filter is a data association algorithm used for tracking targets under clutter and missing measurements [30]. The kinematic information of a target is estimated using the measurements in the validation region, i.e., the gate. The problem is that there might be multiple measurements with false alarms and clutter in addition to the correct measurement or there is no measurement.

PDA assumes that there is only one target which has already been initialized. A measurement in the validation region can belong to the target or it is a false alarm. There is a measurement belonging to the target in the gate with detection probability P_D. At most a single measurement in the gate can belong to the target.

In order to update the target state PDA generates the association hypothesis H_i which are given as,

H_0 : no measurement in the gate belongs to the target, i.e., all measurements are false alarms.

H_i: i^{th} measurement belongs to the target and the rest of the measurements are false alarms. i = 1, 2 ,…, m where m is the number of measurements in the gate.

PDA then calculates the posterior probabilities of these hypotheses defined as

$$\beta_k^i \triangleq p(H_i|z_{0:k}), \tag{3.32}$$

where z_k denotes the set of measurements at time k. PDA also calculates the hypothesis conditioned estimates and covariances defined as

$$x_{k|k}^i = E\,[x_k|H_i, z_{0:k}], \tag{3.33}$$

$$P_{k|k}^i = E[(x_k - x_{k|k}^i)(x_k - x_{k|k}^i)^T|H_i, z_{0:k}]. \tag{3.34}$$

PDA then combines these estimates and covariances using hypothesis probabilities to obtain the final estimate and covariance as follows.

$$x_{k|k} = \sum_{i=1}^{m} \mu_k^i\, x_{k|k}^i, \tag{3.35}$$

$$P_{k|k} = \sum_{i=1}^{m} \mu_k^i\,(P_{k|k}^i + (x_k - x_{k|k}^i)(x_k - x_{k|k}^i)^T). \tag{3.36}$$

A single step of the PDA filter is given as follows.

1- The association probabilities β_k^i are calculated as follows,

$$\beta_k^i = \begin{cases} P_D p_{k|k-1}(z_k^i), & i > 0 \\ P_{FA}(1 - P_d P_G), & i = 0 \end{cases}, \tag{3.37}$$

where

$$p_{k|k-1}(z_k^i) = \mathcal{N}(z_k, \hat{z}_{k|k-1}, S_{k|k-1}). \tag{3.38}$$

$p_{k|k-1}(z_k^i)$ is the innovation likelihood where S is covariance matrix of innovation. P_d is detection probability and P_G is probability that correct detection is in the validation region.

2- Hypothesis conditioned estimates and covariances are calculated as follows

$$x^i_{k|k} = x_{k|k-1} + K_k(z^i_k - \hat{z}_{k|k-1}), \tag{3.39}$$

$$P^i_{k|k} = P_{k|k-1} - K_k S_{k|k-1} K^T_k, \tag{3.40}$$

where

$$\hat{z}_{k|k-1} = H_k x_{k|k-1}, \tag{3.41}$$

$$S_{k|k-1} = H_k P_{k|k-1} H^T_k + R, \tag{3.42}$$

$$K_k = P_{k|k-1} H^T_k S^{-1}_{k|k-1}. \tag{3.43}$$

3- Overall state estimate is calculated as,

$$x_{k|k} = \sum_{i=0}^{m_k} \beta^i_k x^i_{k|k} = x_{k|k-1} + K_k\left(z^{eq}_k - \hat{z}_{k|k-1}\right), \tag{3.44}$$

where

$$z^{eq}_k = \beta^0_k \hat{z}_{k|k-1} + \sum_{i=1}^{m_k} \mu^i_k z^i_k. \tag{3.45}$$

4- Overall state covariance matrix is calculated as,

$$\begin{aligned} P_{k|k} &= \sum_{i=1}^{m_k} \beta^i_k [P^i_{k|k} + (x^i_{k|k} - x_{k|k})(x^i_{k|k} - x_{k|k})^T], \\ &= \sum_{i=1}^{m_k} \beta^i_k P^i_{k|k} + \sum_{i=1}^{m_k} \beta^i_k (x^i_{k|k} - x_{k|k})(x^i_{k|k} - x_{k|k})^T]. \end{aligned} \tag{3.46}$$

3.5 IMM and PDA Combination

IMM-PDA filter is an algorithm which is used for tracking maneuvering targets under clutter and missing measurement. IMM-PDA is basically the combination of IMM and PDA filters. A single step of IMM-PDA filter is given as follows,

1- <u>Calculation of mixing probabilities</u>

Mixing probability is the conditional probability which represents the target model switch probability from model l to model j.

$$\mu^{l|j}_{k-1|k-1} = \frac{1}{\bar{c}_j} p_{lj}\mu^{l}_{k-1}, \qquad l, j = 1,2, \dots, n, \tag{3.47}$$

$$\bar{c}_j = \sum_{j=1}^{n} p_{lj}\mu^{l}_{k-1}, \qquad l, j = 1,2, \dots, n, \tag{3.48}$$

where n is the number of models.

2- Mixing

Mixing is the initiation of the state vector and the covariance matrix for each IMM model by using mixing probability,

$$x^{0j}_{k-1|k-1} = \sum_{l=1}^{n} x^{l}_{k-1|k-1}\, \mu^{l|j}_{k-1|k-1}, \qquad j = 1,2, \dots, n, \tag{3.49}$$

$$P^{0j}_{k-1|k-1} = \sum_{i=1}^{n} \mu^{l|j}_{k-1|k-1}\Big(P^{l}_{k-1|k-1} + \left(x^{l}_{k-1|k-1} - x^{0j}_{k-1|k-1}\right)\left(x^{l}_{k-1|k-1} - x^{0j}_{k-1|k-1}\right)^{T}\Big), \qquad j = 1,2, \dots, n. \tag{3.50}$$

3- Mode-matched filtering

$$\lambda^{j,i}_{k} \triangleq \mathcal{N}\left(z^{i}_{k}; \hat{z}^{j}_{k|k-1}, S^{j}_{k|k-1}\right), \qquad i = 1,2,\dots, m_k,\ j=1,\dots,n. \tag{3.51}$$

where m_k is the number of measurements.

Kalman Filter Equations

- Prediction Update

$$x^{j}_{k|k-1} = F\, x^{0j}_{k-1|k-1}, \tag{3.52}$$

$$P^{j}_{k|k-1} = F\, P^{0j}_{k-1|k-1} F^{T} + G^{j} Q {G^{j}}^{T}. \tag{3.53}$$

The association probabilities β^{i}_{k} are calculated for j^{th} IMM model as follows,

$$\beta_k^{j,i} = \begin{cases} P_D p_{k|k-1}(z_k^i), & i > 0 \\ P_{FA}(1 - P_d P_G), & i = 0 \end{cases} , \tag{3.54}$$

where

$$p_{k|k-1}(z_k^i) = \mathcal{N}(z_k^i, \hat{z}_{k|k-1}^j, S_{k|k-1}^j). \tag{3.55}$$

is the innovation likelihood where S is covariance matrix of innovation. P_d is detection probability and P_G is probability that correct detection is in the validation region.

- Measurement Update

Overall state estimate for j^{th} IMM model is calculated as,

$$x_{k|k}^j = \sum_{i=0}^{m_k} \beta_k^{j,i} x_{k|k}^{j,i} = x_{k|k-1}^j + K_k \left(z_k^{j,eq} - \hat{z}_{k|k-1}^j \right), \tag{3.56}$$

where

$$z_k^{j,eq} = \beta_k^{j,0} \hat{z}_{k|k-1}^j + \sum_{i=1}^{m_k} \beta_k^{j,i} z_k^i. \tag{3.57}$$

Overall state covariance matrix for j^{th} IMM model is calculated as,

$$\begin{aligned} P_{k|k}^j &= \sum_{i=1}^{m_k} \beta_k^{j,i} [P_{k|k}^{j,i} + (x_{k|k}^{j,i} - x_{k|k}^j)(x_{k|k}^{j,i} - x_{k|k}^j)^T], \\ &= \sum_{i=1}^{m_k} \beta_k^{j,i} P_{k|k}^{j,i} + \sum_{i=1}^{m_k} \beta_k^{j,i} (x_{k|k}^{j,i} - x_{k|k}^j)(x_{k|k}^{j,i} - x_{k|k}^j)^T]. \end{aligned} \tag{3.58}$$

4- Mode probability update

Mode probability is updated by using the likelihood.

$$\mu_k^j = \frac{1}{c} \lambda_k^j \bar{c}_j, \tag{3.59}$$

where $c = \sum_{j=1}^{n} \lambda_k^j \bar{c}_j$ is the normalization constant for the mode probability where,

$$\lambda_k^j = \sum_{i=0}^{m_k} \mu_k^{j,i} \lambda_k^{j,i}, \tag{3.60}$$

where

$$\lambda_k^{j,i} \triangleq \mathcal{N}\left(z_k^i; \hat{z}_{k|k-1}^j, S_{k|k-1}^j\right). \tag{3.61}$$

5- <u>Combination of estimates and covariances</u>

The state vector and the covariance matrices calculated for each model are combined to calculate the IMM-PDA filter's output.

$$x_{k|k} = \sum_{j=1}^{n} x_{k|k}^j \, \mu_k^j, \tag{3.62}$$

$$P_{k|k} = \sum_{j=1}^{n} \mu_k^j \left(P_{k|k}^j + (x_{k|k}^j - x_{k|k})(x_{k|k}^j - x_{k|k})^T\right). \tag{3.63}$$

CHAPTER 4

EXTENDED TARGET TRACKING BY USING RANDOM MATRICES

In recent radar systems, the assumption that one target has one measurement is not valid if the resolution of the radar is higher compared to the spatial extension of the target. In modern radars, the target can cause multiple detections from scan to scan. From these multiple detections target's shape information can be estimated. In this thesis, we assume that the shape of the extended targets is an ellipsoid and we use a symmetric positive definite random (SPD) matrix for representing the extended target shape following the recent research direction in extended target tracking. In this chapter we give an overview of the extended target tracking approaches using random matrices in the literature.

4.1 Koch's Approach

In this method, the size of the target represented by a SPD matrix (X_k) is calculated in addition to kinematics (position and velocity) of the target, recursively. This is the first approach using a SPD matrix to represent the target extent and uses the assumption that measurement noise is Gaussian distributed with covariance equal to the target extent covariance.

In Bayesian target tracking, estimation of the kinematic properties of the target is done by calculating the conditional posterior probability density function $p(x_k|z_{0:k})$ of the state vector. In Koch's approach [6], the target size is estimated by calculating the joint probability density function $p(x_k, X_k|Z_{0:k})$ of the kinematic and extent states iteratively. Here, x_k contains the kinematic properties of the target which includes the positions and the velocities in a two dimensional Cartesian coordinate system. $Z_k = \{z_k^i\}_{i=1}^{n_k}$ represents the set of n_k two dimensional position measurements at each time k. The measurements are conditionally independent for each time k and are corrupted by Gaussian noise w_k^i with zero mean and covariance X_k.

$$z_k^i = Hx_k + w_k^i. \tag{4.1}$$

Hence, the measurement likelihood is the normal density given as

$$p(Z_k|n_k, x_k, X_k) = \prod_{i=1}^{n_k} \mathcal{N}\left(z_k^i; Hx_k, X_k\right), \tag{4.2}$$

$$p(Z_k|n_k, x_k, X_k) \propto \mathcal{N}\left(\bar{z}_k; Hx_k, \frac{X_k}{n_k}\right) x W(\bar{Z}_k; n_k - 1, X_k), \tag{4.3}$$

where $\bar{z}_k$ is the mean of measurements and $\bar{Z}_k$ is measurement spread defined as

$$\bar{z}_k = \frac{1}{n_k} \sum_{i=1}^{n_k} z_k^i, \tag{4.4}$$

$$\bar{Z}_k = \sum_{i=1}^{n_k} (z_k^i - \bar{z}_k)(z_k^i - \bar{z}_k)^T. \tag{4.5}$$

The Wishart distribution $W(T; m, C)$ used above is defined as follows,

$$W(T; m, C) = \frac{|T|^{\frac{m-p-1}{2}}}{|C|^{\frac{m}{2}} 2^{\frac{mp}{2}} \Gamma_p\left(\frac{m}{2}\right)} \exp\left[\mathrm{tr}\left(-\frac{1}{2} TC^{-1}\right)\right], \tag{4.6}$$

where the parameter m denotes the degrees of freedom. $|T|$ represents the determinant of a square matrix T, $\mathrm{tr}(T)$ is the trace of a square matrix T and

$$\Gamma_p(a) = \pi^{\frac{1}{4} p(p-1)} \prod_{j=1}^{p} \Gamma\left[a + \frac{(1-j)}{2}\right], \tag{4.7}$$

is the multivariate generalization of the Gamma function Γ. Also, note that $m > p - 1$ to make Wishart distrubuion well-defined. p is the size of the matrix T (i.e., $T \in R^{pxp}$).

The update equations are derived by the approach of conjugate priors which is applied to the measurement model in (4.2). The results of the derivations are given below. Firstly joint density $p(x_k, X_k|Z_{0:k})$ is factored as,

$$p(x_k, X_k|Z_{0:k}) = p(x_k|X_k, Z_{0:k})\, p(X_k|Z_{0:k}). \tag{4.8}$$

The matrix variate density $p(X_k|Z_{0:k})$ in (4.8) is assumed to be given as

$$p(X_k|Z_{0:k}) = \mathrm{IW}\big(X_k; v_{k|k}, \widetilde{X}_{k|k}\big). \tag{4.9}$$

where the Inverse Wishart distribution denoted by $\mathrm{IW}(T; m, C)$ is defined as

$$\mathrm{IW}(T; m, C) = \frac{|C|^{\frac{m}{2}}}{|T|^{\frac{m+p+1}{2}} 2^{\frac{mp}{2}} \Gamma_p\left(\frac{m}{2}\right)} \exp\left[\mathrm{tr}\left(-\frac{1}{2} C T^{-1}\right)\right], \tag{4.10}$$

where the parameter m denotes the degrees of freedom. The vector variate kinematic density is given as,

$$p(x_k|X_k, Z_{0:k}) = \mathcal{N}\big(x_k; x_{k|k}, \widetilde{P}_{k|k} \otimes X_k\big), \tag{4.11}$$

where, $\otimes$ denotes Kronecker product [32].

With the forms of the posterior densities given above, the update equations for the kinematic and extent quantities are given below.

<u>Prediction Update:</u> In this update, the previous updated density $p(x_{k-1}, X_{k-1}|Z_{0:k-1})$ which is defined as

$$\begin{gathered} p(x_{k-1}, X_{k-1}|Z_{0:k-1}) = \\ \mathcal{N}\big(x_{k-1}; x_{k-1|k-1}, \widetilde{P}_{k-1|k-1} \otimes X_{k-1}\big)\, \mathrm{IW}\big(X_{k-1}; v_{k-1|k-1}, \widetilde{X}_{k-1|k-1}\big), \end{gathered} \tag{4.12}$$

is updated to obtain the predicted density $p(x_k, X_k|Z_{0:k-1})$ which is defined as

$$\begin{gathered} p(x_k, X_k|Z_{0:k-1}) = \\ \mathcal{N}\big(x_k; x_{k|k-1}, \widetilde{P}_{k|k-1} \otimes X_k\big)\, \mathrm{IW}\big(X_k; v_{k|k-1}, \widetilde{X}_{k|k-1}\big). \end{gathered} \tag{4.13}$$

This update is carried out using the following formulas,

$$x_{k|k-1} = F x_{k-1|k-1}, \tag{4.14}$$

$$\widetilde{P}_{k|k-1} = F \widetilde{P}_{k|k-1} F^{T} + \widetilde{Q}, \tag{4.15}$$

where F denotes target kinematic state transition matrix and Q is the process noise covariance.

In the extent prediction, the update equations are given as follows.

$$\widetilde{X}_{k|k-1} = \frac{v_{k|k-1}-d-1}{v_{k-1|k-1}-d-1}\,\widetilde{X}_{k-1|k-1}, \tag{4.16}$$

$$v_{k|k-1} = e^{(-T/\tau)} v_{k-1|k-1}. \tag{4.17}$$

Here, $v_{k|k}$ is the extension degrees of freedom parameter; τ is a time constant related to the agility of extension and T denotes sampling time.

Measurement Update: In this update, the predicted density $p(x_k, X_k|Z_{0:k-1})$ which is defined as

$$\begin{gathered} p(x_k, X_k|Z_{0:k-1}) = \\ \mathcal{N}\left(x_k; x_{k|k-1}, \widetilde{P}_{k|k-1} \otimes X_k\right) \mathrm{IW}\left(X_k; v_{k|k-1}, \widetilde{X}_{k|k-1}\right). \end{gathered} \tag{4.18}$$

is updated to obtain the density $p(x_k, X_k|Z_{0:k})$ which is defined as

$$p(x_k, X_k|Z_{0:k}) = \mathcal{N}\left(x_k; x_{k|k}, \widetilde{P}_{k|k} \otimes X_k\right) \mathrm{IW}\left(X_k; v_{k|k}, \widetilde{X}_{k|k}\right). \tag{4.19}$$

This update is carried out using the following formulas,

$$x_{k|k} = x_{k|k-1} + (\widetilde{K}_{k|k-1} \otimes I_d)(\bar{z}_k - H x_{k|k-1}), \tag{4.20}$$

$$\widetilde{P}_{k|k} = \widetilde{P}_{k|k-1} - \widetilde{K}_{k|k-1} \widetilde{S}_{k|k-1} \widetilde{K}^T_{k|k-1}, \tag{4.21}$$

where

$$\widetilde{S}_{k|k-1} = \widetilde{H} \widetilde{P}_{k|k-1} \widetilde{H}^T + \frac{1}{n_k} \quad \text{and } S_{k|k-1} = \widetilde{S}_{k|k-1} X_k, \tag{4.22}$$

$$\widetilde{K}^T_{k|k-1} = \widetilde{P}_{k|k-1} \widetilde{H}^T \widetilde{S}^{-1}_{k|k-1} \quad \text{and} \quad K_{k|k-1} = \widetilde{K}_{k|k-1} \otimes I_d, \tag{4.23}$$

$$\widetilde{H} = [1\ 0] \quad \text{and} \quad H = \widetilde{H} \otimes I_d. \tag{4.24}$$

The update equations for the extension are given as,

$$v_{k|k} = v_{k|k-1} + n_k, \tag{4.25}$$

$$\widetilde{X}_{k|k} = \widetilde{X}_{k|k-1} + \widetilde{S}^{-1}_{k|k-1} N_{k|k-1} + \bar{Z}_k. \tag{4.26}$$

Here $N_{k|k-1}$ is innovations product which is defined as

$$N_{k|k-1} = (\bar{z}_k - Hx_{k|k-1})(\bar{z}_k - Hx_{k|k-1})^T. \quad (4.27)$$

The extent estimate $X_{k|k}$ is then calculated as

$$X_{k|k} = \frac{\widetilde{X}_{k|k}}{v_{k|k} - d - 1}. \quad (4.28)$$

4.2 Feldmann's Approach

In Koch's approach sensor errors are not modeled and the measurement covariance is assumed to be equal to the target extent matrix. Hence if there is high amount of sensor error the estimator will overestimate the target extent. Because of this, Feldmann's approach model the sensor error separately as opposed to Koch's approach [7]. Since the measurements are conditionally independent the measurement likelihood is given as

$$p(Z_k|n_k, x_k, X_k) = \prod_{i=1}^{n_k} \mathcal{N}(z_k^i; Hx_k, sX_k + R). \quad (4.29)$$

Here, in addition to the explicit consideration of the measurement noise, there is also a scaling factor 's' to adjust the contribution of the extension on the measurement spread. In order to account for scaling factor 's', the measurements are shown in Figure 5 and Figure 6. In Figure 5 and Figure 6, the spreads of 1200 measurements based on the SPD matrix X = diag $((300\ m/2)^2$, $((150\ m\ /\ 2)^2)$ with $Z_k \sim \mathcal{N}(Z_k; 0, X_k)$ and $Z_k \sim \mathcal{N}(Z_k; 0, X_k/4)$ are shown respectively [7].

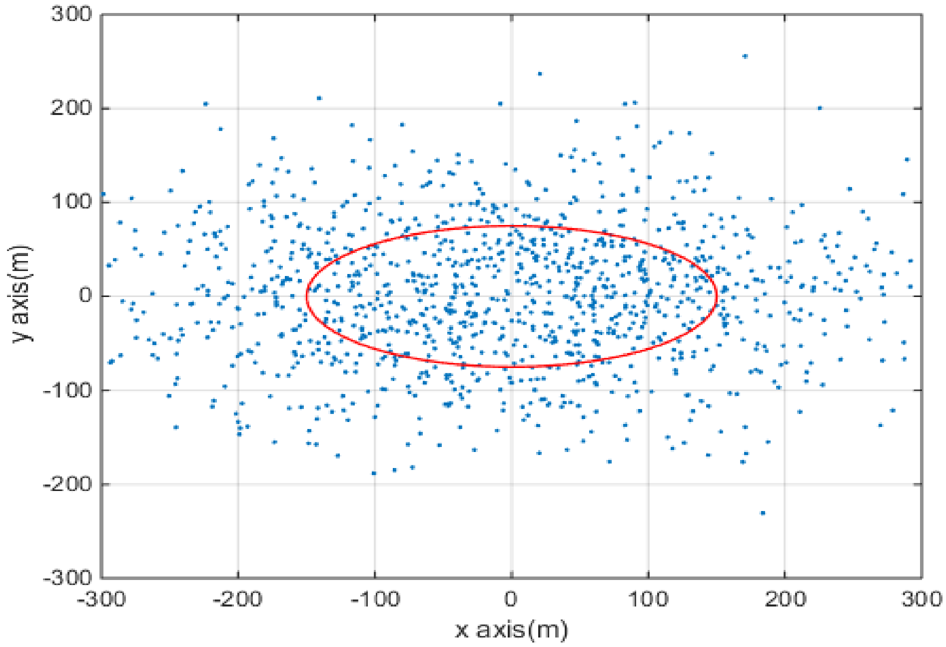

Figure 5: Measurement Model $\mathcal{N}(Z_k; 0, X_k)$ [7].

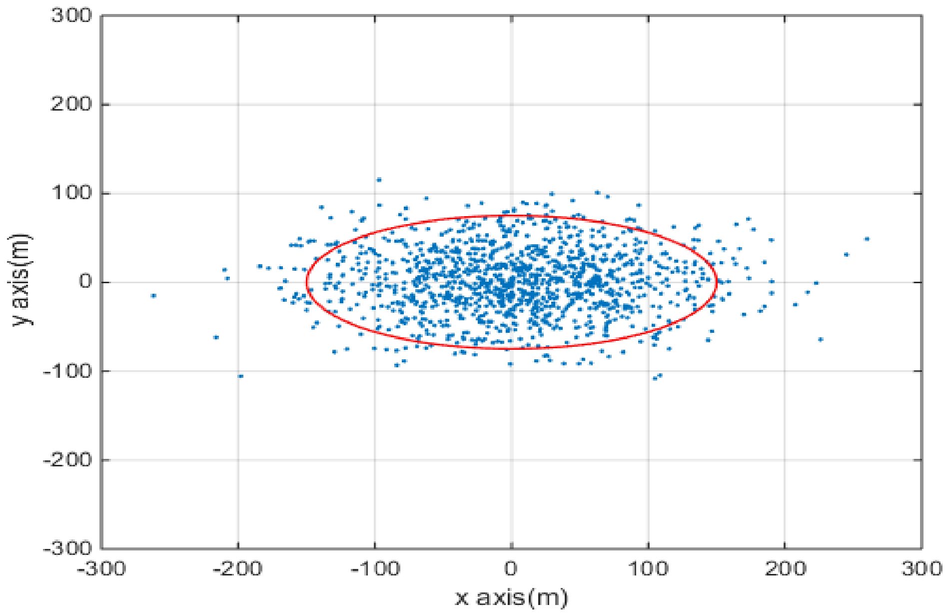

Figure 6: Measurement Model $\mathcal{N}(Z_k; 0, X_k/4)$ [7].

For the prediction update in Feldmann's approach, there is the assumption that the estimations for kinematics and extension are independent. Thus the vector variate density $p(x_k|X_k, Z_k)$ can be expressed as below,

$$p(x_k|X_k, Z_{0:k}) \approx p(x_k|Z_{0:k}) \approx \mathcal{N}(x_k; x_{k|k}, P_{k|k}). \tag{4.30}$$

<u>Prediction Update:</u> In this update, the previous updated density $p(x_{k-1}, X_{k-1}|Z_{0:k-1})$ which is defined as

$$p(x_{k-1}, X_{k-1}|Z_{0:k-1}) = \mathcal{N}(x_{k-1}; x_{k-1|k-1}, P_{k-1|k-1})\, \mathrm{IW}(X_{k-1}; v_{k-1|k-1}, X_{k-1|k-1}), \tag{4.31}$$

is updated to obtain the predicted density $p(x_k, X_k|Z_{0:k-1})$ which is defined as

$$p(x_k, X_k|Z_{0:k-1}) = \mathcal{N}(x_k; x_{k|k-1}, P_{k|k-1})\, \mathrm{IW}(X_k; v_{k|k-1}, X_{k|k-1}). \tag{4.32}$$

This update is carried out using the following formulas,

$$x_{k|k-1} = Fx_{k-1|k-1}, \tag{4.33}$$

$$P_{k|k-1} = Fx_{k-1|k-1}F^T + Q, \tag{4.34}$$

where F denotes the target kinematic state transition matrix.

In the extent prediction, it is assumed that the mean of the extent does not change and its uncertainty increases, which gives the following update equations.

$$X_{k|k-1} = X_{k-1|k-1}, \tag{4.35}$$

$$v_{k|k-1} = 2 + \exp\left(\frac{-T}{\tau}\right)(v_{k-1|k-1} - 2). \tag{4.36}$$

Here, $v_{k|k}$ is the extension degrees of freedom parameter; τ is a time constant related to the agility of extension and T denotes sampling time.

<u>Measurement Update:</u> In this update, the predicted density $p(x_k, X_k|Z_{0:k-1})$ which is defined as

$$p(x_k, X_k|Z_{0:k-1}) = \mathcal{N}(x_k; x_{k|k-1}, P_{k|k-1})\, \mathrm{IW}(X_k; v_{k|k-1}, X_{k|k-1}). \tag{4.37}$$

is updated to obtain the density $p(x_k, X_k|Z_{0:k})$ which is defined as

$$p(x_k, X_k|Z_{0:k}) = \mathcal{N}(x_k; x_{k|k}, P_{k|k})\, \mathrm{IW}(X_k; v_{k|k}, X_{k|k}). \tag{4.38}$$

If the extension matrix X_k was a non-random matrix and known, the kinematic state could be updated by standard Kalman filter equations. Since X_k is not non-random, the measurement covariance computation part of the standard Kalman filter is modified by substituting X_k with $X_{k|k-1}$, which gives the kinematic equations below.

$$x_{k|k} = x_{k|k-1} + K_{k|k-1}\left(\bar{z}_k - Hx_{k|k-1}\right), \tag{4.39}$$

where

$$\bar{z}_k = \frac{1}{n_k}\sum_{j=1}^{n_k} z_k^j, \tag{4.40}$$

is the mean of measurements.

$$P_{k|k} = x_{k|k-1} + K_{k|k-1}S_{k|k-1}K_{k|k-1}{}^{T}, \tag{4.41}$$

where

$$S_{k|k-1} = HP_{k|k-1}H^T + \frac{Z_{k|k-1}}{n_k}, \tag{4.42}$$

where we use $\frac{Z_{k|k-1}}{n_k}$ instead of measurement noise covariance R. In fact, $Z_{k|k-1}$ includes the term R as below,

$$Z_{k|k-1} = sX_{k|k-1} + R. \tag{4.43}$$

In order to calculate the updated extension state $X_{k|k}$, the weighted sum of the innovation product $\widehat{N}_{k|k-1}$, measurement spread $\hat{Z}_{k|k-1}$ and predicted extension matrix $X_{k|k-1}$ is scaled by the extent parameter $v_{k|k}$ as follows,

$$X_{k|k} = \frac{1}{v_{k|k}}\left(v_{k|k-1}X_{k|k-1} + \widehat{N}_{k|k-1} + \hat{Z}_{k|k-1}\right), \tag{4.44}$$

$$v_{k|k} = v_{k|k-1} + n_k. \tag{4.45}$$

$\widehat{N}_{k|k-1}$ and $\hat{Z}_{k|k-1}$ which are scaled versions of $N_{k|k-1}$ (defined in (4.27)) and $\bar{Z}_k$ (defined in (4.5)) are given as follows.

$$\widehat{N}_{k|k-1} = X_{k|k-1}^{1/2}\, S_{k|k-1}^{-1/2} N_{k|k-1} S_{k|k-1}^{-1/2}{}^{T} X_{k|k-1}^{1/2}{}^{T}, \tag{4.46}$$

$$\hat{Z}_{k|k-1} = X_{k|k-1}^{1/2}\, Z_{k|k-1}^{-1/2}\, \bar{Z}_k\, Z_{k|k-1}^{-1/2}{}^{T} X_{k|k-1}^{1/2}{}^{T}. \tag{4.47}$$

4.3 Orguner's Approach

Feldmann's approach is approximate and its optimality properties are not known. The approach given in [8] applies variational Bayesian methodology to obtain an approximate measurement update for the extended target tracking problem using random matrices. The analytical posterior distribution for the problem does not exist due to the covariance summation in $\mathcal{N}\left(z_k^i; Hx_k, sX_k + R\right)$. Due to this [8] first writes this density in the following form.

$$\mathcal{N}\left(z_{0:k}^i; Hx_k, sX_k + R\right) = \int \mathcal{N}\left(z_{0:k}^i; y_k^i, R\right) \mathcal{N}\left(y_{0:k}^i; Hx_k, sX_k\right) dy_k^i \tag{4.48}$$

where y_k^i denotes the noise free measurement.

Prediction Update: In this update, the same assumptions and prediction update formulas as Feldmann's approach are used. In this update, the previous updated density $p(x_{k-1}, X_{k-1}|Z_{0:k-1})$ which is defined as

$$\begin{gathered} p(x_{k-1}, X_{k-1}|Z_{0:k-1}) = \\ \mathcal{N}\left(x_{k-1}; x_{k-1|k-1}, P_{k-1|k-1}\right) IW\left(X_{k-1}; v_{k-1|k-1}, X_{k-1|k-1}\right), \end{gathered} \tag{4.49}$$

is updated to obtain the predicted density $p(x_k, X_k|Z_{0:k-1})$ which is defined as

$$p(x_k, X_k|Z_{0:k-1}) = \mathcal{N}\left(x_k; x_{k|k-1}, P_{k|k-1}\right) IW\left(X_k; v_{k|k-1}, X_{k|k-1}\right). \tag{4.50}$$

This update is carried out using the following formulas,

$$x_{k|k-1} = Fx_{k-1|k-1}, \tag{4.51}$$

$$P_{k|k-1} = Fx_{k-1|k-1}F^T + Q, \tag{4.52}$$

where F denotes the target kinematic state transition matrix.

In the extent prediction, it is assumed that the mean of the extent does not change and its uncertainty increases, which gives the following update equations.

$$X_{k|k-1} = X_{k-1|k-1}, \tag{4.53}$$

$$v_{k|k-1} = 2 + \exp\left(\frac{-T}{\tau}\right)\left(v_{k-1|k-1} - 2\right). \tag{4.54}$$

Here, $v_{k|k}$ is the extension degrees of freedom parameter; τ is a time constant related to the agility of extension and T denotes sampling time.

Measurement Update: Orguner's approach solves this problem by computing $p(x_k, X_{k,} Y_k|Z_{0:k})$ where $Y_k = \{y_k^i\}_{i=1}^{n_k}$ is the set of noise free measurements instead of $p(x_k, X_{k,}|Z_{0:k})$ to get rid of the covariance summation. This approach uses variational approximation given as below.

$$p(x_k, X_{k,} Y_k|Z_{0:k}) \approx q(x_k, X_k, Y_k) = q_x(x_k).\, q_X(X_k).\, q_Y(Y_k) \tag{4.55}$$

where $q_x(x_k), q_X(X_k), q_Y(Y_k)$ are the posterior densities for x_k, X_k, Y_k respectively and given as follows.

$$q_x(x_k) = \mathcal{N}(x_k; x_{k|k}, P_{k|k}), \tag{4.56}$$

$$q_X(X_k) = \mathrm{IW}(X_k; v_{k|k}, X_{k|k}), \tag{4.57}$$

$$q_Y(Y_k) = \prod_{i=1}^{n_k} \mathcal{N}(y_k^i; \hat{y}_k^i, \Sigma_k^y). \tag{4.58}$$

An iterative solution is obtained to optimize densities $q_x(x_k), q_X(X_k), q_Y(Y_k)$. The derivations and details for the computation of these densities are given in [8]. In this section, the update formulas for the estimation of kinematic and extent states for the $(j+1)^{th}$ iteration are given. The results of the last iteration are used as the estimates of kinematics and extent states.

$$x_{k|k}^{(j+1)} = P_{k|k}^{(j+1)}(P_{k|k-1}^{-1} x_{k|k-1} + n_k H^T \overline{(sX_k)^{-1}} \bar{y}_k, \tag{4.59}$$

$$P_{k|k}^{(j+1)} = (P_{k|k-1}^{-1} + n_k H^T \overline{(sX_k)^{-1}} H)^{-1}, \tag{4.60}$$

where

$$\bar{y}_k = \frac{1}{n_k} \sum_{i=1}^{n_k} y_k^i, \tag{4.61}$$

$$V_{k|k}^{(j+1)} = V_{k|k-1} + \frac{1}{s} \sum_{i=1}^{n_k} \overline{(y_k^i - Hx_k)(y_k^i - Hx_{k|k})^T}, \tag{4.62}$$

$$v_{k|k}^{(j+1)} = v_{k|k-1} + n_k. \tag{4.63}$$

The expressions above use the following quantities.

$$\overline{x_k} = x_{k|k}^{(j)}, \tag{4.64}$$

$$\overline{(sX_k)^{-1}} = v_{k|k}(sV_{k|k}^{(j)})^{-1}, \tag{4.65}$$

$$\begin{aligned}\overline{(y_k^l - Hx_k)(y_k^l - Hx_{k|k})^T} = (\hat{y}_k^{i,(j)} - Hx_{k|k}^{(j)})(\hat{y}_k^{i,(j)} - Hx_{k|k}^{(j)})^T + \\ HP_{k|k}^{(j)}H^T + \Sigma_k^{y,(j)},\end{aligned} \tag{4.66}$$

where

$$\hat{y}_k^{i,(j+1)} = \Sigma_k^{y,(j+1)}(\overline{(sX_k)^{-1}}H\overline{x_k} + R^{-1}z_k^i), \tag{4.67}$$

$$\Sigma_k^{y,(j+1)} = (\overline{(sX_k)^{-1}} + R^{-1})^{-1}. \tag{4.68}$$

Initial conditions in the iteration are given as $\hat{y}_k^{i,(0)} = z_k^i$, $\Sigma_k^{y,(0)} = sX_{k|k-1}$, $x_{k|k}^{(0)} = x_{k|k-1}$, $P_{k|k}^{(0)} = P_{k|k-1}$, $v_{k|k}^{(0)} = v_{k|k-1}$, $V_{k|k}^{(0)} = V$.

CHAPTER 5

THE PROPOSED ALGORITHM

Most extended target tracking algorithms in the literature take the list of point measurements from the sensor at each time step and consider their alternative multiple partitionings/clusterings for recursive updates. While this approach gives quite good results and it is close to optimal, processing of alternative partitionings of each measurement set is very time consuming since the number of alternative partitions is quite large even after discarding most of the unlikely partitions as was done in e.g., [11]. This computation load is currently too much for practical applications.

In this thesis, we propose an alternative extended target tracking approach which uses the clustered point measurement data at each time step and applies the IMM-PDA filter to this clustered data. The clustered point measurement data is generated with a clustering algorithm in a preprocessing step. Hence the algorithm proposed here uses a single partitioning/clustering result along with an IMM-PDA filter which makes it suitable for the applications with a low-computational budget.

The structure of the proposed tracking algorithm is shown in Figure 7.

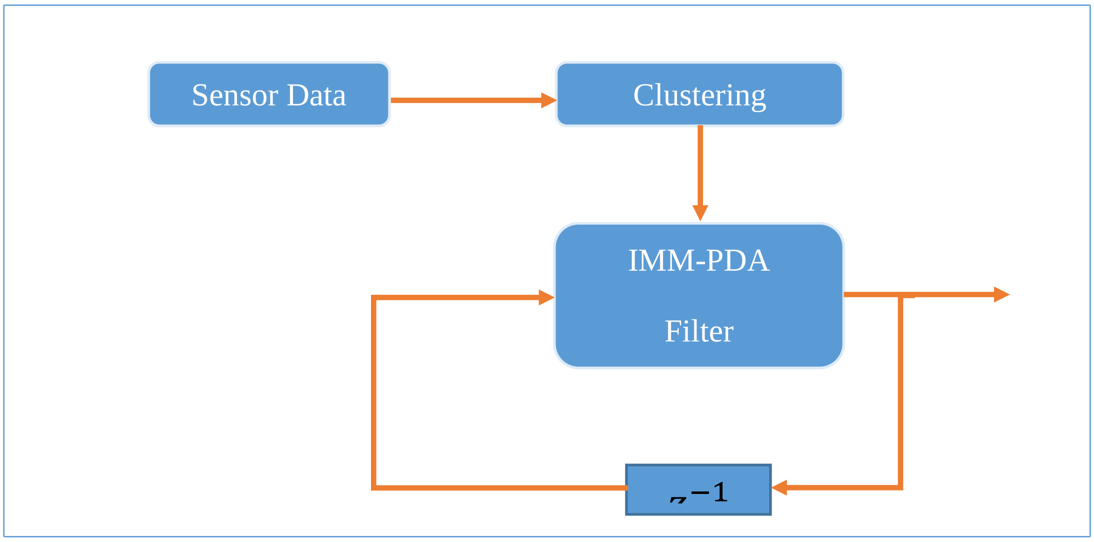

Figure 7: Structure of the extended target tracking algorithm proposed in this thesis.

Here, we only consider the scenarios which can be handled using single target tracking algorithms such as IMM-PDA for each target. Consequently, we consider only the cases where the targets are sufficiently separated compared to measurement uncertainty, the detection probability is high and false alarm probability is low. With these assumptions, in the probabilistic data association, the joint association hypotheses will not be considered as is done in joint probabilistic data association filter (JPDAF). JPDA computes the measurement-to-target association probabilities while considering the case there may be more than one target whereas PDA assumes that there is one target which may be associated with the measurement. However, it must be noted that the ideas presented here can be straightforwardly generalized to an IMM-JPDA algorithm as well.

In the following parts of this chapter, we give the details of the proposed algorithm. In Section 5.1, we describe the clustering algorithm used in the thesis and give its pseudo-code.

Although the consideration of the IMM-PDA with clustered measurements is more or less straightforward, for the calculation of the association probabilities a predictive likelihood function is necessary for the extent states. In Section 5.2

alternative predictive likelihood functions existing in the literature are examined and a new likelihood function is proposed.

When IMM-PDA filter uses clustered measurements since alternative partitionings of the raw data are not considered, close targets might have been combined at the clustering step which might lead to track loss in the IMM-PDA filter. For avoiding this problem, we give a specific algorithm which

- detects that two previously tracked targets have approached each other;
- checks if the clustering algorithm combined the measurements of the close targets in a single cluster or not;
- separates the target measurements using prior information.

The details about this algorithm which handles close targets are given in Section 5.3.

In Section 5.4, a pseudo code of one-step of the overall extended tracking algorithm is given.

5.1 Clustering Algorithm

In this section, we describe the clustering algorithm used to cluster the point measurements given by the sensor at each time step. The clustering algorithm we use is distance based. If two point measurements are separated by a distance (between them) which is smaller than or equal to a threshold, then these two point measurements are considered to lie in the same cluster. A pseudo-code of the clustering algorithm is given in Algorithm 1.

Algorithm 1: Pseudo-code of the Clustering Algorithm

{clusters} = Clustering(ValidationMatrix)

- **FOR** unclustered measurement i
    - **IF** there are measurements close enough to measurement i
        - **IF** there is no cluster

          number of cluster = 1

          Group measurements in first cluster

          Show in which cluster each measurement is

        - **END IF**
    - **ELSEIF** there is the least one cluster
        - **IF** number of cluster > 1
            - **IF** closed measurements are in different clusters

              Merge clusters and delete high cluster id

              number of cluster = number of cluster - 1

              Update in which cluster each measurement is

            - **END IF**
        - **END IF**
        - **IF** any measurement close enough to each other is in a cluster

          Add new measurements to the cluster the clustered measurement is in

          Update in which cluster each measurement is

        - **ELSE**

          number of cluster = number of cluster + 1

          Group new unclustered measurements in a new cluster

Algorithm 1 (continued)

Update in which cluster each measurement is
 	
        ▪ **<u>END IF</u>**
    ○ **<u>END IF</u>**
 	
<u>END FOR</u>

5.2 Predictive Likelihood Function Selection

In IMM-PDA filter, a predictive likelihood function is necessary for calculating the association hypothesis probabilities. The predictive likelihood function can be mathematically expressed as $p(Z_k^i|Z_{0:k-1})$ where Z_k^i is the set of point measurements in the i^{th} cluster at time k and $Z_{0:k-1}$ denotes all previous measurements. Some predictive likelihood functions proposed in the literature are described in the following subsections. Note that the predictive likelihood functions given in the thesis are approximate likelihood functions.

5.2.1 Predictive Likelihood Function 1

Feldmann [7] suggests a heuristic procedure which takes into account how well the measurement spread matches the predicted extension in order to determine the predictive likelihood. The predictive likelihood function $\mathrm{p}(Z_k^i|Z_{0:k-1})$ proposed by Feldmann is computed as follows.

$$\Delta_{k|k-1}^i := \bar{Z}_k^i\,(n_k^i - 1)\,\mathrm{Z}_{\mathrm{k|k-1}} \tag{5.1}$$

$$\mathrm{Z}_{\mathrm{k|k-1}} = (\,\mathrm{sX}_{\mathrm{k|k-1}} + \mathrm{R}) \tag{5.2}$$

where n_k^i is the number of measurements in the i^{th} cluster at time k. In addition, Feldmann defines the following quantity.

$$[\Delta Z_k^i] = \bar{Z}_k^i - (n_k^i - 1)(\mathrm{sX_{k|k-1}} + \mathrm{R}). \tag{5.3}$$

Mean of $[\Delta Z_k^i]$ is zero and variance of $[\Delta Z_k^i]$ is obtained from the Wishart density $\mathrm{W}(\bar{Z}_k^i\,; n_k^i - 1, \mathrm{sX_k} + \mathrm{R})$ as

$$\mathrm{Var}\,[\Delta Z_k^i] \approx (n_k^i - 1)(\mathrm{tr}\left(\mathrm{Z_{k|k-1}}\right)\mathrm{Z_{k|k-1}} + Z_{k|k-1}^2, \tag{5.4}$$

$$\mathrm{Var}\,[\Delta_{k|k-1}^i] \approx \mathrm{Var}\,[\Delta Z_k^i] + \mathrm{s}^2(n_k^i - 1)^2\mathrm{X_{k|k-1}v_{k|k-1}}. \tag{5.5}$$

Finally the predictive likelihood function $\mathrm{p}(Z_k^i|Z_{0:k-1})$ is given as

$$\mathrm{p}(Z_k^i|Z_{0:k-1}) \propto \mathcal{N}\left(\bar{z}_k^i, \mathrm{Hx_{k|k-1}}, \mathrm{S_{k|k-1}}\right) \left|2\pi\mathrm{Var}\left[\Delta_{k|k-1}^i\right]\right|^{-\frac{d+1}{4}}$$
$$\mathrm{etr}\left(-\tfrac{1}{2}\Delta_{k|k-1}^i\left(\mathrm{Var}\left[\Delta_{k|k-1}^i\right]\right)^{-1}\Delta_{k|k-1}^i\right). \tag{5.6}$$

5.2.2 Predictive Likelihood Function 2

In [8], Orguner proposes the predictive likelihood function $\mathrm{p}(Z_k^i|Z_{0:k-1})$ calculated as

$$\mathrm{p}(Z_k^i|Z_{0:k-1}) = \int p(Z_k^i|x_k, X_k)p(x_k, X_k|Z_{0:k-1})dx_k dX_k. \tag{5.7}$$

Since the calculation of the integral in (5.7) is not possible, the predictive likelihood is approximated as

$$\mathrm{p}(Z_k^i|Z_{0:k-1}) \approx \left|2\pi\mathrm{P_{k|k}}\right|^{\frac{1}{2}} \left|\frac{2}{\mathrm{s}}\Sigma_{\mathrm{k}}^{\mathrm{y}}\right|^{\frac{n_k^i}{2}} \frac{\left|\mathrm{V_{k|k-1}}\right|^{\frac{\mathrm{v_{k|k-1}}}{2}}}{\left|\mathrm{V_{k|k}}\right|^{\frac{\mathrm{v_{k|k}}}{2}}}$$
$$\exp\left(-\frac{1}{2}\left[\mathrm{tr}\left(\mathrm{P_{k|k}P_{k|k-1}^{-1}}\right) - \mathrm{m_x} + n_k^i\left(\mathrm{tr}\left(\Sigma_{\mathrm{k}}^{\mathrm{y}}\mathrm{R_k^{-1}}\right) - \mathrm{m_y}\right)\right]\right) \tag{5.8}$$
$$\times\ \frac{\Gamma_{\mathrm{m_y}}\left(\frac{\mathrm{v_{k|k}}}{2}\right)}{\Gamma_{\mathrm{m_y}}\left(\frac{\mathrm{v_{k|k-1}}}{2}\right)}\left(\prod_{\mathrm{j=1}}^{n_k^i}\mathcal{N}\left(\mathrm{z_k^j}; \hat{\mathrm{y}}_{\mathrm{k}}^{\mathrm{j}}, \mathrm{R}\right)\right) \times \mathcal{N}\left(\mathrm{x_{k|k}}; \mathrm{x_{k|k-1}}, \mathrm{P_{k|k}}\right).$$

All the quantities used in the likelihood function above are obtained in the last iteration of the algorithm given between (4.59) and (4.68).

5.2.3 Problem with the Existing Likelihoods and a Heuristic Fix

The predictive likelihoods given in the previous sections results in a significant problem when they are used in a PDA algorithm due to their units. In order to illustrate the problem, suppose that at time k cluster 1 and cluster 2 are associated to a track. In PDA we then generate three hypotheses.

H_0: None of the clusters belongs to the target

H_1: Cluster 1 has originated from the target and the other cluster is a false alarm.

H_2: Cluster 2 has originated from the target and the other cluster is a false alarm.

We can calculate the posterior probabilities of the hypotheses as follows.

$$p(H_0|Z_{0:k}) = \frac{\beta_{FA}(1 - P_D P_G)}{\beta_{FA}(1 - P_D P_G) + P_D p(Z_k^1|Z_{0:k-1}) + P_D p(Z_k^2|Z_{0:k-1})}, \quad (5.9)$$

$$p(H_1|Z_{0:k}) = \frac{P_D p(Z_k^1|Z_{0:k-1})}{\beta_{FA}(1 - P_D P_G) + P_D p(Z_k^1|Z_{0:k-1}) + P_D p(Z_k^2|Z_{0:k-1})}, \quad (5.10)$$

$$p(H_2|Z_{0:k}) = \frac{P_D p(Z_k^2|Z_{0:k-1})}{\beta_{FA}(1 - P_D P_G) + P_D p(Z_k^1|Z_{0:k-1}) + P_D p(Z_k^2|Z_{0:k-1})}. \quad (5.11)$$

Here as it is seen in the denominator of the probabilities, the quantities $p(Z_k^1|Z_{0:k-1})$ and $p(Z_k^2|Z_{0:k-1})$ are summed. However, if the clusters 1 and 2 have different number of measurements then these likelihoods have different units. Therefore, a unit inconsistency exists in the probability calculation operation. Notice that if the number of measurements in the i^{th} cluster is n_k^i, then the unit of $p(Z_k^i|Z_{0:k-1})$ is $\frac{1}{u^{n_k^i}}$ where u denotes the unit of a single measurement.

Hence the quantities $\sqrt[n_k^i]{p(Z_k^i|Z_{0:k-1})}$ would have the same unit of a single measurement and they can be added safely without a unit inconsistency. As a result, we can consider using the modified predictive likelihood $\sqrt[n_k^i]{p(Z_k^i|Z_{0:k-1})}$ instead of the original predictive likelihood $p(Z_k^i|Z_{0:k-1})$ in the probability calculations to avoid unit inconsistency. With this idea, the probability calculations in the example given above can be written as follows.

$$p(H_0|Z_{0:k}) = \frac{\beta_{FA}(1-P_D P_G)}{\beta_{FA}(1-P_D P_G) + P_D \sqrt[n_k^1]{p(Z_k^1|Z_{0:k-1})} + P_D \sqrt[n_k^2]{p(Z_k^2|Z_{0:k-1})}}, \quad (5.12)$$

$$p(H_1|Z_{0:k}) = \frac{P_D \sqrt[n_k^1]{p(Z_k^1|Z_{0:k-1})}}{\beta_{FA}(1-P_D P_G) + P_D \sqrt[n_k^1]{p(Z_k^1|Z_{0:k-1})} + P_D \sqrt[n_k^2]{p(Z_k^2|Z_{0:k-1})}}, \quad (5.13)$$

$$p(H_2|Z_{0:k}) = \frac{P_D \sqrt[n_k^2]{p(Z_k^2|Z_{0:k-1})}}{\beta_{FA}(1-P_D P_G) + P_D \sqrt[n_k^1]{p(Z_k^1|Z_{0:k-1})} + P_D \sqrt[n_k^2]{p(Z_k^2|Z_{0:k-1})}}. \quad (5.14)$$

5.2.4 Predictive Likelihood Function 3

In order to avoid the unit inconsistency problem of the existing likelihoods and its heuristic fix proposed in the previous subsection, we here propose an alternative heuristic predictive likelihood in this section.

The kinematic part of the predictive likelihood we propose for the extended target is the same as that of Feldmann's and can be written as

$$p_{kin}\left(Z_k^i\middle|\mathrm{Z}_{0:\mathrm{k}-1}\right) = \mathcal{N}(\bar{z}_k^i, \bar{\hat{z}}_k^i, \mathrm{S}_{\mathrm{k|k}-1}), \quad (5.15)$$

where $\bar{\mathrm{z}}_\mathrm{k}$ is the mean of measurements originated from an extended target, $\bar{\hat{z}}_k^i$ is the measurement prediction and $\mathrm{S}_{\mathrm{k|k}-1}$ is defined as in (4.42).

The extension part of the predictive likelihood $\mathrm{p}\left(Z_k^i \middle| \mathrm{Z}_{0:\mathrm{k}-1}\right)$ for the extended target is proposed as

$$p_{ext}\left(Z_k^i \middle| \mathrm{Z}_{0:\mathrm{k}-1}\right) = \mathrm{IW}\left(\bar{Z}_k^i; \mathrm{v}_{\mathrm{k}|\mathrm{k}-1}, \mathrm{X}_{\mathrm{k}|\mathrm{k}-1}\right), \tag{5.16}$$

where $\bar{Z}_k^i$ is measurement spread for the i^{th} cluster defined in (4.5), $\mathrm{v}_{\mathrm{k}|\mathrm{k}-1}$ and $\mathrm{X}_{\mathrm{k}|\mathrm{k}-1}$ are given in (4.35) and (4.36).

The heuristic assumption herein is that the measurement spread and the extension should be close to each other when the measurement noise is small compared to the target extent. The overall predictive likelihood $\mathrm{p}\left(Z_k^i \middle| \mathrm{Z}_{0:\mathrm{k}-1}\right)$ is obtained by multiplying the kinematic part and extension part of the likelihood function as follows.

$$\mathrm{p}\left(Z_k^i \middle| \mathrm{Z}_{0:\mathrm{k}-1}\right) = \mathcal{N}\left(\bar{z}_k^i, \bar{\bar{z}}_k^i, \mathrm{S}_{\mathrm{k}|\mathrm{k}-1}\right)\mathrm{IW}\left(\bar{Z}_k^i; \mathrm{v}_{\mathrm{k}|\mathrm{k}-1}, \mathrm{X}_{\mathrm{k}|\mathrm{k}-1}\right) \tag{5.17}$$

5.2.5 Simulation Results for Predictive Likelihood Functions

In this section, we present some simulation results for the predictive likelihoods we presented above to highlight some properties of these likelihoods. In the simulations our aim is to show the performance of the predictive likelihoods in rewarding the correct sized clusters and punishing wrong sized clusters.

In the simulation, we assume that we are at time k and we have a predicted target posterior density which is defined as

$$\mathrm{p}(\mathrm{x}_\mathrm{k}, \mathrm{X}_\mathrm{k} | \mathrm{Z}_{0:\mathrm{k}-1}) = \mathcal{N}\left(\mathrm{x}_\mathrm{k}; \mathrm{x}_{\mathrm{k}|\mathrm{k}-1}, \mathrm{P}_{\mathrm{k}|\mathrm{k}-1}\right)\mathrm{IW}\left(\mathrm{X}_\mathrm{k}; \mathrm{v}_{\mathrm{k}|\mathrm{k}-1}, \mathrm{X}_{\mathrm{k}|\mathrm{k}-1}\right). \tag{5.18}$$

where the parameters $\mathrm{x}_{\mathrm{k}|\mathrm{k}-1}, \mathrm{P}_{\mathrm{k}|\mathrm{k}-1}, \mathrm{v}_{\mathrm{k}|\mathrm{k}-1}$ and $\mathrm{X}_{\mathrm{k}|\mathrm{k}-1}$ are given as

$\mathrm{x}_{\mathrm{k}|\mathrm{k}-1} = \begin{bmatrix} 0 \\ 20 \\ 0 \\ 20 \end{bmatrix}$, $\mathrm{P}_{\mathrm{k}|\mathrm{k}-1} = \begin{bmatrix} 100 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix}$, $\mathrm{X}_{\mathrm{k}|\mathrm{k}-1} = \begin{bmatrix} 2500 & 0 \\ 0 & 400 \end{bmatrix}$ and $\mathrm{v}_{\mathrm{k}|\mathrm{k}-1} = 25$.

In order measure the performance of the likelihood functions given in the previous sections, we generate random measurements from the scaled versions of the predicted extent $\mathrm{X}_{\mathrm{k|k-1}}$ as follows. We run 100 Monte Carlo runs to obtain the results. In each run, the random measurements are generated as follows. First we scale the predicted extent $\mathrm{X}_{\mathrm{k|k-1}}$ with a scale factor α where α changes between 0.1 to 5.

Uniformly spaced measurements are created in two ways inside the scaled target ellipsoid.

- In the first way, uniformly spaced measurements are created in the scaled ellipsoid around the predicted target position as in Figure 8 where the sensor resolution (the distance between the measurements) of 4 meters is preserved.
- In the second way, the number of measurements inside the target extent is preserved and the sensor resolution is scaled instead as in Figure 9.

After creating the set of uniformly spaced measurements, each point measurement inside the scaled target extent is either randomly kept with probability $\bar{P}_d = 0.8$ or deleted. In this way for each Monte Carlo run we have a random measurement set. Example measurement sets obtained using this method are shown in Figure 10.

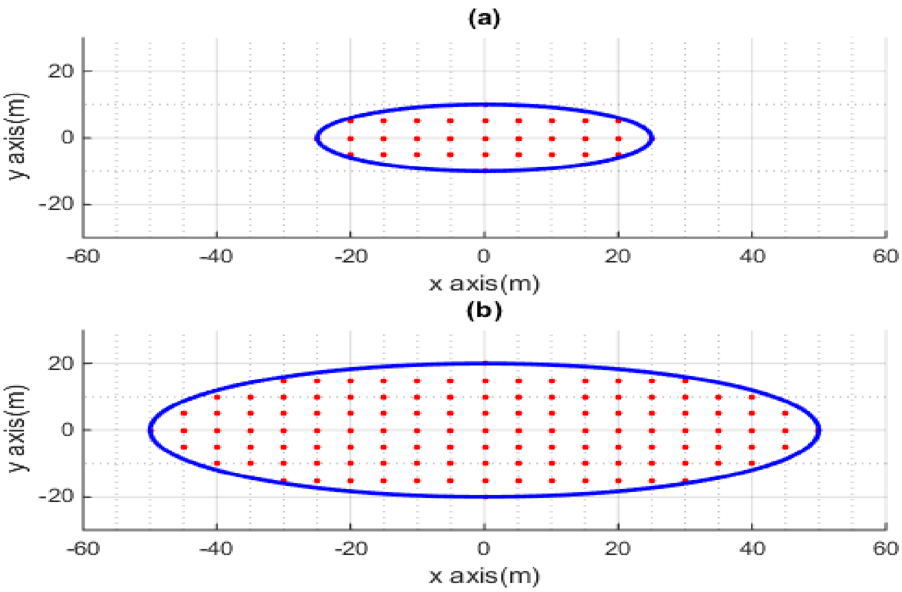

Figure 8: First way of obtaining uniformly spaced set of measurements: The measurements when a) scaling factor is equal to 1 b) scaling factor is equal to 2.

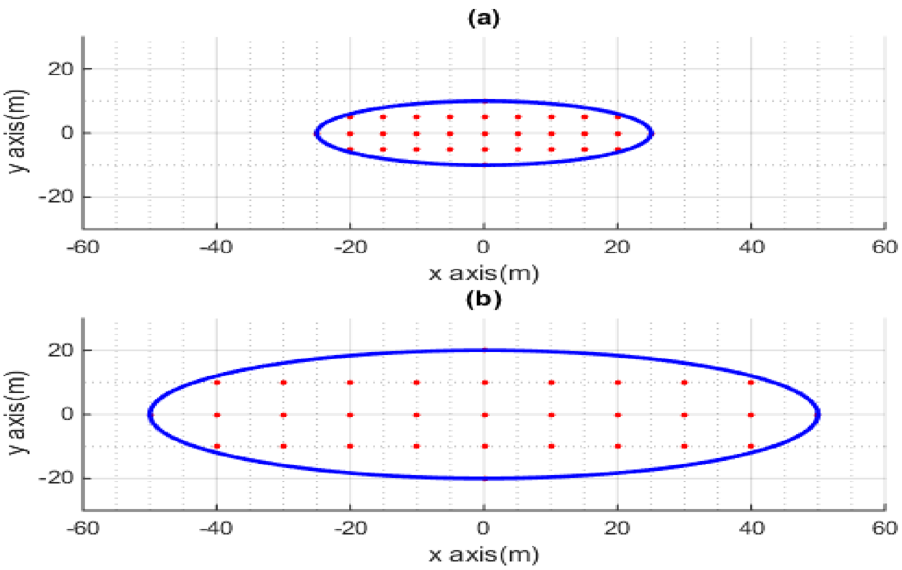

Figure 9: The second way of obtaining uniformly spaced set of measurements: The measurements when a) scaling factor is equal to 1 b) scaling factor is equal to 2.

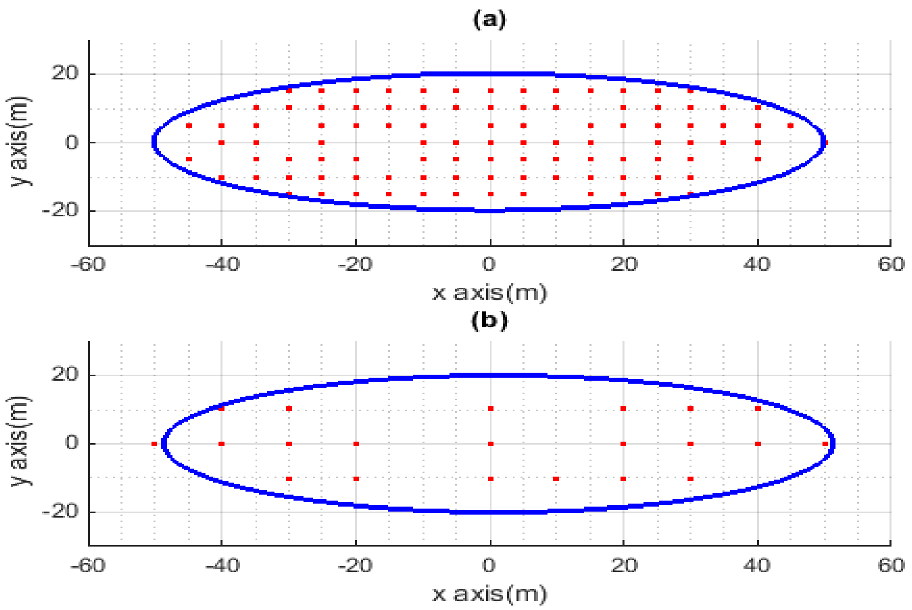

Figure 10: The point measurements inside the scaled target extent are randomly kept with probability $\bar{P}_d = 0.8$.

The average (over the MC runs) of three predictive (log-)likelihood functions calculated using random measurements are given in Figure 11 and Figure 12 respectively. As observed in the figures, the likelihood function of Orguner always decreases with increasing scale factor. This is rather strange because this means that this likelihood function would always reward smaller sized clusters than the predicted target extent. On the other hand, the likelihood of Feldmann is maximized around the scale factor 2 which is rather counter-intuitive because we expect the likelihood to be maximized around the scale factor 1. If one uses Feldmann's likelihood in extended target tracking, this means that the tracker would favor larger sized clusters than the predicted target extent size. It is seen that the proposed likelihood function has the desired property, i.e., it is maximized close to the scale factor unity. These results show that Feldmann's and Orguner's likelihoods might give counter-intuitive results when used in ETT by favoring larger and smaller sized clusters than the predicted target extent size, respectively.

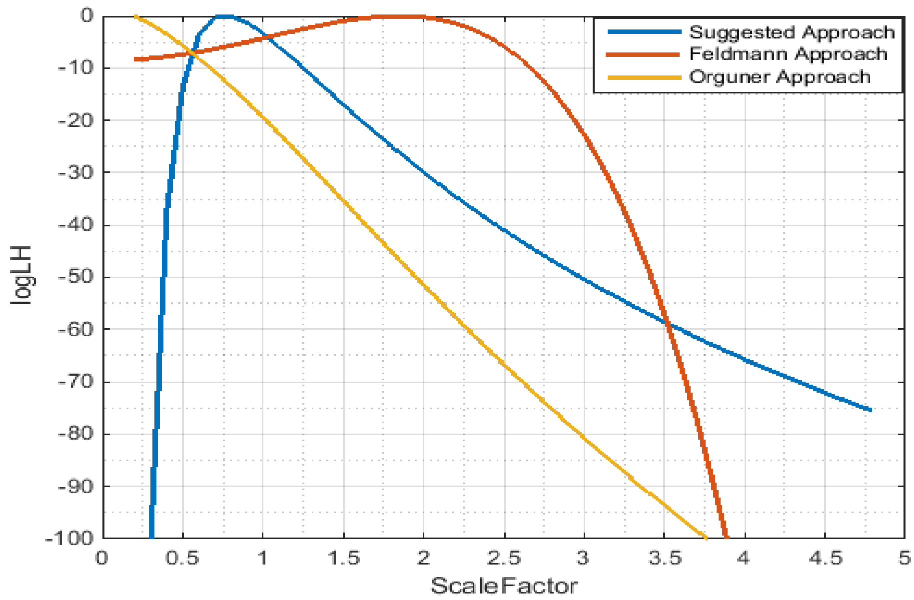

Figure 11: Average log-likelihood functions calculated using the uniformly spaced measurements generated from the scaled target extent (the first way).

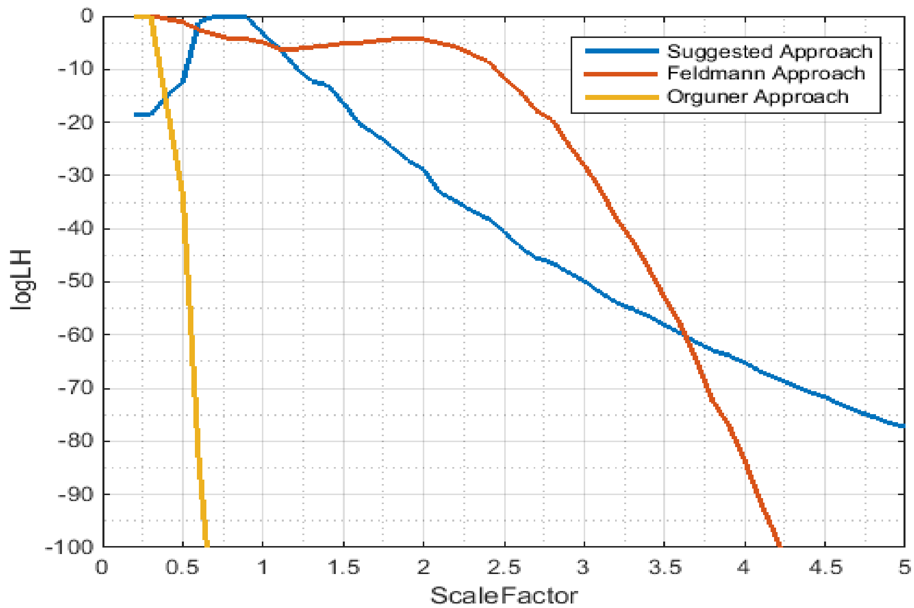

Figure 12: Average log-likelihood functions calculated using the uniformly spaced measurements generated from the scaled target extent (the second way).

In order to understand whether the results given above are dependent on our data generation mechanism, we further investigated the predictive likelihood functions using different data sets. First instead of uniformly spaced measurements, we generated Gaussian distributed measurements with $Z_k \sim \mathcal{N}(Z_k; 0, X_k/4 + R)$ where X_k = diag $((50\ \mathrm{m})^2$, $((20\ \mathrm{m})^2)$. The results are given in Figure 13 for the first way (extent X_k is scaled) and the second way (measurements are scaled after generation with the true extent X_k). In Figure 13, it is seen that the same conclusions above holds for Gaussian distributed measurements as well.

Second, we obtained the results (with uniformly spaced measurements) with different detection probabilities. The detection probabilities were selected as 0.7, 0.6 and 0.5. The results are shown in Figure 14 and Figure 15 for the first and second way of data generation, respectively. In both of the figures, it is seen that the results obtained above are not sensitive to the detection probability used in the simulations.

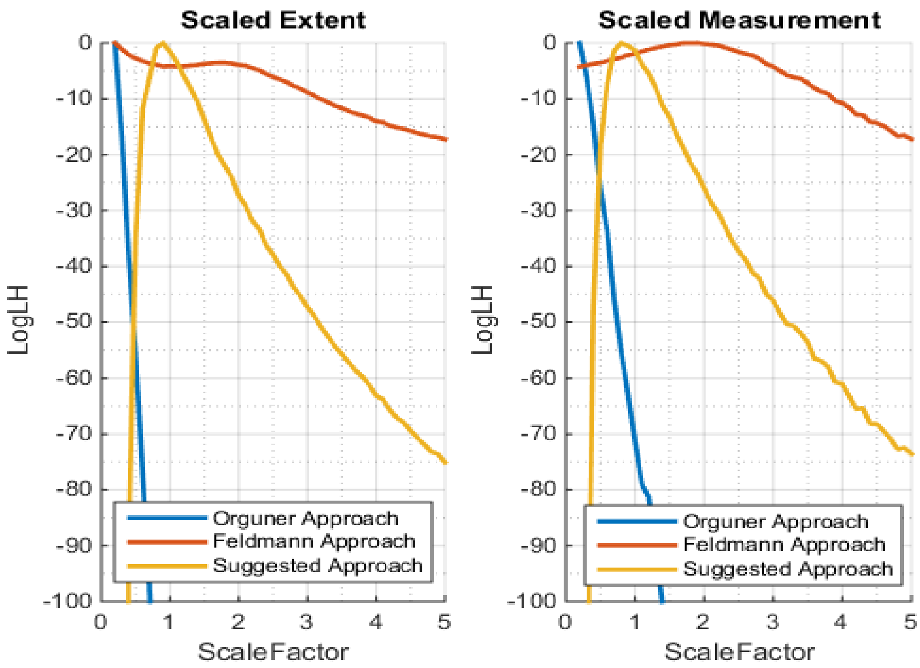

Figure 13: Average log-likelihood functions calculated using the Gaussian distributed measurements.

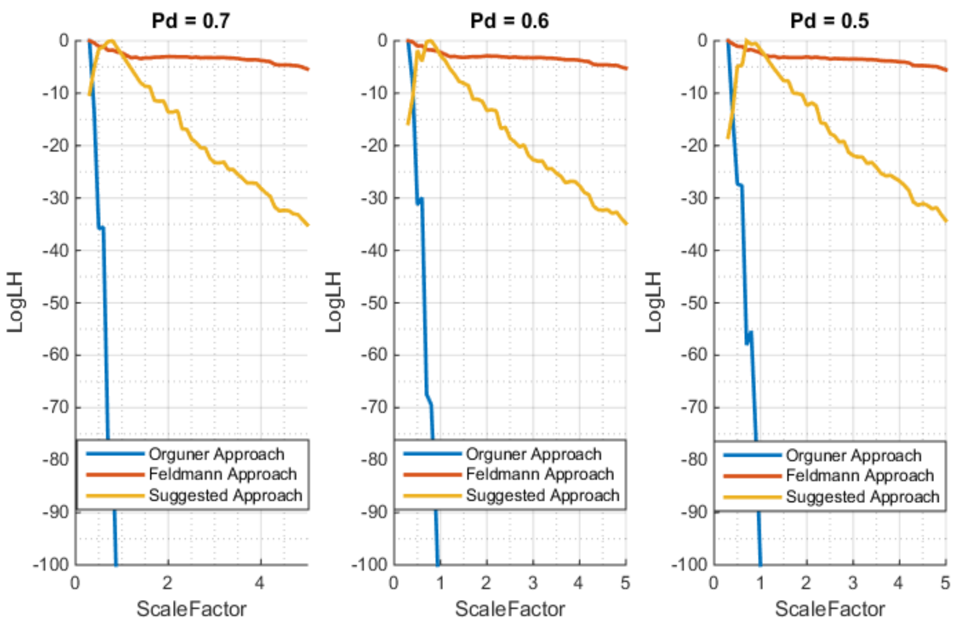

Figure 14: Average log-likelihood functions calculated using the uniformly spaced measurements generated from the scaled target extent (first way) for the detection probabilities 0.7, 0.6 and 0.5.

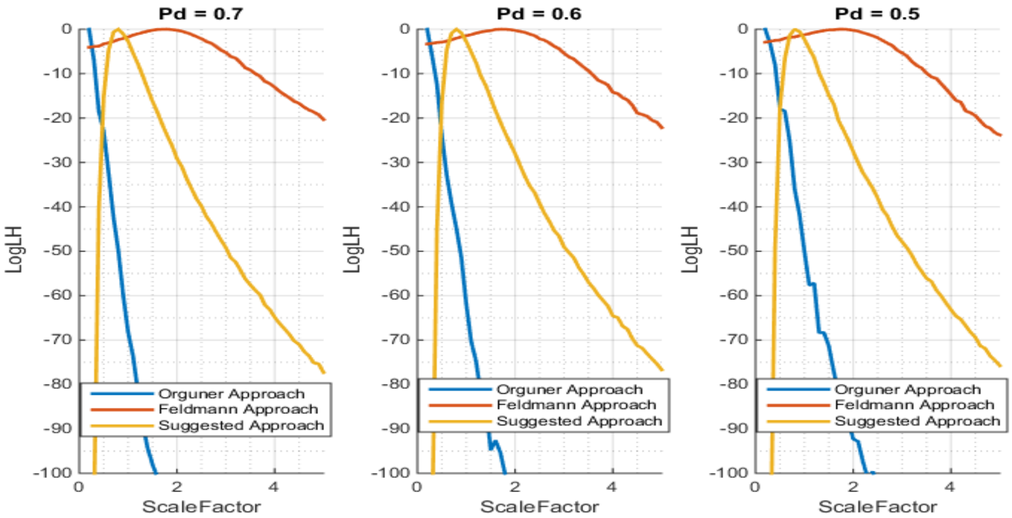

Figure 15: Average log-likelihood functions calculated using the uniformly spaced measurements generated from the scaled target extent (second way) for the detection probabilities 0.7, 0.6 and 0.5.

Finally, we repeated the experiments with different aspect ratios of the object extent. In these trials, all parameters are kept the same except the following:

Aspect Ratio=1: $\mathrm{X}_{k|k-1} = \begin{bmatrix} 400 & 0 \\ 0 & 400 \end{bmatrix}$ & $\mathrm{v}_{k|k-1} = 10,$

Aspect Ratio=3: $\mathrm{X}_{k|k-1} = \begin{bmatrix} 3600 & 0 \\ 0 & 400 \end{bmatrix}$ & $\mathrm{v}_{k|k-1} = 30,$

Aspect Ratio=5: $\mathrm{X}_{k|k-1} = \begin{bmatrix} 10000 & 0 \\ 0 & 400 \end{bmatrix}$ & $\mathrm{v}_{k|k-1} = 50.$

The results are shown for the aspect ratios 1, 3 and 5 in Figure 16 and Figure 17 for the measurements generated using the first way and the second way respectively. In these results we still see that the proposed likelihood is the best but as the aspect ratio gets larger, the average of the proposed likelihood is maximized at a slightly smaller scale than unity. Hence we might predict that the proposed predictive likelihood might have a tendency to choose slightly smaller measurement clusters than the actual target extent as the true target extent gets more and more skewed.

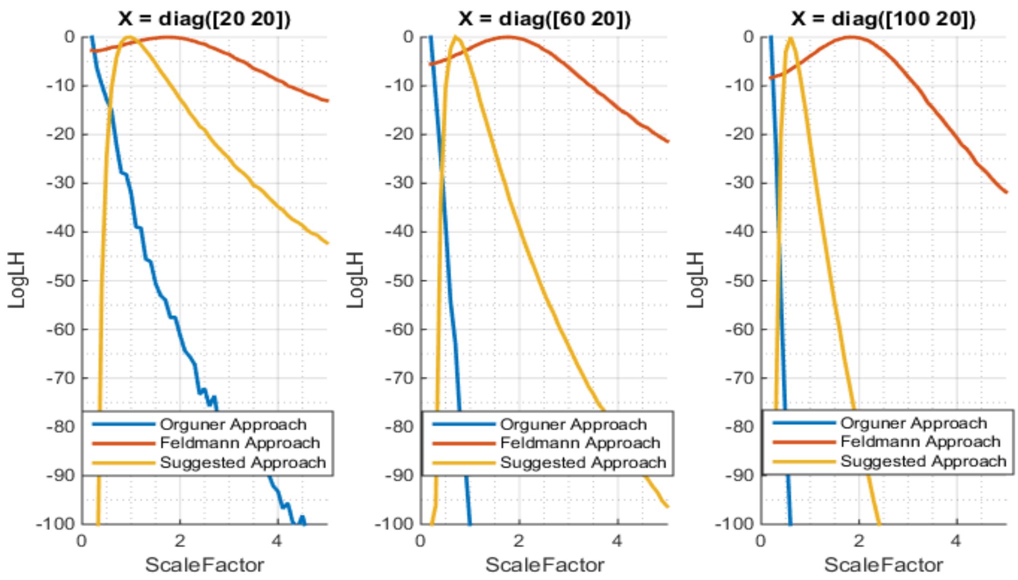

Figure 16: Average log-likelihood functions calculated using the uniformly spaced measurements generated from the scaled target extent (the first way) for different aspect ratios.

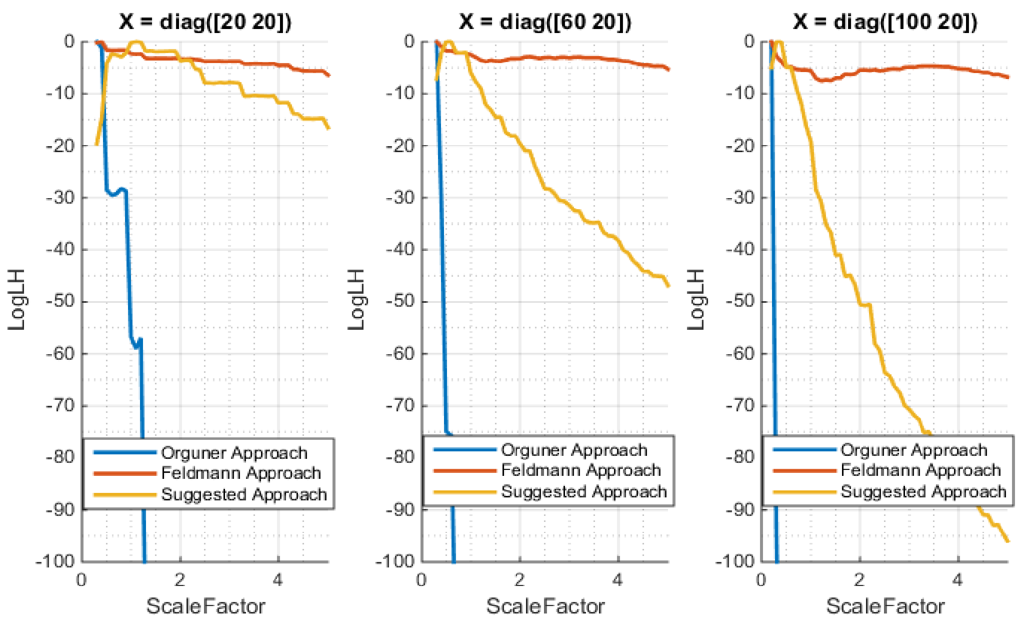

Figure 17: Average log-likelihood functions calculated using the uniformly spaced measurements generated from the scaled target extent (the second way) for different aspect ratios.

5.3 Handling of Close Extended Targets

When IMM-PDA filter uses clustered measurements, errors that are made in the clustering step can propagate into the tracking algorithm. Consider for example, the case when two previously far away targets, which are tracked, get so close to each other that their measurements are erroneously clustered into the same cluster, i.e., the clusters of the individual targets are erroneously merged. In such a case, since there is only a single large merged cluster corresponding to two separate targets, the tracking algorithm will not be able to assign a cluster to one of the targets. In this case, the track for the target which is not assigned with a measurement might be dropped. In this section, in order to avoid such a track loss, we suggest an algorithm which

- detects if two previously tracked targets got closer or not
- determines if the clusters of the targets have been erroneously merged
- if there is erroneous merging, obtains the individual target clusters from the merged cluster using prior extent sizes of the targets.

The details of this algorithm are provided in the following section.

5.3.1 The Algorithm Developed to Track Close Extended Targets

We consider a scenario similar to the one shown in Figure 18. If two targets approach each other as shown in Figure 18, the clustering algorithm can give a single merged cluster for the two targets as shown in Figure 19. To avoid performance degradation in tracking, the erroneous merging should be detected and compensated. The algorithm presented in this section aims to do this. The algorithm is composed of the following steps.

- **Step 1:** detects that two previously tracked targets have approached each other;

- **Step 2:** check if the clustering algorithm merged their clusters in a single cluster or not;
- **Step 3:** separates the merge cluster into individual target clusters using prior information of estimated target extent sizes.

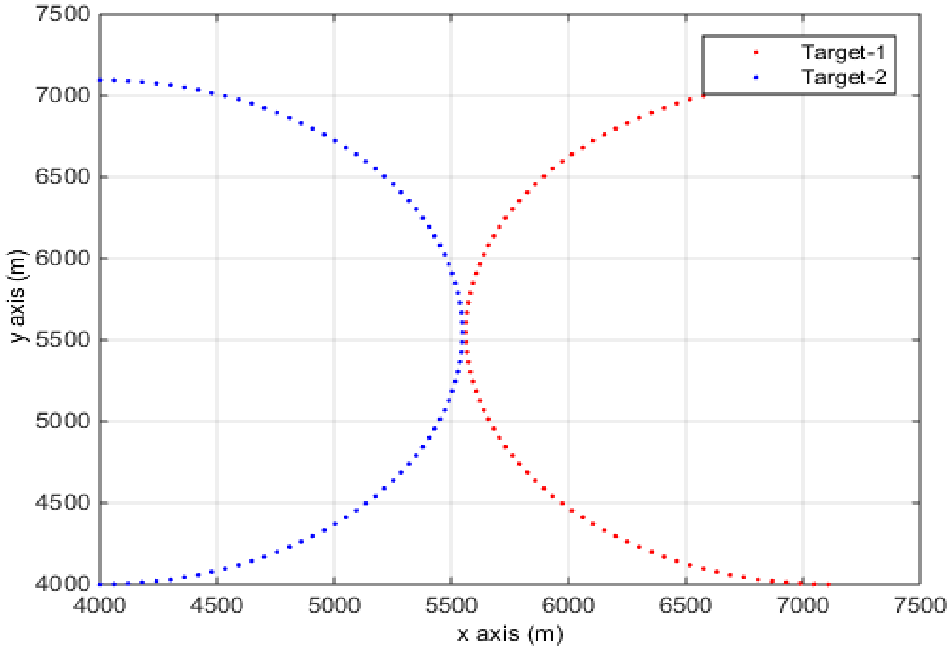

Figure 18: The scenario considered for the algorithm where two targets are approaching each other.

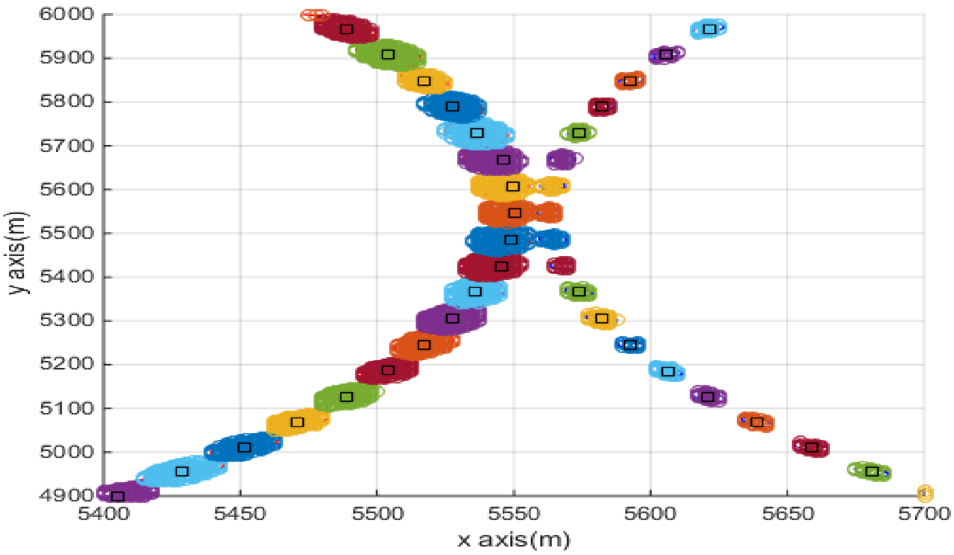

Figure 19:The merged clusters when the targets are close to each other. Notice the same colored ellipsoids (which represent the same cluster) in the center of the figure when targets are close to each other.

Step 1: Determine close targets

In order to check if the extended targets are close to each other, at each time instant the following test is used on the tracked targets.

$$\left(x^1_{k|k-1} - x^2_{k|k-1}\right)^T \left(X^1_{k|k-1} + X^2_{k|k-1}\right)^{-1} \left(x^1_{k|k-1} - x^2_{k|k-1}\right) \lesssim \lambda. \tag{5.19}$$

where the quantities $x^1_{k|k-1}, x^2_{k|k-1}$ and $X^1_{k|k-1}, X^2_{k|k-1}$ denote the predicted kinematic state vectors and predicted extension matrices for the two targets. If the value on the left hand side is smaller than a threshold λ, the targets are declared sufficiently close to cause a merged cluster (in the clustering algorithm) and further processing is started.

Step 2: Checking for a Merged Cluster

In the case that two tracked targets are declared to be sufficiently close to cause a merged cluster (Step-1), the algorithm checks if there is only one cluster in the unions of the gates of the two targets or not. If there is a single cluster in the unions of the gates of the two targets then the clustering algorithm is declared to have made the erroneous merging of the clusters of the two targets.

Step 3: Re-clustering of the Merged Cluster to Recover the Individual Clusters of the Targets

If there is one clustered measurement in the unions of the gates in Step 2, the individual measurements in the merged cluster are re-clustered using the predicted extent sizes of the targets using a version of the EM algorithm where the cluster covariances are fixed at the predicted target extent sizes.

The measurements that the algorithm tries to re-cluster are shown as $\{z_k^i\}_{i=1}^m$. The recursion index is shown as n. The cluster weights and means at the recursion $(n-1)$ are shown as $\{w_j^{n-1},\}_{j=1}^2$ and $\{\mu_j^{n-1}\}_{j=1}^2$. The n^{th} step of the recursive algorithm is given as follows,

- Responsibility computation

$$\gamma_{ij} = \frac{w_j^{n-1} N\left(z_k^i; \mu_j^{n-1}, X_{k|k-1}^j\right)}{\sum_{j=1}^2 w_j^{n-1} N\left(z_k^i; \mu_j^{n-1}, X_{k|k-1}^j\right)}, \tag{5.20}$$

for $j = 1,2$ and $i = 1, \ldots, m$.

- Updated weight computation

$$w_j^n = \frac{\sum_{i=1}^m \gamma_{ij}}{\sum_{j=1}^2 \sum_{i=1}^m \gamma_{ij}}. \tag{5.21}$$

- Updated mean computation

$$\mu_j^n = \frac{\sum_{i=1}^m \gamma_{ij}}{\sum_{j=1}^2 \sum_{i=1}^m \gamma_{ij}} z_k^i. \tag{5.22}$$

- Stopping criterion

$$\sqrt{\sum_{j=1}^{2} \left\| \mu_j^n - \mu_j^{n-1} \right\|^2} \leqslant \lambda. \tag{5.23}$$

- If the left hand side of (5.23) is smaller than threshold λ, it is assumed that the algorithm has converged. So, each measurement y_k^i is assigned to the target whose index j_i by using the last calculated responsibilities $\{\gamma_{ij} | \ j = 1{,}2 \quad i = 1, \ldots, m\}$.

$$j_i = \text{arg max}_j \gamma_{ij.} \tag{5.24}$$

5.4 Pseudo Code of One Step of the Overall Extended Target Tracker

In this section, a pseudo code of one step of the overall extended tracking algorithm proposed in this thesis is given.

Algorithm 2: One Step of Overall Algorithm

$\mathrm{m_k}$	:	Number of total measurements
Ω	:	Validation matrix by $\mathrm{m_k} x \, \mathrm{m_k}$ denoting the closeness of measurements
$\mathrm{n_k}$	:	number of the measurements in a cluster
$\{\tilde{\mathrm{z}}_\mathrm{k}^\mathrm{i}\}_{\mathrm{i}=1}^{\mathrm{n_k}}$	:	the set of $\mathrm{n_k}$ two dimensional clustered position measurements at each time k
$\{\mathrm{z}_\mathrm{k}^\mathrm{i}\}_{\mathrm{i}=1}^{\mathrm{n_k}}$	:	Re-clustered measurements by considering the track extension
$\bar{z}_\mathrm{k}^\mathrm{i}$	:	Mean of the measurements in cluster i at time k
$\hat{z}_{\mathrm{k\|k-1}}^\mathrm{j}$	:	Predicted measurement for target j
$\mathrm{S}_{\mathrm{k\|k-1}}^\mathrm{j}$	:	Predicted measurement covariance matrix for target j
$\mathrm{P}_{\mathrm{k\|k-1}}^{\mathrm{j,l}}$	:	Predicted state covariance matrix for target j and model l
$\hat{z}_{\mathrm{k\|k-1}}^{\mathrm{j,l}}$	:	Predicted measurement for target j, model l

Algorithm 2 (continued)

$S\_ext_{k\|k-1}^{j,l}$	:	Predicted measurement covariance matrix considering the extension for target j
$v_{k\|k-1}^{j}$	:	Predicted number of the measurements in a cluster for target j
$X_{k\|k-1}^{j,l}$	:	Predicted extension for target j, model l
$x_{k\|k}^{j,l,i}$	:	Estimated state vector for target j, model l, measurement i
$P_{k\|k}^{j,l,i}$	:	Estimated state covariance matrix for target j, model l, measurement i
$v_{k\|k}^{j}$	:	Estimated number of the measurements in a cluster for target j
$X_{k\|k}^{j,l,i}$	:	Estimated extension for target j, model l, measurement i
$\lambda_{k\|k-1}^{j,l,i}$	:	Predicted likelihood function for target j, model l, measurement i
$\pi_{k\|k}^{j,l}$	:	Transition probability matrix
$\mu_{k\|k}^{j,l}$	:	IMM model probability
TS	:	Track status denoting tentative track, confirmed track or deleted track

Set size of extension as $\begin{bmatrix} 2500 & 0 \\ 0 & 400 \end{bmatrix}$ for big target and scale it with scale factor 0.2 for small target.

Set parameters as

- P_g=0.99, P_{FA}=10^{-5}, P_D=0.8

P_g: Gating Probability, P_{FA}:False Alarm Probability, P_D:Detection Probability

Algorithm 2 (continued)

ALGORITHMS

1- $\{\tilde{z}_k^i\}_{i=1}^{n_k}$ = Clustering(Ω)

2- $[\bar{z}_k^i, \{z_k^i\}_{i=1}^{n_k}]$ = Re-cluster $(x_{k|k-1}^{j=1}, x_{k|k-1}^{j=2}, P_{k|k-1}^{j=2}, P_{k|k-1}^{j=1}, X_{k|k-1}^{j=1}, X_{k|k-1}^{j=2}, \{\tilde{z}_k^i\}_{i=1}^{n_k})$

3- $\bar{z}_{k_{gated}}^{i,j}$ = Gating$(\bar{z}_k^i, \hat{\bar{z}}_{k|k-1}^j, S_{k|k-1}^j)$

4- $[x_{k|k}^{j,l,i}, P_{k|k}^{j,l,i}, v_{k|k}^j, X_{k|k}^{j,l,i}, \lambda_{k|k-1}^{j,l,i}]$ = Measurement Update $(x_{k|k-1}^{j,l}, P_{k|k-1}^{j,l}, v_{k|k-1}^j, X_{k|k-1}^{j,l}, \bar{z}_{k_{gated}}^{i,j}, \hat{\bar{z}}_{k|k-1}^{j,l})$

5- $[x_{k|k}^{j,l}, P_{k|k}^{j,l}, v_{k|k}^j, X_{k|k}^{j,l}, \lambda_{k|k-1}^{j,l}]$ = Data Association $(x_{k|k}^{j,l,i}, P_{k|k}^{j,l,i}, v_{k|k}^j, X_{k|k}^{j,l,i}, \lambda_{k|k-1}^{j,l,i})$

6- $[x_{k|k}^{j,0l}, P_{k|k}^{j,0l}, X_{k|k}^{j,0l}, \mu_{k|k-1}^{j,l}]$ = IMM Mixing $(x_{k|k}^{j,l}, P_{k|k}^{j,l}, v_{k|k}^j, X_{k|k}^{j,l}, \lambda_{k|k-1}^{j,l}, \pi_{k-1|k-1}^{j,l})$

7- $[x_{k+1|k}^{j,l}, P_{k+1|k}^{j,l}, v_{k+1|k}^j, X_{k+1|k}^{j,l}, \hat{\bar{z}}_{k+1|k}^{j,l}, S_{k+1|k}^{j,l}]$ = Prediction Update$(x_{k|k}^{j,0l}, P_{k|k}^{j,0l}, v_{k|k}^j, X_{k|k}^{j,0l})$

8- $[x_{k|k}^j, P_{k|k}^j, X_{k|k}^j, \hat{\bar{z}}_{k+1|k}^j, S_{k+1|k}^j]$ = Combination IMM Models $(x_{k|k}^{j,l}, P_{k|k}^{j,l}, X_{k|k}^{j,l}, \hat{\bar{z}}_{k+1|k}^{j,l}, S_{k+1|k}^{j,l}, \mu_{k|k-1}^{j,l})$

9- TS = Track Maintenance $(\bar{z}_{k_{gated}}^{i,j})$

10- $\bar{z}_{k_{rest}}^i$ = Select the Remaining Measurements $(\bar{z}_{k_{gated}}^{i,j}, \bar{z}_k^i)$

$x_{1|1}, P_{1|1}, v_{1|1}, X_{1|1}$ = Initiate New Track $(\bar{z}_{k_{rest}}^i, \bar{z}_{k-1_{rest}}^i)$

The details of the steps in Algorithm 2 are given below. The details of the clustering step was already given in Algorithm 1.

Algorithm 3: Re-clustering Algorithm

- **<u>Determine the measurements of two close targets</u>**

  <u>Input :</u> $x^1_{k|k-1}, x^2_{k|k-1}, X^1_{k|k-1}, X^2_{k|k-1}, P^1_{k|k-1}, P^2_{k|k-1}, \{z^i_k\}_{i=1}^{n_k}$

  <u>Output:</u> $\{z^i_k\}_{i=1}^{n_k}$

Compute the distance between two targets

IF the targets are close enough

- Compute threshold which equals to 3 times of predicted measurement covariance matrix
- Compute the distance between centroid of measurements in the cluster and the targets
- Select the measurements exceeding the threshold
- Obtain two clusters from the remaining measurements under assumption that the measurements are normally distributed.

END IF

Algorithm 4: Gating Algorithm

- **<u>Gate the measurements</u>**

  <u>Input :</u> $\bar{z}^i_k, \hat{\bar{z}}^j_{k|k-1}, S^j_{k|k-1}$

  <u>Output:</u> $\bar{z}^i_{k_{gated}}$

- **FOR** target j
  - **FOR** clustered measurements i
    - Gate measurements by ellipse gate
  - **END FOR**
- **END FOR**

Algorithm 5: Measurement Update Algorithm

- **Measurement Update**

  Input : $x^{j,l}_{k|k-1}$, $P^{j,l}_{k|k-1}$, $v^{j}_{k|k-1}$, $X^{j,l}_{k|k-1}$, $\bar{z}^{i,j}_{k_{gated}}$, $\hat{\bar{z}}^{j,l}_{k|k-1}$

  Output: $x^{j,l,i}_{k|k}$, $P^{j,l,i}_{k|k}$, $v^{j}_{k|k}$, $X^{j,l,i}_{k|k}$, $\lambda^{j,l,i}_{k|k-1}$

  - **FOR** target j
    - **FOR** IMM filter model l
      - ❖ **FOR** gated and clustered measurement i
      - \- Update the kinematic state vector
      - \- Update the kinematic state covariance matrix
      - \- Update the extension matrix
      - \- Update the extension parameter
      - \- Compute likelihood
      - ❖ **END FOR**
    - **END FOR**
  - **END FOR**

Algorithm 6: Data Association Algorithm

- **Data Association**

  Input : $x^{j,l,i}_{k|k}$, $P^{j,l,i}_{k|k}$, $v^{j}_{k|k}$, $X^{j,l,i}_{k|k}$, $\lambda^{j,l,i}_{k|k-1}$

  Output: $x^{j,l}_{k|k}$, $P^{j,l}_{k|k}$, $v^{j}_{k|k}$, $X^{j,l}_{k|k}$, $\lambda^{j,l}_{k|k-1}$]

  - **FOR** target j
    - **FOR** IMM filter model l
      - ❖ **FOR** gated and clustered measurement i
      - \- Calculate association probabilities

Algorithm 6 (continued)

- Weight estimates with association probabilities
  - ❖ **END FOR**
- ▪ **END FOR**
- ○ **END FOR**

Algorithm 7: IMM Mixing Algorithm

- ○ **IMM Mixing**

Input : $x^{j,l}_{k|k}$, $P^{j,l}_{k|k}$, $v^{j}_{k|k}$, $X^{j,l}_{k|k}$, $\lambda^{j,l}_{k|k-1}$, $\pi^{j,l}_{k-1|k-1}$

Output: $x^{j,0l}_{k|k}$, $P^{j,0l}_{k|k}$, $X^{j,0l}_{k|k}$, $\mu^{j,l}_{k|k-1}$

- ○ **FOR** target j
  - Compute the mixing probabilities
  - Mix estimates
  - Compute the IMM model probabilities
- ○ **END FOR**

Algorithm 8: Prediction Update Algorithm

- **<u>Prediction Update</u>**

  <u>Input :</u> $x^{j,0l}_{k|k}$, $P^{j,0l}_{k|k}$, $v^{j}_{k|k}$, $X^{j,0l}_{k|k}$

  <u>Output:</u> $x^{j,l}_{k+1|k}$, $P^{j,l}_{k+1|k}$, $v^{j}_{k+1|k}$, $X^{j,l}_{k+1|k}$, $\hat{\hat{z}}^{j,l}_{k+1|k}$, $S^{j,l}_{k+1|k}$

  - **FOR** target j
    - **FOR** IMM filter model l
    - Predict the state vector related to kinematics
    - Predict the state covariance matrix related to kinematics
    - Predict the extension matrix
    - Predict the extension parameter
    - Predict the measurement position
    - Predict the measurement covariance
    - **END FOR**
  - **END FOR**

Algorithm 9: Combination of IMM Models

- **Combination of IMM Models**

  <u>Input :</u> $x^{j,l}_{k|k}$, $P^{j,l}_{k|k}$, $X^{j,l}_{k|k}$, $\hat{\hat{z}}^{j,l}_{k+1|k}$, $S^{j,l}_{k+1|k}$, $\mu^{j,l}_{k|k-1}$,

  <u>Output:</u> $x^{j}_{k|k}$, $P^{j}_{k|k}$, $X^{j}_{k|k}$, $\hat{\hat{z}}^{j}_{k+1|k}$, $S^{j}_{k+1|k}$

  - **FOR** target j
    - Combine the kinematic state vectors
    - Combine the kinematic state covariance matrices
    - Combine the extension matrices
    - Combine the predicted measurements
    - Combine the predicted measurement covariances
  - **END FOR**

Algorithm 10:Track Maintenance Algorithm

- **Tracking Maintenance**

  Input : $\bar{z}^{i,j}_{k_{gated}}$

  Output: TS

  - **FOR** target j
    - M/N logic based track life
    - Confirmed track deletion
  - **END FOR**

Algorithm 11: Selecting the Remaining Measurements

- **Selection of Rest Measurements for Updating Track**

  Input : $\bar{z}^{i,j}_{k_{gated}}$, $\bar{z}^{i}_{k}$

  Output: $\bar{z}^{i}_{k_{rest}}$

  - **FOR** target j
    - Delete gated measurements in the measurement list
  - **END FOR**

Algorithm 12: Track Initiation Algorithm

- **Initiate Track**

  Input : $\bar{z}^{i}_{k_{rest}}, \bar{z}^{i}_{k-1_{rest}}$

  Output: $x_{1|1}, P_{1|1}, v_{1|1}, X_{1|1}$

All the rest after initiation are held for two points initiation.

$$\varsigma_{1|1} = \bar{z}^{i}_{k_{rest}}$$

Algorithm 12(continued)

$$\dot{\varsigma}_{1|1} = \frac{\bar{z}_1 - \bar{z}_0}{T}$$

$$x_{1|1} = \begin{bmatrix} \varsigma_{1|1} \\ \dot{\varsigma}_{1|1} \end{bmatrix}$$

$$P_{1|1} = \begin{bmatrix} R & R/T \\ R/T & 2R/T^2 \end{bmatrix}$$

$$X_{1|1} = \operatorname{cov}\left(\{z_k^i\}_{i=1}^{n_k}\right)$$

$$v_{1|1} = n_k$$

CHAPTER 6

SIMULATION STUDIES

In this chapter, simulation studies conducted with the proposed algorithms are presented. The chapter is divided into two parts. The first part presents simulation results for the IMM-PDA filter for extended targets and the second part gives results of the IMM-PDA filter for extended targets with close targets.

6.1 Simulation Results for the IMM-PDA Filter for Extended Targets

In this section, we first give information about the simulation environment we use and then present the simulation results.

6.2 Simulation Environment

We will describe the simulation environment in three parts:

1- Measurement Generation
2- Target Tracking
3- Performance Measures

6.2.1 Measurement Generation

We intend to generate measurements which are close to real radar data. An example real radar data from an extended target is shown in Figure 20. We see that the measurements are spread over the target extent with uniform separation along the range vector and the cross-range vector. In our simulations we generate the extended target measurements similar to the real data shown in Figure 20.

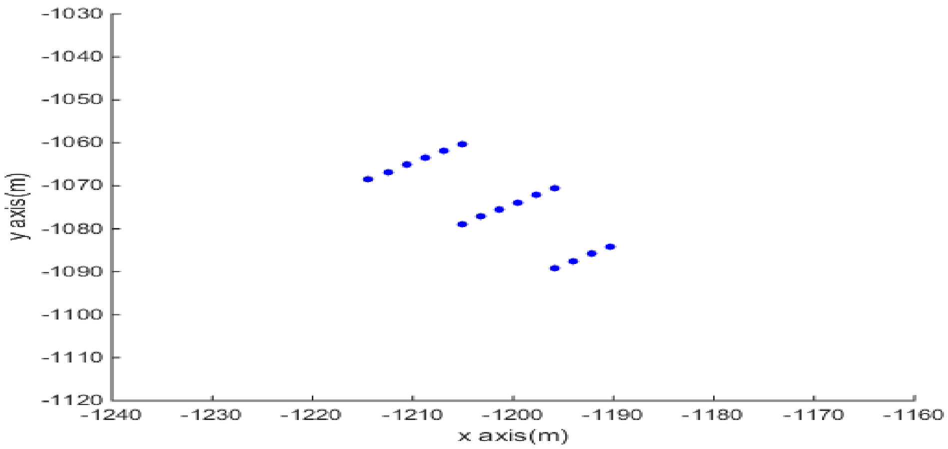

Figure 20: Real Radar Data.

In the simulations, we consider a random scenario a realization of which is shown in Figure 21, where a fighter aircraft first gets away from the radar with constant velocity between t_0 and t_1 which are 1^{th} second and 28^{th} second; then launches a missile which is also to be tracked with the IMM-PDA filter and makes a coordinated turn with acceleration of 6.7 g between t_1 and t_2 which are 28^{th} second and 39^{th} second after the missile launch. Missile goes straight between t_1 and t_3 which are 28^{th} second and 56^{th} second. The missile launch operation presents a challenge for tracking since it contains a spawned target (missile) which is smaller in extent than the original target (fighter aircraft). The measurements for the fighter aircraft and the missile are generated using the following parameters:

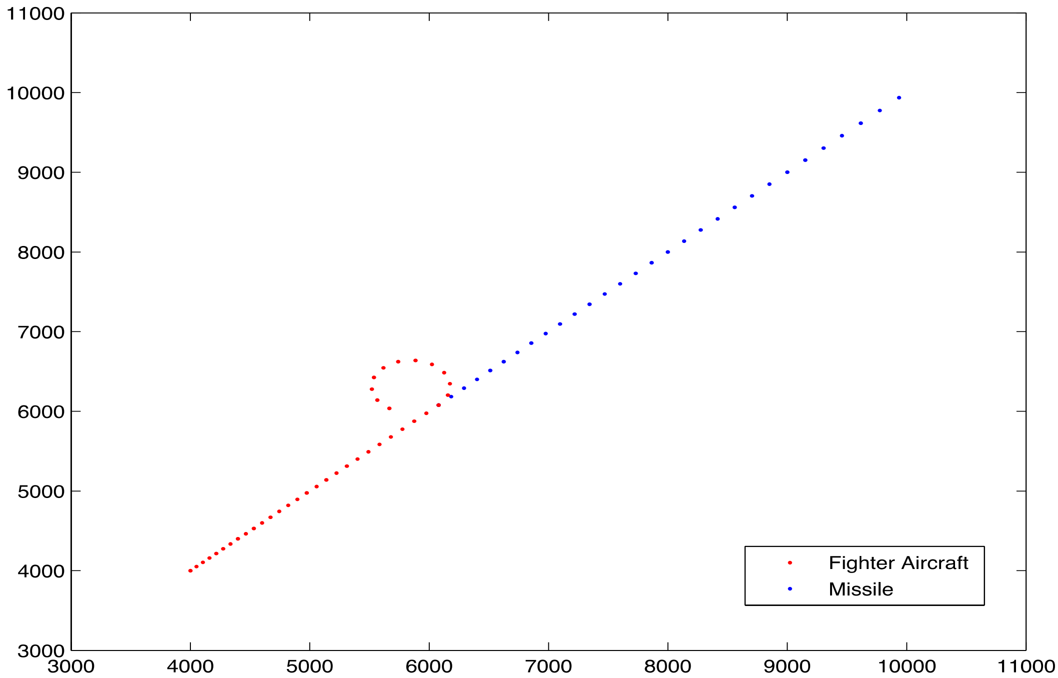

Figure 21: One realization of the random target tracking scenario used in the simulations.

- The sampling time in the scenario is selected as 1 second.
- The lengths of the semi major and minor axes of the fighter aircraft are selected as 25 and 10 meters respectively. For each Monte Carlo run, a variation that is normally distributed with zero-mean and standard deviation of 2.5 and 1 meters (i.e., 10% of the true size), respectively is added to these numbers.
- The lengths of the semi major and minor axes of the missile are selected as 5 and 2 meters respectively. For each Monte Carlo run, a variation that is normally distributed with zero-mean and standard deviation of 0.5 and 0.2 meters (i.e., 10% of the true size), respectively is added to these numbers.
- The fighter aircraft moves with constant velocity for a random number of samples before launching the missile for each run in the Monte Carlo run. The corresponding random number is obtained by sampling from the

normal distribution with mean 25 and standard deviation 10 and then rounding to the nearest integer.

- The fighter aircraft maneuvers with a coordinated turn for a random number of samples after launching the missile for each Monte Carlo run. The corresponding random number is obtained by sampling from the normal distribution with mean 10 and standard deviation 5 and then rounding to the nearest integer.
- The missile moves with constant velocity for a random number of samples for each Monte Carlo run. The corresponding random number is obtained by sampling from normal distribution with mean 25 and standard deviation 10 and then rounding to the nearest integer.
- The measurements for each target are generated inside the target ellipsoids with 2.5 meters uniform separation. After generating measurements, some of them are removed randomly with detection probability 0.88.

The measurements for the false alarm process are generated using the following parameters.

- The number of false measurements for each Monte Carlo run is determined by Poisson distribution with the mean $\mu = \beta_{FA}V$ which is where p_{FA} is false alarm rate and V is volume of the surveillance region the radar covers [33]. We use β_{FA} as 10^{-6} per unit area per scan.
- The size of false measurements changes in each scan and each Monte Carlo run. The change is normal distributed with zero mean and standard deviation 10.
- Quantization step is 2.5 meters.

When the measurements are generated, they are clustered using the clustering algorithm presented in Section 5.1.

In Figure 22, all clustered measurements for a single Monte Carlo run are shown. False measurements which are represented by circles in different colors are also separated. The measurements which belong to targets are represented by dark blue circles. In Figure 23, a zoomed version of Figure 22 illustrates the individual measurements in a cluster for the fighter aircraft. The blank spots in the fighter aircraft cluster are caused by the non-unity detection probability which results in the elimination of some measurements.

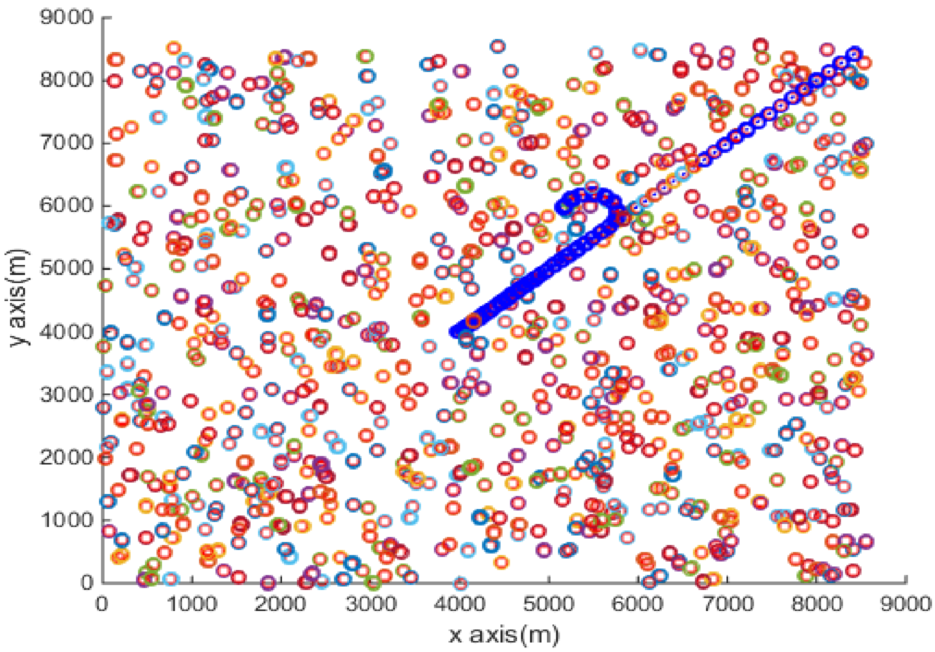

Figure 22: All clustered measurements in a single Monte Carlo run.

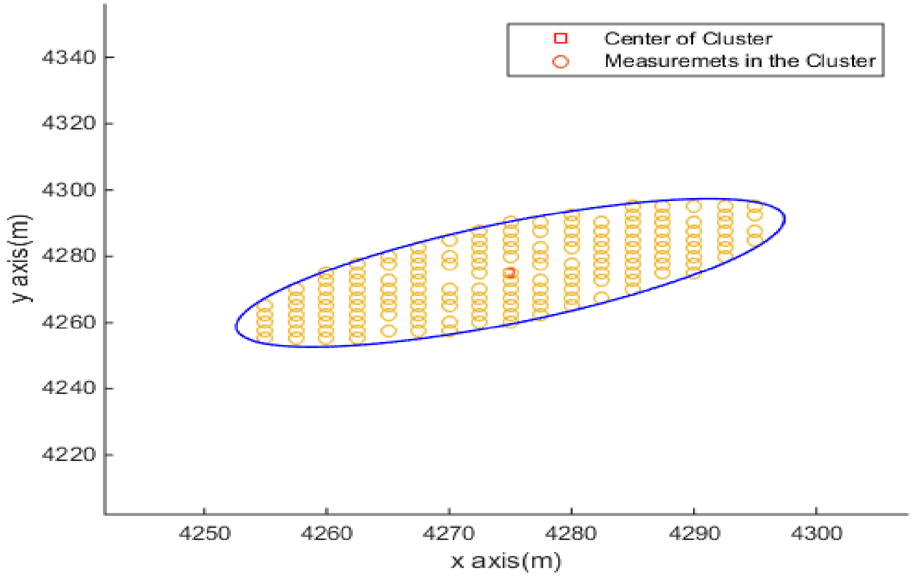

Figure 23: The measurements in a cluster belonging to the fighter aircraft.

6.2.2 Target Tracking

In this sub-section, we give the related parameters of the overall target tracking algorithm we used in the simulations. The target tracking algorithm uses M/N logic with ellipsoidal gating (with gating probability 0.99) to decide to delete, keep or confirm a track. The parameters for the M/N logic are 2/2 & 2/3. In other words, for initiating a target we need at least two consecutive measurements (2/2) and in order to confirm a track we need 2 measurements in the subsequent 3 sampling times (2/3). Once we get two consecutive measurements, we initiate a Kalman filter from these two measurements using two-point initiation. The kinematic state vector x_k we use has two components, 2-D position ς and 2-D velocity $\dot{\varsigma}$, which are initiated as follows.

$$\varsigma_{1|1} == \bar{z}_1, \tag{6.1}$$

$$\dot{\varsigma}_{1|1} = \frac{\bar{z}_1 - \bar{z}_0}{T}, \tag{6.2}$$

where $\bar{z}_1$ and $\bar{z}_0$ denote the cluster centers obtained from the two measurements and T is the sampling period.

The corresponding kinematic state covariance matrix is set to be

$$P_{1|1} = \begin{bmatrix} R & R/T \\ R/T & 2R/T^2 \end{bmatrix}, \tag{6.3}$$

where we select the measurement covariance as R = $\begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}$. The initial extent state X_k and extension parameter v_k are set as

$$X_{1|1} = \operatorname{cov}\left(\{z_1^i\}_{i=1}^{n_1}\right), \tag{6.4}$$

$$v_{1|1} = n_1, \tag{6.5}$$

where z_1^i denote the ith measurement in the second measured cluster and n_1 is the number of measurements in the cluster.

The IMM-PDA filter as described in the pseudo-code in Section 5.4 is implemented with 3 different likelihood functions given in Section 5.2 and their performances are compared to each other.

6.2.3 Performance Measures

In order to measure the performance of the tracking algorithm with different likelihoods, we make 500 Monte Carlo (MC) runs for each method. Root Mean (over MC runs) Square (RMS) errors are calculated for the kinematic and extent states as follows [34].

$$E_x \triangleq \sqrt{\frac{1}{N}\sum_{j=1}^{N}\left\| x_{k|k}^j - x_k^{true}\right\|^2}, \tag{6.6}$$

$$E_X \triangleq \left[\frac{1}{4N}\sum_{j=1}^{N} tr\left(\left(X_{k|k}^j - X_k^{true}\right)^2\right)\right]^{\frac{1}{4}}. \tag{6.7}$$

The RMS errors are then averaged over time to obtain the average (over time) RMS errors.

In addition to the kinematic errors, data association errors are counted in the MC runs. In the IMM-PDA filter, for each track, the cluster with the highest posterior hypothesis probability in the gate is assumed to be assigned to the track. If a track is assigned with a cluster that does not belong to the corresponding target at least for a single time instant in a MC run, the tracker is deemed to have made an association error in that MC run. The performance of the trackers is also evaluated based on the percentage of the MC runs in which they made no association error. The larger this percentage is, the better the performance of the corresponding tracker is.

6.3 Simulation Results

In this subsection, we present the results obtained by the trackers using different likelihoods by illustrating the typical results for their MC runs. The results for trackers with the three different likelihood functions are given in three separate sections. The comparisons of the results obtained by the MC runs are made in the fourth subsection.

In the figures in this subsection, the following notations are used.

- Numbers 1 and 2 denote the track ID
- The black squares denote the position estimates of the targets
- The red points denote the measurements
- The green ellipses denote the estimated extension of the targets
- The pink ellipses denote the real extension of the fighter aircraft.
- The blue ellipses denote the real extension of the missile.

6.3.1 The Results of IMM-PDA filter with Feldmann's Likelihood

A typical run of the IMM-PDA filter using the Feldmann's likelihood is shown in Figure 24. In Section 5.2.5, it was seen that Feldmann's likelihood function gives unexpected results by assigning higher likelihoods to larger extended targets than the predicted target size. On the other hand, Feldmann's likelihood gives similar results for smaller targets than the predicted target size. In the scenario considered in the thesis, when a small target (i.e., the missile) is spawned from the fighter aircraft, the Feldmann's likelihood does not punish the missile measurement sufficiently. Hence the extension part of the likelihoods for the missile and fighter aircraft measurements become similar and the kinematic part of the likelihoods dominates. Due to this, the tracker confuses the fighter aircraft with the missile as seen in Figure 24.

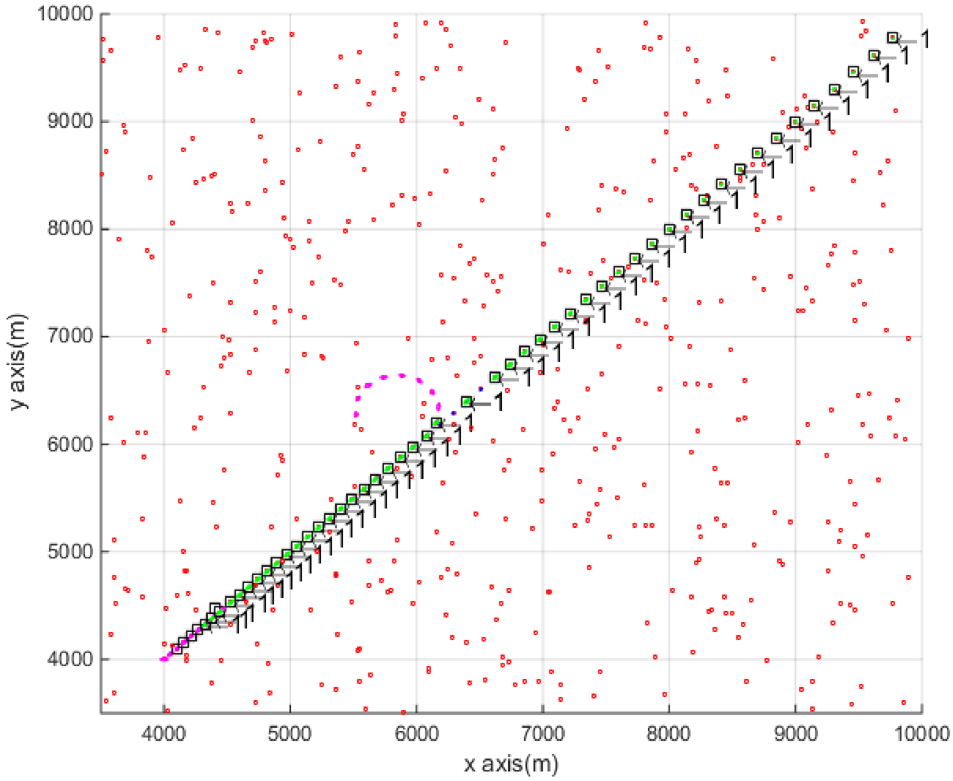

Figure 24: The estimations of IMM-PDA filter with Feldmann's likelihood.

In Figure 25, which shows the zoomed view of Figure 24, it is seen that the fighter aircraft continues to use missile measurements. The tracker actually tries to initiate a new target for the fighter measurements but the initiation algorithm fails due to the maneuver.

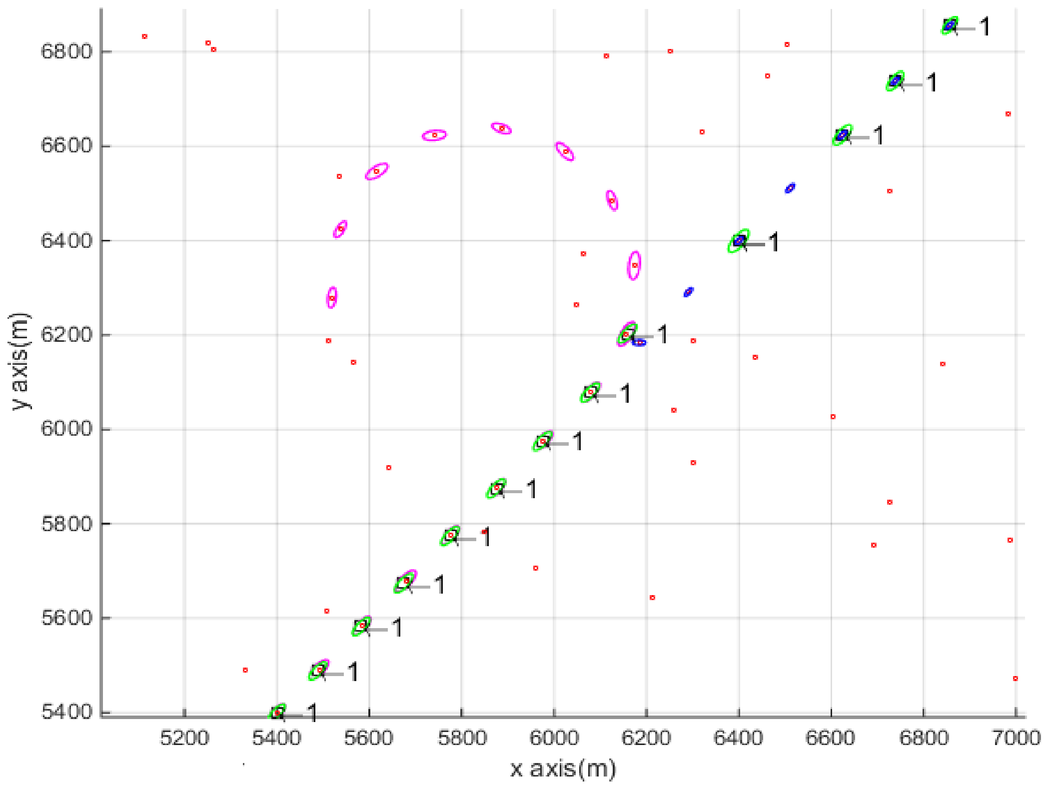

Figure 25: Zoomed version of Figure 24.

In Figure 26, the extensions of the fighter aircraft are shown before the missile launch. In the figure the pink ellipses represent the true extension, green ellipses represent the estimated extension. It is obvious that the extension computation is accurate for the fighter aircraft before the missile launch. In Figure 27, the situation after the missile launch is shown. It is observed that green ellipses which are much larger than the blue ellipses which represent the true extension. This happens due to the fact that the extent estimates take time to converge when the fighter aircraft is assigned with the missile measurements.

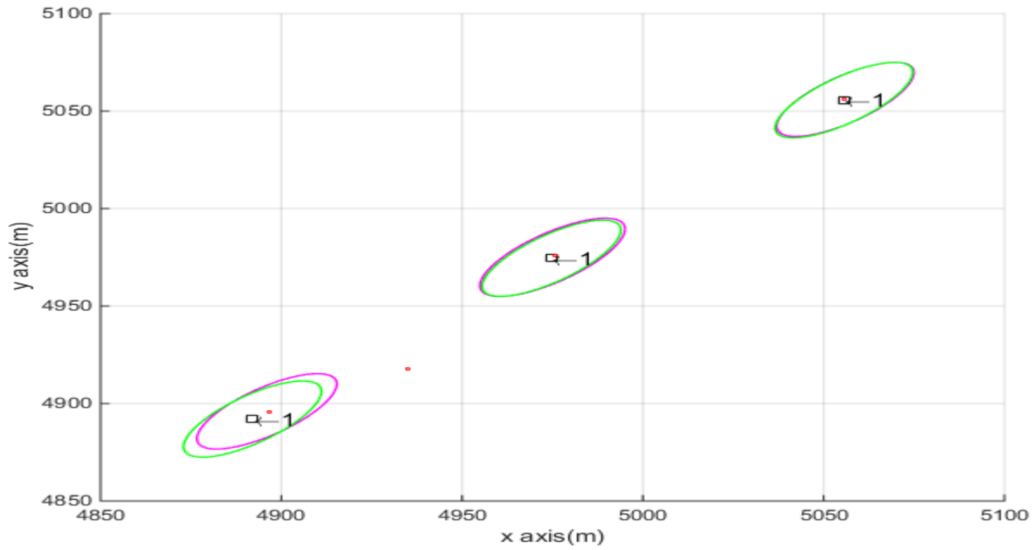

Figure 26: The true and the estimated extensions of the fighter aircraft before the missile launch.

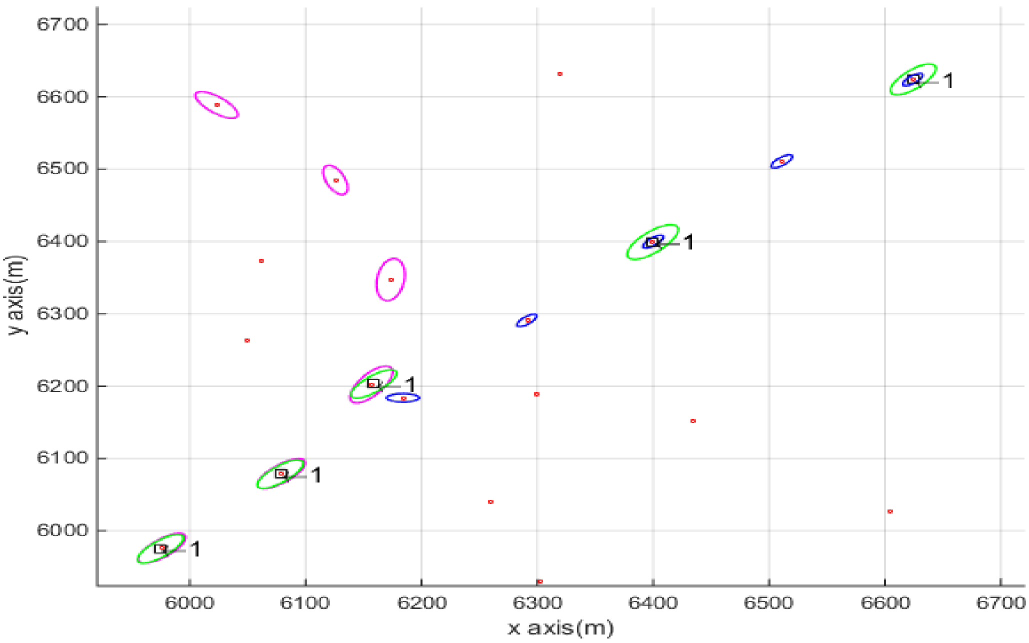

Figure 27: The true and the estimated extensions of the fighter aircraft after the missile launch.

6.3.2 The Results of IMM-PDA filter with Orguner's Likelihood

In the previous chapter it was seen that Orguner's likelihood gave smaller targets a larger likelihood. As a result, the IMM-PDA tracker using this likelihood function mostly associates the cluster of the missile with the track of the fighter aircraft which leads to a false fighter aircraft track as illustrated in Figure 28. In the typical run, in Figure 28, the tracker simply thinks that fighter aircraft goes straight on the trajectory of the missile and a new track is created from the future measurements of the fighter aircraft.

In Figure 29 and Figure 30, we show the true and the estimated extents of the fighter aircraft before and after the missile launch. It is seen that while the extension estimates match the cluster size well before the missile launch, the extension estimates do not converge to the measured cluster sizes after the missile for a long time.

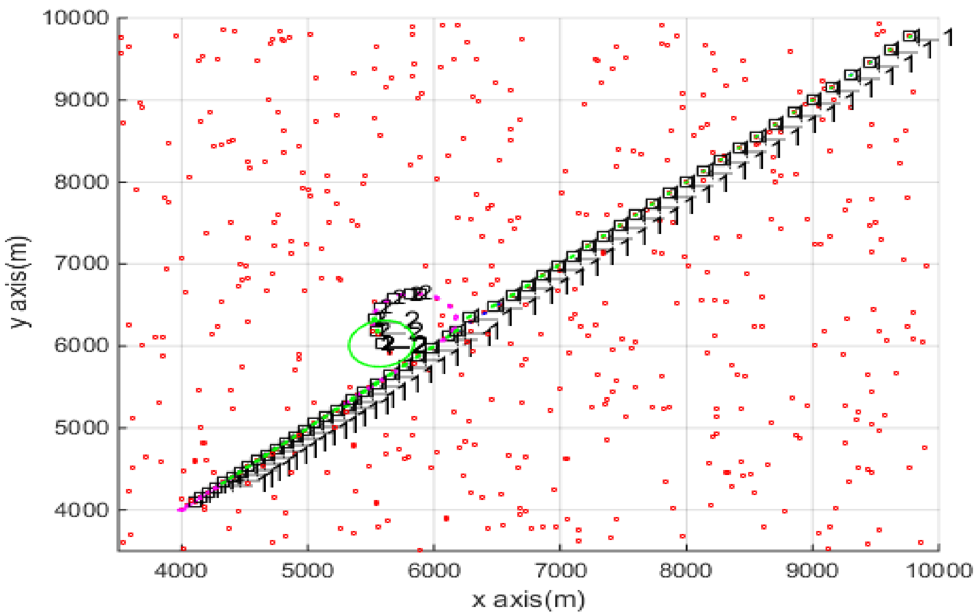

Figure 28: Results of a typical run of the IMM filter using Orguner's likelihood.

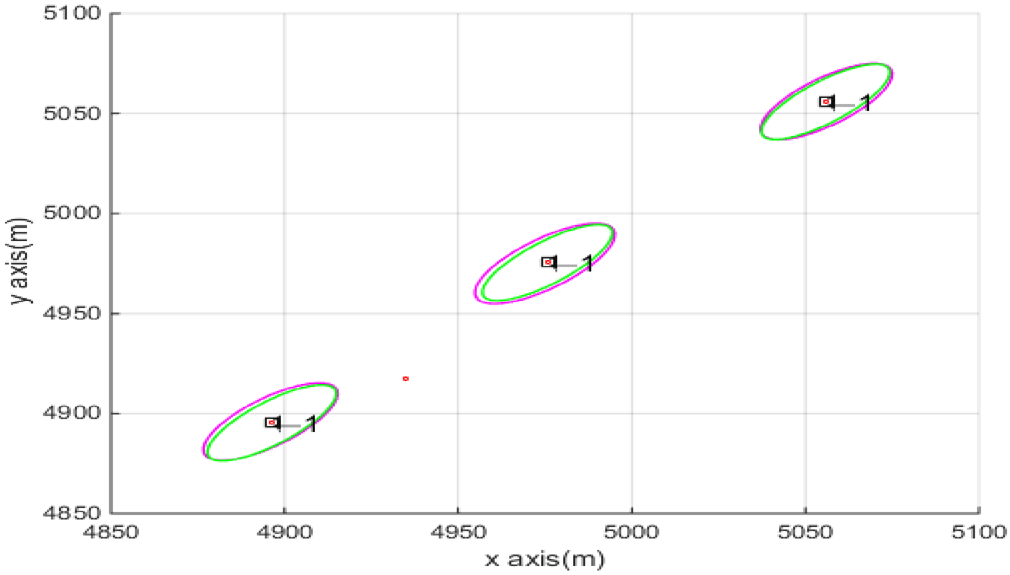

Figure 29: Estimated and true target extents of the fighter aircraft for the IMM-PDA with Orguner's likelihood before the missile launch.

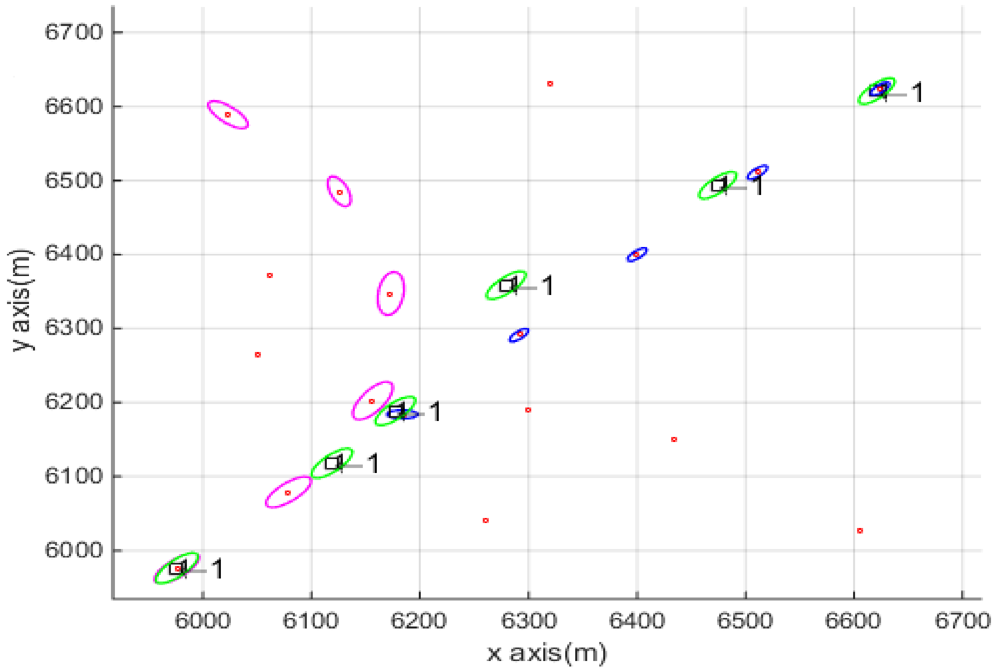

Figure 30: Estimated and true target extents of the fighter aircraft for the IMM-PDA with Orguner's likelihood after the missile launch.

6.3.3 The Results of IMM-PDA filter with the Proposed Likelihood

A typical run of the IMM-PDA filter using the proposed likelihood is shown in Figure 31. As observed the filter is capable of associating the correct cluster measurement to the fighter aircraft. In Figure 32 and Figure 33, it is seen that the extension of both tracks are computed accurately because the targets can be separated thanks to better data association.

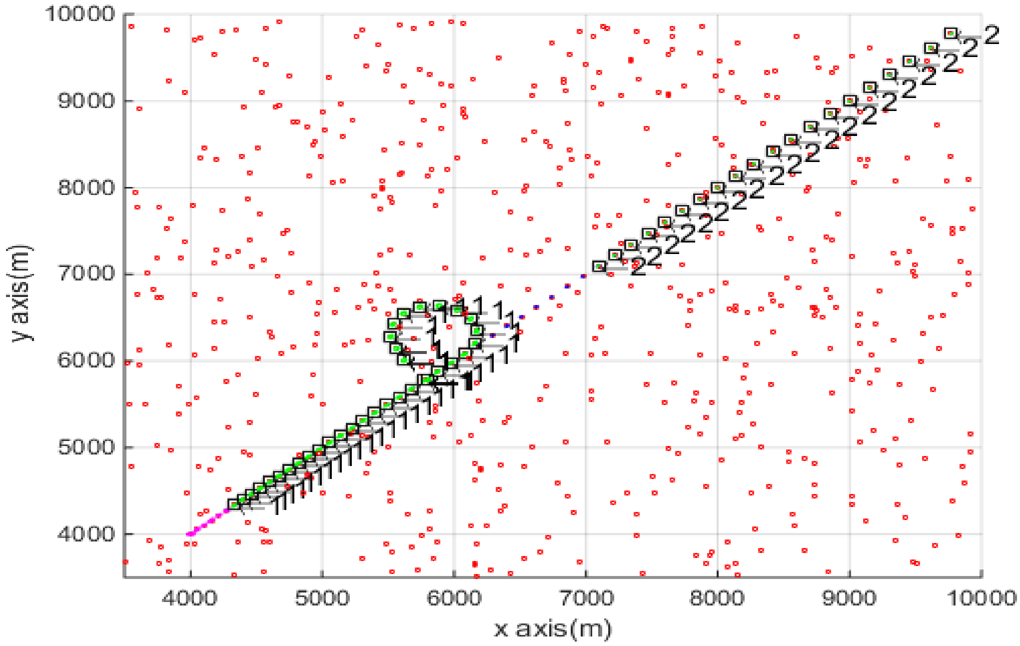

Figure 31: Results of a typical run of the IMM filter using the proposed likelihood.

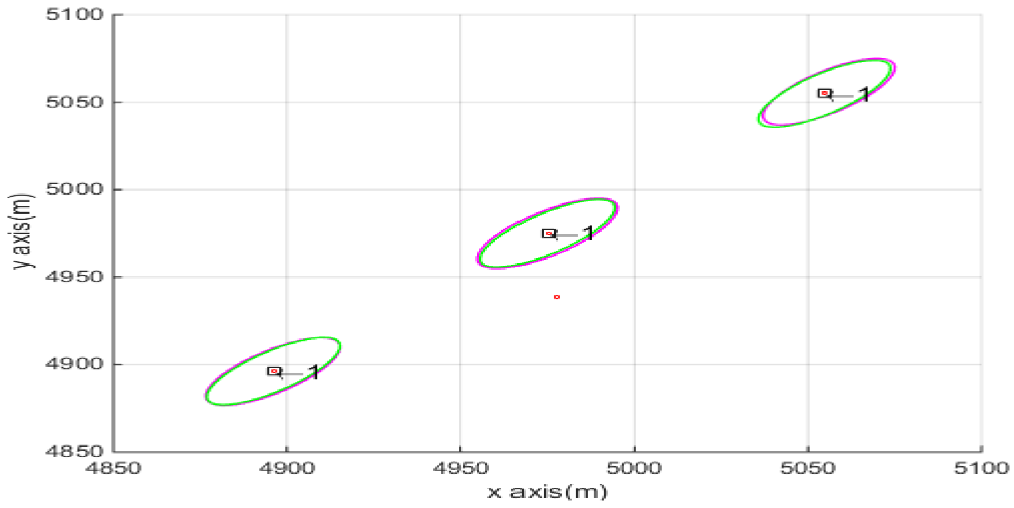

Figure 32: Estimated and true target extents of the fighter aircraft for the IMM-PDA with the proposed likelihood before the missile launch.

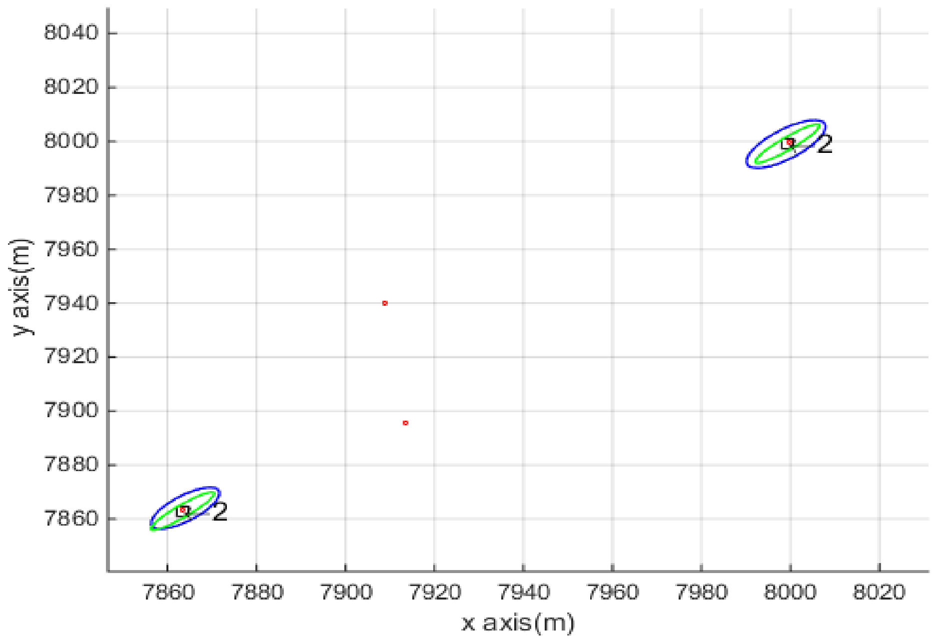

Figure 33: Estimated and true target extents of the fighter aircraft for the IMM-PDA with the proposed likelihood after the missile launch.

6.3.4 Monte Carlo Run Results for the IMM-PDA Filters with Different Likelihoods

The average RMS errors and the percentage of runs with correct data association for the fighter aircraft and the missile are given in Table 1 and Table 2 respectively. As observed performance measures are much better for the filters with the proposed likelihood than for those using the other likelihoods.

Table 1: The performance measures for the fighter aircraft.

	IMM-PDA with Feldmann's Likelihood	IMM-PDA with Orguner's Likelihood	IMM-PDA with Proposed Likelihood
Percentage of Correct Data Association	67%	20%	98%
RMS position error (m)	5.7238	8.9853	3.7997
RMS velocity error (m/s)	5.2849	8.5624	3.3133
RMS extension error (m)	7.2129	8.5716	7.3734

Table 2: The performance measures for the missile.

	IMM-PDA with Feldmann's Likelihood	IMM-PDA with Orguner's Likelihood	IMM-PDA with Proposed Likelihood
Percentage of Correct Data Association	67%	20%	98%
RMS position error (m)	28.4474	82.7754	2.7899
RMS velocity error (m/s)	22.8347	22.5537	2.0450
RMS extension error (m)	11.9793	120.9314	3.6440

6.4 Simulation Results for the IMM-PDA filter with Close Targets

In this section, we consider the scenario shown in Figure 18 and Figure 19 and run the IMM-PDA filter (using the proposed likelihood function) with the algorithm for handling close targets which is described in Section 5.3. The results are shown in Figure 34. It is seen that the cluster separation can be done accurately which results in correct data association when the targets are close and the extensions of both tracks are computed accurately. The numbers in Figure 34 denote track ID/time stamp.

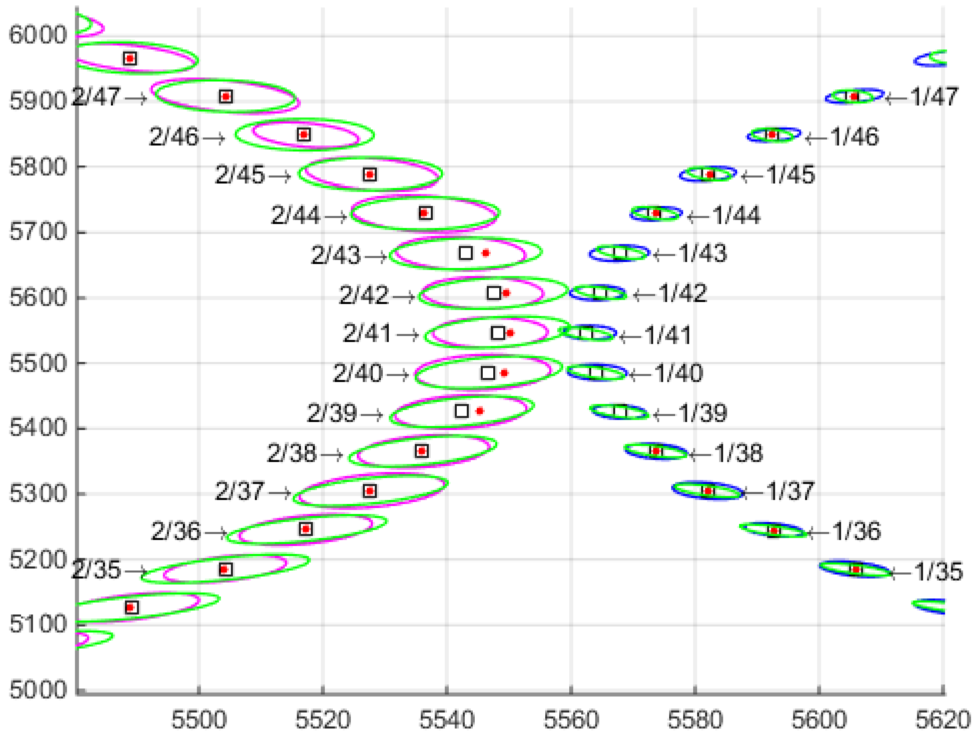

Figure 34: Result of IMM-PDA filter with predicted likelihood for close extended targets.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

An IMM-PDA filter is proposed for tracking extended targets which uses random matrices for modeling the target extent. There are many alternative extended target trackers in the literature but these trackers use many alternative partitionings/clustering of the set of measurements at each scan to get close to optimality. The IMM-PDA filter proposed in the thesis, on the other hand, is specifically designed to use a single partitioning/clustering of the measurement data to be useful in computationally constrained applications while sacrificing optimality.

When an IMM-PDA filter with kinematic and extent states uses clustered measurement data, the predictive likelihood function used for calculating hypothesis probabilities becomes very important. In this thesis, we have shown that the predictive likelihood functions proposed for random matrix models are not very suitable for this purpose. Instead we proposed an alternative predictive likelihood function and shown its advantages by comparing IMM-PDA filters using different likelihood functions. The simulation studies have shown that the new likelihood functions could increase the data association performance to a great extent compared to other likelihood functions, which improved the kinematic and extent state estimation performance of the IMM-PDA filter significantly.

The pre-clustering step used in the IMM-PDA filter has the side effect of combining the measurement clusters of close targets. Since alternative partitionings/clustering of the measurement set are not used, this would lead to track loss in the IMM-PDA filter. In order to solve this problem we have proposed a specific algorithm for handling close targets. This algorithm detects the close targets when their estimates are close and if their clusters are merged separates the merged clusters to avoid low tracking performance. The algorithm has shown good performance on an example simulation.

Future studies that can be pursued in the subject area considered in this thesis can be the following:

- The tracking algorithm presented here assumed that the tracking of multiple targets can be made by applying single target trackers (IMM-PDA) for each target. A future study might apply the ideas given here to an IMM-JPDA algorithm.
- It has been shown in the thesis that the predictive likelihood functions proposed in the literature have some shortcomings in favoring clustered measurements with close size to the predicted extent. This has been done using simulations. A theoretical analysis of the reasons for the unexpected behavior of these likelihood functions has been left for a future study.
- A new predictive likelihood function was proposed in the thesis using heuristic arguments. Theoretically more justified likelihood functions can be found in a detailed future study.
- The algorithm proposed for handling close targets in this thesis modifies the already clustered set of measurements based on the predicted target extents. A more unified framework in this respect would be to come up with a clustering algorithm which actually uses the predicted target estimates as a prior information, which is left as an interesting future study.

REFERENCES

[1] K. Granström, “Extended target tracking using PHD filters,” Pg. 71, Ph.D. Thesis, Department of Electrical Engineering, Linköping University, SE–581 83 Linköping, Sweden.

[2] M.J. Waxman and O.E. Drummond, “A bibliography of cluster (group) tracking,” Proc. SPIE, vol. 5428, pp. 551–560, 2004.

[3] M. Buhren and B. Yang, “Simulation of automotive radar target lists using a novel approach of object representation,” in Proceedings of IEEE Intelligent Vehicles Symposium, pp. 314–319, 2006.

[4] L. Hammarstrand, L. Svensson, F. Sandblom, and J. Sorstedt, “Extended object tracking using a radar resolution model,” IEEE Transactions on Aerospace and Electronic Systems, vol. 48, no. 3, pp. 2371–2386, 2012.

[5] M. Baum and, U. Hanebeck, “Random hypersurface models for extended object tracking,” in Proceedings of IEEE International Symposium on Signal Processing and Information Technology (ISSPIT, pp. 178– 183), Dec 2009.

[6] W. Koch, “Bayesian approach to extended object and cluster tracking using random matrices,” IEEE Transactions on Aerospace and Electronic Systems, vol. 44, no. 3, pp. 1042–1059, 2008.

[7] M. Feldmann, D. Franken, and W. Koch, “Tracking of extended objects and group targets using random matrices,” IEEE Transactions on Signal Processing, vol. 59, no. 4, pp. 1409–1420, Apr. 2011.

[8] U. Orguner, “A variational measurement update for extended target tracking with random matrices,” IEEE Transactions on Signal Processing, vol. 60, no. 7, pp. 3827–3834, Jul. 2012.

[9] K. Granström and U.Orguner, “New prediction update for extended targets with random matrices,” IEEE Transactions on Aerospace and Electronic

Systems, vol, 50,no. 2, pp, 1577-1589, Apr. 2014.

[10] J. Lan and X.R. Li, “Tracking of extended object or target group using random matrix - Part I: New model and approach,” IEEE Transactions on Aerospace and Electronic Systems, Vol. 52, No.6, pp. 2973-2989, Dec. 2016.

[11] K. Granström, C. Lundquist, and O. Orguner, “Extended target tracking using a Gaussian-mixture PHD filter,” IEEE Transactions on Aerospace and Electronic Systems, vol. 48, no. 4, pp. 3268–3286, Oct. 2012.

[12] M. Wieneke and W. Koch, “A PMHT approach for extended objects and object groups,” IEEE Transactions on Aerospace and Electronic Systems, vol. 48, no. 3, pp. 2349–2370, Jul. 2012.

[13] K. Granström and U. Orguner, “A PHD filter for tracking multiple extended targets using random matrices,” IEEE Transactions on Signal Processing, vol. 60, no. 11, pp. 5657-5671, Nov. 2012.

[14] K. Granström, A. Natale, P. Braca, G. Ludeno, ‘‘Gamma Gaussian Inverse Wishart probability hypothesis density for extended target tracking using X-Band marine radar data,’’ IEEE Transactions on Geoscience and Remote Sensing, vol. 53, no. 12, Dec. 2015.

[15] M. Schuster, R. Jeuter, G. Wanielik, ‘‘Probabilistic data association for tracking extended targets under clutter using random matrices,’’ in Proceedings of 18th International Conference Information Fusion, Jul. 2015.

[16] D.J. Salmond, M.C. Parr, ‘‘Track maintenance using measurements of target extent’’, IEE Proc.-Radar Sonar Navig. vol. 150, no. 6, Dec. 2003.

[17] J. Vermaak, N. Ikoma, S. J. Godsill, ‘‘Sequential Monte Carlo framework for extended object tracking,’’ IEE Proc.-Radar Sonar Navig. vol. 152, no.5, Oct. 2005

[18] Z. Zhong, H.Meng, X. Wang, ‘‘Extended target tracking using an IMM based Rao-Blackwellised unscented Kalman filter,’’ in Proceedings of International Conference on Signal Processing, 2008.

[19] M. Baum and U.D. Hanebeck, “Shape tracking of extended objects and group targets with star-convex RHMs,” in Proceedings of the International Conference on Information Fusion, Chicago, Jul. 2011.

[20] N. Petrov, L. Mihaylova, A. Gning, ‘‘Rectangular extended object tracking with box particle filter using dynamic constraints,’’ in Proceedings of the IET Target Tracking and Data Fusion Conference, Apr. 2014.

[21] C. Lundquist, K. Granström, and U. Orguner, “An extended target CPHD filter and a gamma Gaussian inverse Wishart implementation,” IEEE Journal of Selected Topics in Signal Processing, Special Issue on Multitarget Tracking, vol. 7, no. 3, pp. 472–483, Jun. 2013.

[22] K. Granström, M. Fatemi, L. Svensson, “Gamma Gaussian inverse-Wishart Poisson multi-Bernoulli Filter for Extended Target Tracking,’’ in Proceedings of International Conference on Information Fusion, Jul. 2016.

[23] Y. Bar-Shalom, P.K. Willett and X. Tian, Tracking and Data Fusion: A Handbook of Algorithms,’ YBS Publishing, 2011.

[24] S. Blackman and R. Popoli, Design and Analysis of Modern Tracking Systems, Norwood, MA. Artech House,1999.

[25] U. Orguner, Lecture Notes of the Course EE793 Target Tracking, Department of Electrical and Electronics Engineering, Middle East Technical University, Ankara, Turkey.
http://users.metu.edu.tr/umut/ee793/files/METULecture3.pdf

[26] G. Welch, and G. Bishop, “An Introduction to the Kalman Filter,” Transaction of Department of Computer Science University of North Carolina, Chapel Hill, 2002.

[27] Y. Bar-Shalom, X.R. Li, and T. Kirubarajan, ‘Estimation with Applications to Tracking and Navigation, New York: Wiley, 2001.

[28] B.D.O Anderson and J.B. Moore, Optimal Filtering, New York: Dover, 2005.

[29] R. Faragher, “Understanding the basis of the Kalman filter via a simple and intuitive derivation,” IEEE Signal Processing Magazine, vol. 29, no. 5, pp.

128–132, Sep. 2012.

[30] Y. Bar-Shalom, F. Daum and J. Huang, "The probabilistic data Association filter," IEEE Control Systems Magazine, vol. 29, no. 6, pp. 82-100, Nov. 2009.

[31] S.S. Blackman, Multiple Target Tracking with Radar Applications, Norwood, MA: Artech House, 1986.

[32] D.A. Harville, Matrix Algebra From a Statistician's Perspective, New York: Springer-Verlag, 1997.

[33] M.A. Richards, Fundamentals of Radar Signal Processing, 2nd ed., McGraw Hill, 2005.

[34] A.A. Gorji, R. Tharmarasa and T. Kirubarajan, "Performance measures for multiple target tracking problems", in Proceedings of 14th International Conference on Information Fusion, Jul. 2011.