TRACKING NON-ELLIPSOIDAL EXTENDED OBJECTS USING SEQUENTIAL MONTE CARLO

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

BY

SÜLEYMAN FATİH KARA

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

SEPTEMBER 2018

Approval of the thesis:

TRACKING NON-ELLIPSOIDAL EXTENDED OBJECTS USING SEQUENTIAL MONTE CARLO

submitted by **SÜLEYMAN FATİH KARA** 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. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. Tolga Çiloğlu
Head of Department, **Electrical and Electronics Engineering** _____

Assist. Prof. Dr. Emre Özkan
Supervisor, **Electrical Electronics Engineering Dept., METU** _____

Examining Committee Members:

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

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

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

Assist. Prof. Dr. Mustafa Mert Ankaralı
Electrical and Electronics Eng. Dept., METU _____

Prof. Dr. Özgül Salor Durna
Electrical and Electronics Eng. Dept., GAZİ UNI. _____

Date: _____

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

Name, Last Name: Süleyman Fatih Kara

Signature :

ABSTRACT

TRACKING NON-ELLIPSOIDAL EXTENDED OBJECTS USING SEQUENTIAL MONTE CARLO

Kara, Süleyman Fatih
M.S., Department of Electrical and Electronics Engineering
Supervisor : Assist. Prof. Dr. Emre Özkan

September 2018, 73 pages

The problem of extended target tracking is considered in which the target extent is represented with multiple ellipses. The resulting inference problem, which is considered in the sequential Monte Carlo (SMC) framework, includes association of the measurements between sub-objects. We make use of different particle filtering approaches to solve the aforementioned association problem under the assumption of known extent. When the extent is unknown, parameters of the multiple ellipses should also be estimated. For this purpose, a particle filter based method is derived for joint estimation of target's kinematic and extent states. The proposed method uses variational Bayes technique to obtain an approximate conditional analytical expression, which enables the use of Rao-Blackwellization (a.k.a. marginalization) idea in particle filtering.

Keywords: Sequential Monte Carlo, Extended Target Tracking, Marginalized Particle Filter, Variational Inference

ÖZ

SIRALI MONTE CARLO KULLANARAK ELİPS OLMAYAN GENİŞLETİLMİŞ NESNELERİN TAKİBİ

Kara, Süleyman Fatih
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi : Dr. Öğr. Üyesi Emre Özkan

Eylül 2018 , 73 sayfa

Çoklu elips ile temsil edilen genişletilmiş hedef takip problemi ele alınmıştır. Ortaya çıkan ve içerisinde ölçümlerin alt nesnelere sınıflandırılmasını içeren bu problem sıralı Monte Carlo (SMC) teknikleri kullanılarak değerlendirilmiştir. Çeşitli parçacık filtre tabanlı yöntemler kullanılarak bu birleştirme problemi hedefin şeklinin bilindiği farz edilerek çözülmüştür. Hedefin şeklinin bilinmediği takdirde ise çoklu elips parametreleri de tahmin edilmelidir. Bu amaçla hedefin hareketsel durumunu ve şeklini birlikte kestirmek için parçacık filtresi tabanlı bir yöntem tasarlanmıştır. Bu tasarlanan yöntem, değişken Bayes tekniği kullanarak yaklaşık koşullu analitik ifadeler elde eder ve bu ifadeleri parçacık filtresinde marjinalleşme yöntemi uygulamak için kullanır.

Anahtar Kelimeler: Sıralı Monte Carlo, Genişletilmiş Hedef Takibi, Marjinal Parçacık Filtresi, Değişken Çıkarımsama

To my family

ACKNOWLEDGMENTS

I would like to express my deep gratitude to my supervisor Assist. Prof. Dr. Emre Özkan for making me start this study, his valuable supervision and valuable contributions throughout the preparation of this thesis.

I would like to forward my appreciation to my lovely family for their unconditional love and encouragements.

I would like to thank Aselsan for their support.

Most of all, I want to thank my wonderful wife Büşra Kara for always believing in me, for her love and motivation support even in my most difficult moments.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ALGORITHMS

ALGORITHMS

LIST OF ABBREVIATIONS

SMC	Sequential Monte Carlo
PF	Particle Filter
MPF	Marginalized Particle Filter
KF	Kalman Filter
EKF	Extended Kalman Filter
UKF	Unscented Kalman Filter
VB	Variational Bayes
KL	Kullback-Leibler
MC	Monte Carlo
ETT	Extended Target Tracking
IID	Independent Identically Distributed
RMSE	Root Mean Square Error
IOU	Intersection Over Union

CHAPTER 1

INTRODUCTION

Many systems in the field of engineering (statistics, economics, etc.) can be modeled by state-space models (SSMs). SSMs are constructed on a definition of the state variable, together with its dynamics, and its relation to measurements. In most cases, the state variables cannot be observed directly. Therefore, some elements of the state must be estimated from the measurements. This estimation problem can be cast into Bayesian framework, where one tries to find the posterior density of the states given the observations.

For linear Gaussian SSMs, Kalman filter (KF) provides the optimal solution to the state estimation problem by minimizing the mean square error. It computes the exact posterior density of the states given the measurements. However, nonlinear SSMs are common in practical problems and using KF on these systems is not possible without modifications. There are two common variants of KF which apply to nonlinear systems: extended Kalman filter (EKF) [19] and unscented Kalman filter (UKF) [20]. The EKF linearizes the equations that describe the system using first-order terms of Taylor series expansion so that the standard KF can be used with the linearized model. However, if the higher order terms are not negligible, linearization errors will result in a poor approximation. UKF, on the other hand, uses carefully chosen so-called sigma points to approximate relevant integrals. Since both methods approximate the posterior density with a Gaussian, they both fail to represent the posterior density accurately when the posterior involves multi-modalities.

At the beginning of the '90s, the first sequential Monte Carlo (SMC) based algorithm was proposed by Gordon et al. [10] as a solution to the state estimation

problem for nonlinear systems. SMC methods (a.k.a. particle filters) [1] are robust tools which are based on simulating a number of hypotheses referred to as particles. The particles that do not match the observations are discarded, and the rest are updated in a sequential manner. Particle filters (PFs) find application in positioning, target tracking, fault diagnostic, network systems, signal and image processing, bioinformatics, economics, robotics, etc. [3], [16], [28]. In this thesis, we will deal with a specific problem in the target tracking literature and obtain solutions using PFs.

Early target tracking algorithms were developed to process radar measurements which monitor large surveillance regions. In these methods, the targets are modeled as point sources that generate at most one measurement per scan. With evolving sensor technology, sensor systems which are capable of collecting multiple measurements from a single object were developed. The measurements are assumed to originate from different spatially distributed detection/reflection points on the target's surface. Using multiple measurements, one can extract more information from the target, such as its shape, size, etc. The parameters that define the target extent are called as the extent state variables whereas the parameters that describe the dynamics and motion of the target are called as the kinematic state variables. Estimating the extent state jointly with the kinematic state is called as extended target tracking (ETT). Early methods for ETT assume a pre-determined geometric shape for representing the target extent, such as a rectangle, stick, circle or an ellipse [6], [12], [13]. Among these methods, the elliptical extent model (a.k.a. random matrix approach), which was initially proposed by Koch [23], has become the most popular. In this approach, the extent is represented by a symmetric positive semi-definite random matrix, which is estimated jointly with the kinematic state in a recursive manner. However, in this model, the kinematic state is strongly coupled with the extent state which limits the applicability of the model to more general problems. In [9], Feldman *et al.* have overcome this restriction by removing the dependency between the extent and the kinematic states at the expense of exact inference. More recently, a variational Bayes based measurement update

[1] Particle filter is a subset of SMC methods

solution is proposed [27] for the improved random matrix model in [9]. The method provides an approximate posterior by minimizing the Kullback-Leibler (KL) divergence between the true and the approximate posterior. In [25], two random matrix based ETT models are proposed with multiple model extensions to track dynamic objects.

Other than random matrix based methods, there are also algorithms based on random hyper-surface models [4]. More complex shapes are represented with this model and successfully estimated using corresponding methods in [5]. Another random hyper-surface based method is proposed in [31] using Gaussian process. A comprehensive survey of the ETT literature can be found in [11].

The focus of this study is to extend the random matrix approach to multi-ellipse ETT models. Representing extended targets with a single ellipse can result in a coarse estimation of the true extent. Inability to capture the extent details also results in poor tracking performance (see: [31]). Therefore we investigate alternative ways of representing an extended target using multiple ellipses and seek appropriate inference methods within SMC.

First, we investigate possible marginalization strategies to find an efficient PF for a simplified version of the multi-ellipse ETT problem. In the simplified multi-ellipse ETT problem, the extent parameters are assumed to be known and the PF focuses on solving the association problem between the measurements and the ellipses. For this purpose, three different PFs are implemented and their performances are compared in simulations.

Second, we treat the full multi-ellipse ETT problem within SMC and propose an efficient algorithm that is capable of tracking an object and estimating its multi-ellipse extent jointly. The model defines a unified kinematic model for the multiple ellipses. Unlike the existing methods proposed for multi-ellipse ETT [14], [17], [24] our solution does not require any clustering, partitioning, mixture reduction and merging methods. Our method is based on marginalized particle filters where the analytical conditional expressions are found approximately by minimizing the KL divergence between the true and the approximate densities.

The rest of this chapter covers background information for SMC methods. Chapter 2 introduces some of the common PF approaches in the literature. These approaches are employed in a multi-ellipse ETT problem with known extents in Chapter 3. In Chapter 4, an SMC based solution to multi-ellipse unknown extent ETT problem is presented. Finally, the conclusion is given in Chapter 5.

CHAPTER 2

PARTICLE FILTER

The first practical SMC based filtering algorithm was proposed in 1993 by Gordon et. al. [10]. Several number of algorithms sharing similar ideas were proposed such as bootstrap filters [10], survival of the fittest [21], Monte Carlo filters [22] and condensation [18]. These algorithms are now referred to as SMC methods or particle filters. Particle filters are used to approximate the posterior density of the states given the observations for general state space models (SSM).

Here, we will consider discrete time SSM where the state sequence $\{x_k\}_{k\geq 0}$ follows a Markov process. State variable itself is latent, and it is observed only via available measurements $\{y_k\}_{k\geq 1}$. This system can also be interpreted as a hidden Markov model (HMM) with continuous state variables. The system can be described with the following equations,

$$x_{k+1} = f(x_k, e_k), \tag{2.1a}$$

$$y_k = h(x_k, v_k), \tag{2.1b}$$

where $f(\cdot)$ is a known linear/nonlinear function of the system dynamics and $h(\cdot)$ is a known linear/nonlinear function of the measurements. e_k and v_k represents the noise terms, commonly known to as the process noise and the measurement noise, respectively.

PF approximates the posterior density $p(x_{0:k}|y_{1:k})$ using a set of particles and their weights,

$$\widehat{p}^N(x_{0:k}|y_{1:k}) = \sum_{i=1}^{N} w_{0:k}^{(i)} \delta_{x_{0:k}^{(i)}}(x_{0:k}), \tag{2.2}$$

where $\{x_{0:k}^{(i)}\}_{i=1}^{N}$ are referred to as particles and $\{w_{0:k}^{(i)}\}_{i=1}^{N}$ represent the corresponding weights [28].

The flow of a generic PF can be described as follows: At each time step, we extend the existing state trajectories by generating samples from an importance density. Then a weight update is performed to reflect the effect of the new measurement on the weights. Lastly, particles with higher weights, (which are more likely to explain the dynamics and measurements) are copied and the particles with negligible weights are deleted in the resampling stage. This recursive process can be illustrated in the figure below.

Figure 2.1: Particle filter flow scheme.

Two essential steps of PF are the sampling and weight update stages. We first sample the particles from the importance density $q(x_k|x_{0:k-1}^{(i)}, y_k)$, then, the weights are updated according to

$$\bar{w}_k^{(i)} = \frac{p(y_k|x_k^{(i)})p(x_k^{(i)}|x_{k-1}^{(i)})}{q(x_k^{(i)}|x_{0:k-1}^{(i)}, y_k)} w_{k-1}^{(i)}. \tag{2.3}$$

After the weight update the weights are normalized to sum up to 1,

$$w_k^{(i)} = \bar{w}_k^{(i)} / \sum_{s=1}^{N} \bar{w}_k^{(s)}. \tag{2.4}$$

Lastly, we perform resampling [3]. This process is iterated from $k = 1$ to the final time, T. A summary of a generic PF is given in Algorithm 2.1.

2.1 Particle Filter Output

Output of a particle filter at each time instant is the weighted samples which approximate the posterior density. If a point estimate of the state is needed, different estimates can be extracted with respect to different criteria. One can choose to use the particle with the highest weight, which may not be accurate

Algorithm 2.1: Summary of the PF

1: Initialization at $k = 0$:
2: Sample the initial particles from the initial density $\{x_0^{(i)} \sim p_0(x_0)\}_{i=1}^N$ and set their weights equally $\{w_0^{(i)} = 1/N\}_{i=1}^N$
3: Iterations:
4: **for all** $k = 1, \ldots, T$ **do**
5: **for all** $i = 1, \ldots, N$ **do**
6: Sample the particles from importance density $x_k^{(i)} \sim q(x_k | x_{0:k-1}^{(i)}, y_k)$
7: Update the weights according to (2.3)
8: **end for**
9: Normalize the weights according to (2.4)
10: Resample, if necessary
11: **end for**

due to resampling stages involved in PF. The most common choice is to minimize the expected mean square error (MSE) and use the mean of the approximated posterior as a point estimate,

$$\hat{x}_{0:k}^{MSE} = \sum_{i=1}^{N} w_{0:k}^{(i)} x_{0:k}^{(i)}. \tag{2.5}$$

In PF, state trajectories $x_{0:k}$ (particles) which are candidates for representing the true state are sampled from the importance density. Therefore, a good choice of importance density improves the performance of PF [3]. In the following section, we will introduce the simplest, also the most common, choice of importance density, the bootstrap proposal density.

2.2 Bootstrap Particle Filter

The bootstrap particle filter (BPF) is a subset of the generic particle filter whose importance density $q(x_k | x_{k-1}^{(i)}, y_k)$, is chosen as the transition density, $p(x_k | x_{k-1})$ that is induced by (2.1a).

This choice of importance density also simplifies the weight update equation.

Substituting $q(x_k|x_{k-1}^{(i)}, y_k)$ in (2.3) results,

$$w_k^{(i)} \propto \frac{p(y_k|x_k^{(i)})p(x_k^{(i)}|x_{k-1}^{(i)})}{q(x_k^{(i)}|x_{k-1}^{(i)}, y_k)} w_{k-1}^{(i)}, \tag{2.6a}$$

$$w_k^{(i)} \propto p(y_k|x_k^{(i)}) w_{k-1}^{(i)}. \tag{2.6b}$$

The method is summarized in Algorithm 2.2.

Algorithm 2.2: Summary of the BPF

1: Initialization at $k = 0$:
2: **for all** $i = 1, \ldots, N$ **do**
3: Sample $x_0^{(i)} \sim p_0(x_0)$
4: Set initial weights $w_0^{(i)} = \frac{1}{N}$
5: **end for**
6: Iterations:
7: **for all** $k = 1, \ldots, T$ **do**
8: **for all** $i = 1, \ldots, N$ **do**
9: Sample $x_k^{(i)} \sim p(x_k|x_{k-1}^{(i)})$
10: Update the weights according to (2.6b)
11: **end for**
12: Normalize the weights
13: Resample, if necessary
14: **end for**

2.2.1 Example: Comparison of KF and BPF

In linear Gaussian systems, KF provides the exact analytical expression for the posterior density. PF provides an approximation for the posterior density using particles. The performance of PF heavily depends on the number of particles. We will illustrate this dependency in the following simulation. Consider the linear Gaussian system given below,

$$x_{k+1} = 0.7x_k + e_k, \qquad e_k \sim \mathcal{N}(0, 0.1), \tag{2.7a}$$

$$y_k = 0.5x_k + v_k, \qquad v_k \sim \mathcal{N}(0, 0.1), \tag{2.7b}$$

Table 2.1: RMSE values between the BPF and the KF estimates

	Number of Particles		
	10	100	1000
RMSE	0.1342	0.0481	0.0197

where the initial density is chosen as $x_0 \sim \mathcal{N}(0, 1)$. We run the BPF with $N = 10, 100$ and 1000 particles and compare the results. The state estimates of the filters are given in Figures 2.2, 2.3 and 2.4, respectively.

By knowing the fact that KF provides the optimal state estimate (in the mean square sense) we compare the state estimates of BPF with that of KF. Root mean square error (RMSE) between the two estimates is defined as

$$RMSE = \sqrt{\frac{1}{N}\sum_{k=1}^{T}(x_k^{\mathrm{BPF}} - x_k^{\mathrm{KF}})^2}, \tag{2.8}$$

where x_k^{KF} is the state estimate of KF and x_k^{BPF} is the state estimate of BPF. RMSE results are given in Table 2.1.

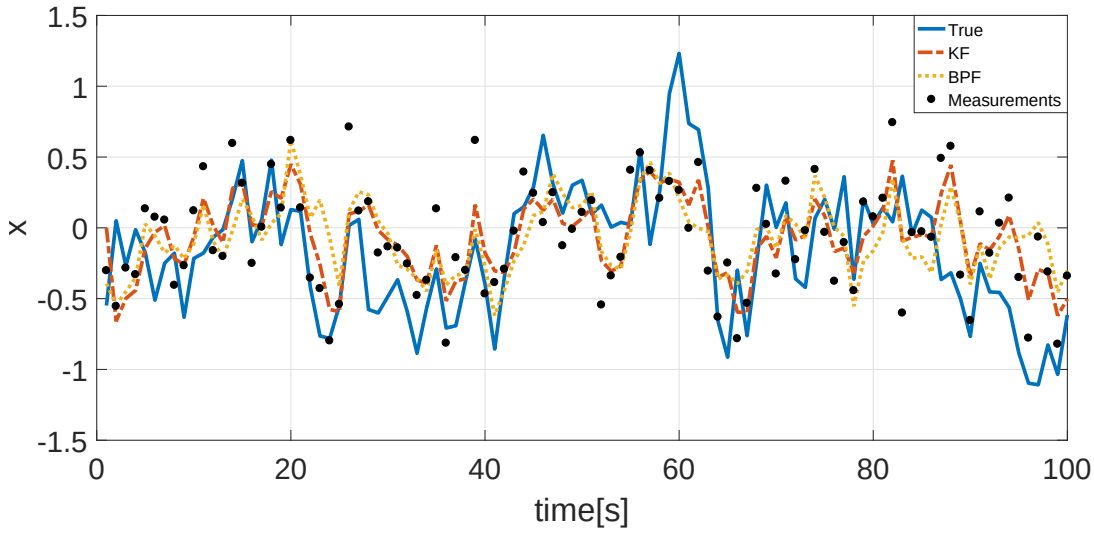

Figure 2.2: Estimation results of the KF and the BPF for 10 particles. The true value is shown with blue solid line, the KF estimate is shown with orange dashed line and the BPF estimate is shown with yellow dotted line. The measurements at each scan are represented with black dots.

As it is observed from the figures and RMSE results, the BPF estimates converge

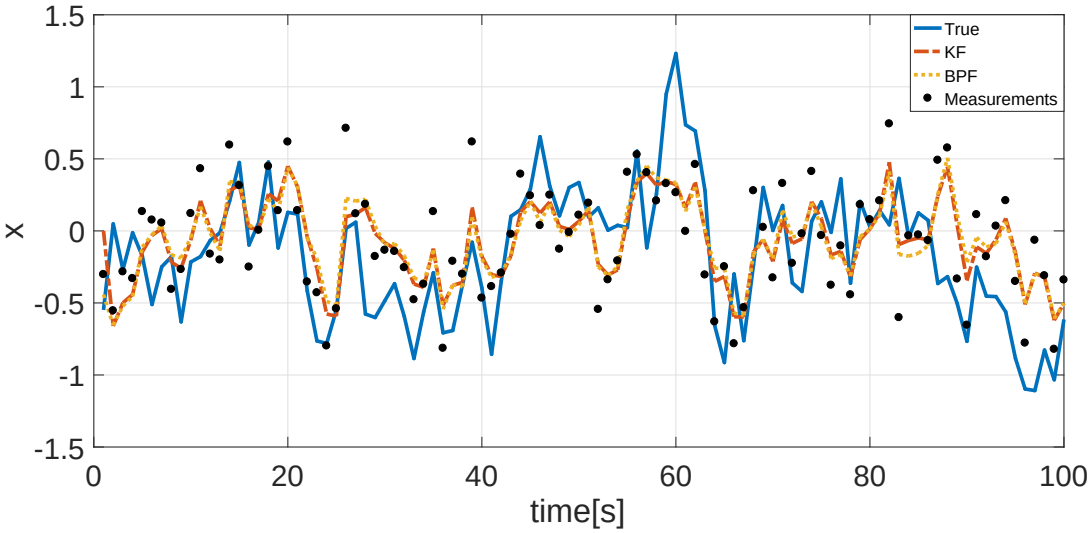

Figure 2.3: Estimation results of the KF and the BPF for 100 particles. The true value is shown with blue solid line, the KF estimate is shown with orange dashed line and the BPF estimate is shown with yellow dotted line. The measurements at each scan are represented with black dots.

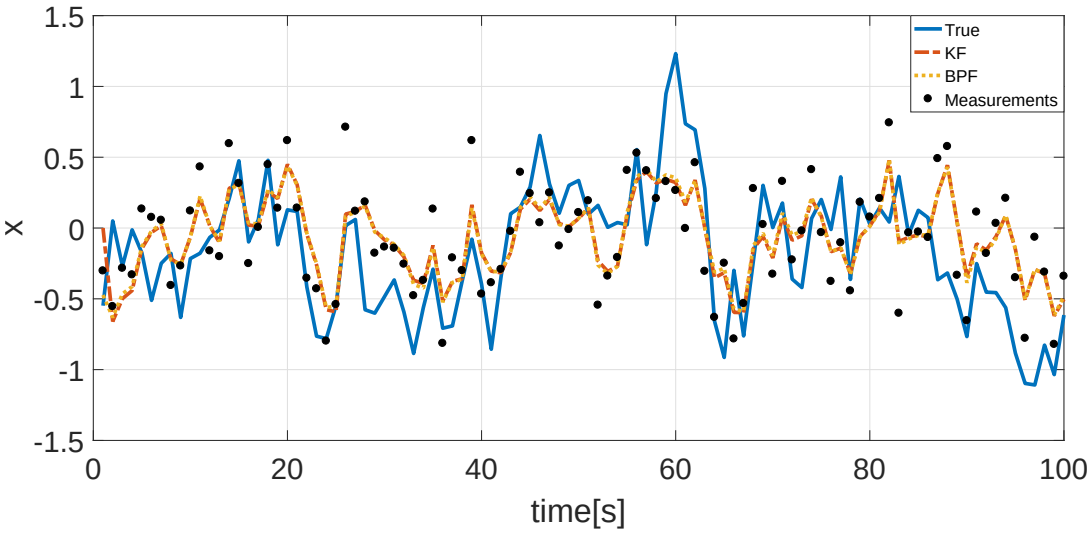

Figure 2.4: Estimation results of the KF and the BPF for 1000 particles. The true value is shown with blue solid line, the KF estimate is shown with orange dashed line and the BPF estimate is shown with yellow dotted line. The measurements at each scan are represented with black dots.

to the KF's with increasing number of particles for a linear Gaussian model.

In the BPF, the importance density only depends on the system model which

can degrade the performance of the algorithm. A more efficient approach is to exploit the information provided by the measurements. This can be achieved by using the measurements in the importance density which is described in the following subsection.

2.3 Optimal Proposal Density

At the sampling step of a PF, new particles are generated by sampling from the importance density. Using the measurement y_k results in a more efficient importance density than the bootstrap proposal density. As a result of that, less particles will have negligible weights after the weight update, which increases particle efficiency. Consider the following choice on the importance density,

$$\begin{aligned} q_{opt}(x_k|x_{k-1}^{(i)}, y_k) &\triangleq p(x_k|x_{k-1}^{(i)}, y_k) \\ &= \frac{p(y_k|x_k, x_{k-1}^{(i)})p(x_k|x_{k-1}^{(i)})}{p(y_k|x_{k-1}^{(i)})}. \end{aligned} \tag{2.9}$$

Substituting (2.9) into (2.3) results in,

$$w_k^{(i)} \propto p(y_k|x_{k-1}^{(i)})w_{k-1}^{(i)}, \tag{2.10}$$

where

$$p(y_k|x_{k-1}^{(i)}) = \int p(y_k|x_k)p(x_k|x_{k-1}^{(i)})dx_k. \tag{2.11}$$

The above choice of proposal density minimizes the variance of the weights $\mathrm{Var}\left(w_k^{(i)}\right)$ and maximizes the effective sample size [3]. Therefore it is called as the optimal proposal density.

To be able to use the optimal proposal density, one should evaluate the integral (2.11) and sample from $p(x_k|x_{k-1}^{(i)}, y_k)$. This is possible only in a few cases, and the most common ones are listed below:

- The density $p(x_k|x_{k-1}^{(i)}, y_k)$ should be Gaussian and the measurement equation should be linear. The system dynamics could be either nonlinear or linear. Detailed derivation for this case is given in [3].

- x_k should be a member of a finite set. In this case, the integral (2.11) turns into a sum and it is possible to sample from $p(x_k|x_{k-1}^{(i)}, y_k)$.

2.4 Marginalization Idea

Particle filters approximate densities by weighted random samples. This approach is quite efficient when the state dimension is low. Unfortunately, the efficiency of a PF does not scale up well with the dimension of the state, hence one needs a vast number of particles to approximate the posterior of high dimensional states. One way to mitigate this problem is to exploit some inherent analytical structures in the target density $p(x_k|y_{1:k})$, if available.

A generic PF aims at approximating the target density $p(x_k|y_{1:k})$. Consider the factorization of the target density into non-linear and conditionally linear parts,

$$p(x_k|y_{1:k}) = p(x_k^l, x_{0:k}^n|y_{1:k}) = p(x_k^l|x_{0:k}^n, y_{1:k})p(x_{0:k}^n|y_{1:k}). \tag{2.12}$$

A Marginalized particle filter (MPF) (a.k.a. Rao-Blackwellized particle filter) [29], [8] aims at sampling from the non-linear states and computing analytical expressions for the conditionally linear part of the state (first factor in (2.12)). By doing so, the target density can be represented as

$$p(x_k^l, x_{0:k}^{n,(i)}|y_{1:k}) \approx \sum_{i=1}^{N} w_k^{(i)} \delta_{x_{0:k}^{n,(i)}}(x_{0:k}^n) p(x_k^l|x_{0:k}^{n,(i)}, y_{1:k}). \tag{2.13}$$

As an analogy, KF computes the posterior density as analytical expressions. PF approximates the posterior density with weighted samples. MPF approximates the posterior by using both weighted samples and analytical expressions. This results in sampling efficiency in PF because, depending on the system model, it might be possible to sample from a low-dimension non-linear state x_k^n, and compute the posterior of the full state x_k.

The MPF requires conditional analytical expressions in the posterior density. Such analytical expressions exist in only certain type of systems. In the following subsection, we will introduce multiple switching dynamics models which exhibit

this property. Later, we will investigate different marginalization approaches for this model.

2.4.1 Multiple Switching Dynamics Models

Common problems in engineering includes nonlinear models which are characterized by different possible modes of operation. These types of problems are referred to as hybrid-state estimation problems which involve continuous state and discrete mode variables [30]. Here, we consider the following dynamic and measurement equations for describing the hybrid system,

$$x_{k+1} = f(x_k, r_k, e_k), \tag{2.14a}$$

$$y_k = h(x_k, r_k, v_k), \tag{2.14b}$$

where r_k is the discrete mode variable. It is possible to model r_k using a M-state, first order Markov chain with transition probabilities,

$$\pi_{m\ell} \triangleq \mathbb{P}\{r_k = \ell | r_{k-1} = m\} \quad (m, \ell \in S), \tag{2.15}$$

where $S \triangleq \{1, \ldots, M\}$. Then, $M \times M$ transition probability matrix is defined as $\Pi \triangleq [\pi_{m\ell}]$ where the elements satisfy the following conditions:

- $\pi_{m\ell} \geq 0$,
- $\sum_{\ell=1}^{M} \pi_{m\ell} = 1$.

An example of hybrid systems is jump Markov linear Gaussian systems where the discrete mode variable is described as a jump Markov process and the continuous state is described as a Gaussian random variable. This system can also be classified as a subset of conditionally linear Gaussian systems (CLGS). The system definition is given as

$$x_{k+1} = F^{r_k} x_k + e_k^{r_k}, \tag{2.16a}$$

$$y_k = H^{r_k} x_k + v_k^{r_k}, \tag{2.16b}$$

where F^{r_k} is and $n_x \times n_x$ matrix which defines the system dynamics and H^{r_k} is an $n_y \times n_x$ matrix defining the relation between the measurements and the state

variables. $e_k^{r_k} \sim \mathcal{N}(0, Q^{r_k})$ and $v_k^{r_k} \sim \mathcal{N}(0, R^{r_k})$ are zero-mean Gaussian noise terms, known as the process noise and the measurement noise, respectively. $Q^{r_k} \in \mathbb{R}^{n_x \times n_x}$ is the process noise covariance matrix where, n_x is the state dimension and $R^{r_k} \in \mathbb{R}^{n_y \times n_y}$ is the measurement noise covariance matrix where, n_y is the measurement dimension. These terms are dependent upon the mode variable r_k, and the system becomes linear Gaussian conditioned on r_k.

It is possible to extend this model by removing the Markov assumption on the discrete state. In that case, the mode variable at time k, will be independent from the previous one at time $k-1$. We will call this system as multinomial CLGS and use it for the multi-ellipse ETT problem. Under the independence assumption, the prior probabilities of the discrete states are distributed according to the probability vector,

$$\pi = [\pi_1, \ldots, \pi_M]. \tag{2.17}$$

The elements of the probability vector are defined as

$$\pi_\ell \triangleq \mathbb{P}\{r_k = \ell\} \quad (\ell \in S) \tag{2.18}$$

where $\sum_{\ell=1}^{M} \pi_\ell = 1$. If the prior assignment probabilities are equal, we have the probability vector

$$\pi = \left[\frac{1}{M}, \ldots, \frac{1}{M}\right]. \tag{2.19}$$

In the following subsections, two different marginalization strategies will be investigated to compute the joint posterior density of the continuous and discrete states for the multinomial CLGS.

2.4.2 Marginalizing Out the Continuous State

In the multinomial CLGS, the target density to be approximated is the posterior of the continuous and discrete states $p(x_k, r_{0:k}|y_{1:k})$. Consider the following factorization of the target density

$$p(x_k, r_{0:k}|y_{1:k}) = p(x_k|r_{0:k}, y_{1:k})p(r_{0:k}|y_{1:k}). \tag{2.20}$$

Within the MPF framework, the posterior of the discrete states can be approximated using particles,

$$\widehat{p}^N(r_{0:k}|y_{1:k}) = \sum_{i=1}^{N} w_k^{(i)} \delta_{r_{0:k}^{(i)}}(r_{0:k}). \tag{2.21}$$

Conditioned on the discrete states, the system equations become linear and Gaussian. Therefore, the conditional posterior of the continuous state can be written as a Gaussian density $p(x_k|r_{0:k}^{(i)}, y_{1:k}) = \mathcal{N}(x_k; \hat{x}_{k|k}^{(i)}, P_{k|k}^{(i)})$. The mean and covariance of this density are calculated analytically using a KF. Furthermore, the posterior density of the continuous and the discrete states are approximated as

$$p(x_k, r_{0:k}^{(i)}|y_{1:k}) \approx \sum_{i=1}^{N} w_k^{(i)} \delta_{r_{0:k}^{(i)}}(r_{0:k}) \mathcal{N}(x_k; \hat{x}_{k|k}^{(i)}, P_{k|k}^{(i)}). \tag{2.22}$$

Iterations of MPF at each time include:

'Measurement Update' and *'Time Update'* steps for continuous variables;

'Sampling' and *'Weight Update'* steps for discrete variables.

The details of the aforementioned steps will be described as follows.

At any time k, we first propagate the continuous state per particle which corresponds to the KF's time update equations provided as

$$\hat{x}_{k|k-1}^{(i)} = F_k \hat{x}_{k-1}^{(i)}, \tag{2.23a}$$

$$P_{k|k-1}^{(i)} = F_k P_{k-1}^{(i)} F_k^T + Q. \tag{2.23b}$$

Then, we sample the discrete state r_k from the importance density,

$$r_k^{(i)} \sim q(\cdot), \tag{2.24}$$

which is the prior probability vector $q(\cdot) = \pi$ if we use the bootstrap proposal density. The importance weights are updated with the following expression,

$$w_k^{(i)} \propto p(y_k|r_{0:k}^{(i)}, y_{1:k-1}) w_{k-1}^{(i)}. \tag{2.25}$$

For the continuous state, the measurement update is performed using the KF

measurement update equations,

$$\hat{x}_k^{(i)} = \hat{x}_{k|k-1}^{(i)} + K_k(y_k - H\hat{x}_{k|k-1}^{(i)}), \tag{2.26a}$$

$$P_k^{(i)} = (I - K_k H)P_{k|k-1}^{(i)}, \tag{2.26b}$$

$$K_k = P_{k|k-1}^{(i)} H^T (H P_{k|k-1}^{(i)} H^T + R)^{-1}. \tag{2.26c}$$

As a last step, resampling is employed, if necessary.

We will call the MPF where the continuous states are marginalized out as CMPF for the rest of this thesis. The CMPF is summarized in Algorithm 2.3.

Algorithm 2.3: Summary of the CMPF

Initialization at $k = 0$:
for all $i = 1, \ldots, N$ **do**
  Sample the discrete state $r_0^{(i)} \sim \pi_0$
  Set the initial weights $w_0^{(i)} = \frac{1}{N}$
  Set the initial parameters of the continuous state $\hat{x}_0^{(i)} = \bar{x}_0$, $\hat{P}_0^{(i)} = \bar{P}_0$
end for
Iterations:
for $k = 1, \ldots, T$ **do**
  for all $i = 1, \ldots, N$ **do**
    Perform time update of the continuous state according to (2.23)
    Sample the discrete state, $r_k^{(i)} \sim q(\cdot)$
    Update the weights according to (2.25)
    Perform the measurement update of the continuous state according to (2.26)
  end for
  Normalize the weights
  Resample, if necessary
end for

2.4.3 Marginalizing Out the Discrete State

In order to marginalize out the discrete state, we slightly modify the target density as $p(x_{0:k}, r_k|y_{1:k})$. In multinomial CLGS, it is possible to obtain an

analytical expression for the density of the discrete state when conditioned on the continuous states. Consider the following factorization of the target density,

$$p(x_{0:k}, r_k|y_{1:k}) = p(r_k|x_{0:k}, y_{1:k})p(x_{0:k}|y_{1:k}). \tag{2.27}$$

The continuous states are expressed using a set of weighted particles [32] where each particle set represents the state trajectory $x_{0:k} \in \mathbb{R}^{k+1\times n_x}$. The posterior of the continuous states can be approximated using particles,

$$\widehat{p}^N(x_{0:k}|y_{1:k}) = \sum_{i=1}^{N} w_{0:k}^{(i)} \delta_{x_{0:k}^{(i)}}(x_{0:k}). \tag{2.28}$$

When $x_{0:k}$ is given, the conditional density of r_k can be found using a conditional HMM filter [32]. We define the mode probabilities as

$$\alpha_k^{(i)}(\ell) \triangleq \mathbb{P}(r_k = \ell|x_{0:k}^{(i)}, y_{1:k}). \tag{2.29}$$

We will call the MPF where the discrete states are marginalized out as DMPF for the rest of this thesis. The basic flow of DMPF for each time instant k, can be summarized as follows.

Due to independence of the multinomial CLGS, we can propagate the mode probabilities as follows,

$$\alpha_{k|k-1}^{(i)}(\cdot) = \alpha_{k-1}^{(i)}(\cdot), \tag{2.30}$$

where $\{\alpha_{k-1}^{(i)}(\ell)\}_{\ell=1}^{M}$ is represented with $\alpha_{k-1}^{(i)}(\cdot)$ for the sake of simplicity. The continuous state x_k is propagated by sampling from the importance density,

$$x_k^{(i)} \sim q_k(x_k|x_{0:k-1}^{(i)}, y_{1:k}). \tag{2.31}$$

Here, one can use the bootstrap proposal density provided as

$$\begin{aligned} q(x_k|x_{0:k-1}^{(i)}, y_{1:k}) =& p(x_k|x_{0:k-1}^{(i)}, y_{1:k-1}) \\ =& \sum_{\ell=1}^{M} f(x_k|x_{k-1}^{(i)})\alpha_{k|k-1}^{(i)}(\ell). \end{aligned} \tag{2.32}$$

When a new measurement is available, we perform the weight update. Since the continuous state $x_k^{(i)}$ carries information about the discrete state r_k, it serves as

an extra measurement [32]. $\alpha_k(\cdot)$ in (2.29) can also be expressed in terms of the joint density

$$\begin{aligned}\alpha_k^{(i)}(\cdot) &= \mathbb{P}(r_k|x_k^{(i)}, x_{0:k-1}^{(i)}, y_k, y_{1:k-1}) \\ &= \frac{\mathbb{P}(r_k, x_k^{(i)}, y_k|x_{0:k-1}^{(i)}, y_{1:k-1})}{\mathbb{P}(x_k^{(i)}, y_k|x_{0:k-1}^{(i)}, y_{1:k-1})} \\ &\propto \mathbb{P}(r_k, x_k^{(i)}, y_k|x_{0:k-1}^{(i)}, y_{1:k-1}). \end{aligned} \tag{2.33}$$

For simplicity, we define a mid quantity γ_k,

$$\begin{aligned}\gamma_k^{(i)}(r_k) &\triangleq p(r_k, x_k^{(i)}, y_k|x_{0:k-1}^{(i)}, y_{1:k-1}) \\ &= g_{r_k}(y_k|x_k^{(i)}) f(x_k^{(i)}|x_{k-1}^{(i)}) \alpha_{k|k-1}^{(i)}(r_k). \end{aligned} \tag{2.34}$$

Using γ_k we can compute $\alpha_k(\cdot)$,

$$\alpha_k^{(i)}(\ell) \propto \gamma_k^{(i)}(\ell), \tag{2.35a}$$

$$\alpha_k^{(i)}(\ell) = \frac{\gamma_k^{(i)}(\ell)}{\sum_{m=1}^{M} \gamma_k^{(i)}(m)}. \tag{2.35b}$$

The importance weights are updated with,

$$w_k^{(i)} \propto \frac{p(x_k^{(i)}, y_k|x_{0:k-1}^{(i)}, y_{1:k-1})}{q(x_k^{(i)}|x_{0:k-1}^{(i)}, y_{1:k})} w_{k-1}^{(i)}, \tag{2.36}$$

where the numerator of (2.36) is the marginalization of (2.34) over r_k which is equal to $\sum_{m=1}^{M} \gamma_k^{(i)}(m)$. Substituting this expression in the numerator of (2.36) results in,

$$w_k^{(i)} \propto \frac{\sum_{m=1}^{M} \gamma_k^{(i)}(m)}{q(x_k^{(i)}|x_{0:k-1}^{(i)}, y_{1:k})} w_{k-1}^{(i)}. \tag{2.37}$$

Lastly, we perform resampling, if necessary.

The DMPF is summarized in Algorithm 2.4.

2.5 Conclusion

We have provided a general framework of particle filtering. Popular approaches in PF, such as bootstrap proposal density, optimal proposal density, and marginalization idea are introduced. Multiple switching dynamic models are defined

Algorithm 2.4: Summary of the DMPF

Initialization at $k = 0$:
for all $i = 1, \ldots, N$ **do**
  Sample the continuous state $x_0^{(i)} \sim p_0(x_0)$
  Set initial weights $w_0^{(i)} = \frac{1}{N}$
end for
Iterations:
for $k = 1, \ldots, T$ **do**
  for all $i = 1, \ldots, N$ **do**
    Compute $\{\alpha_{k|k-1}^{(i)}(\ell)\}_{\ell=1}^{M}$ according to (2.30)
    Sample the continuous state, $x_k^{(i)} \sim q_k(x_k|x_{0:k-1}^{(i)}, y_{1:k})$
    Compute $\{\gamma_k^{(i)}(\ell)\}_{\ell=1}^{M}$ according to (2.34)
    Compute $\{\alpha_{k|k}^{(i)}(\ell)\}_{\ell=1}^{M}$ according to (2.35)
    Update the weights according to (2.37)
  end for
  Normalize the weights
  Resample, if necessary
end for

where two different marginalization approaches are possible. This model and the particle filtering approaches will be used in a multi ellipse ETT problem whose details will be provided in the next chapter.

CHAPTER 3

NON-ELLIPSOIDAL EXTENDED TARGET TRACKING WITH KNOWN EXTENT

In an ETT application, there are two nested problems to be solved: estimation of the target extent; and tracking the kinematic state of the target. In this chapter, we focus on the latter problem assuming that the multi-ellipse target extent is known. Under this assumption, different marginalization strategies are investigated to solve the association problem between the measurements and ellipses. The performances of the algorithms are tested in simulations and the results are compared to present the advantages and drawbacks of the algorithms.

3.1 Target Extent Model

Here, the target extent is represented with multiple ellipses where each ellipse is called as a sub-object. Various complex shapes can be represented by this model as shown in Figure 3.1. Each sub-object is associated with a known symmetric positive definite (SPD) matrix $\{X^\ell\}_{\ell=1}^M \in \mathbb{R}^{d\times d}$ and a known mean vector $\{\mu^\ell\}_{\ell=1}^M \in \mathbb{R}^d$ where M is the number of sub-objects and d is the dimension of the extent. Here, μ^1 is assumed to be zero to define the first sub-object as the main body of the target.

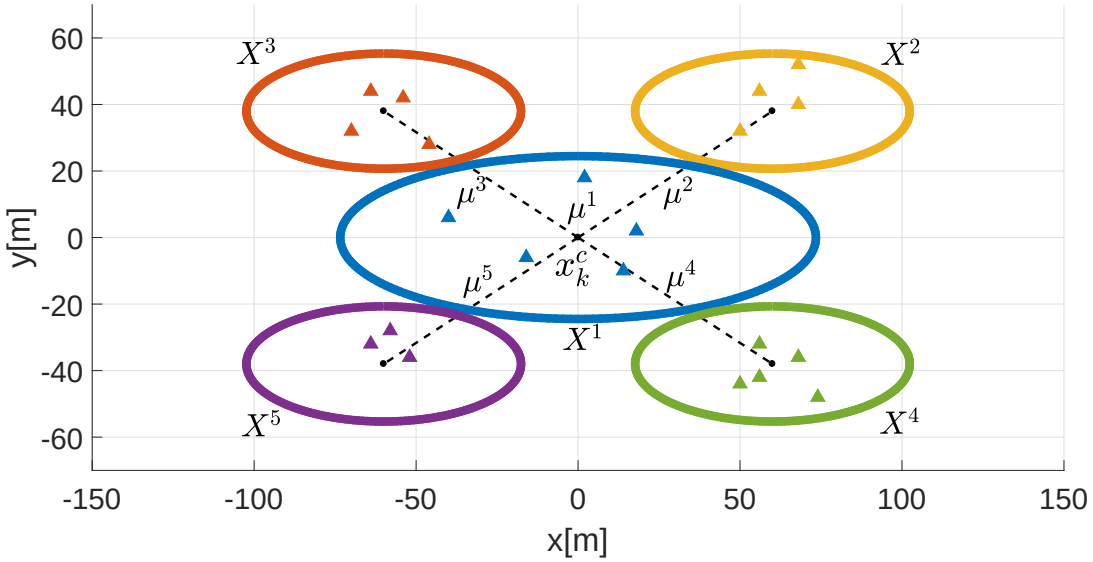

Figure 3.1: An example of target extent representation with 5 ellipses under the assumption of known extent.

Let $\mathbf{Y}_k = \{y_k^j\}_{j=1}^{m_k} \in \mathbb{R}^{n_y}$ be the set of m_k random measurements collected from the target extent at time k. The measurement equation corresponding to the extent model can be written as a mixture of Gaussians under additive Gaussian measurement noise assumption as

$$p(y_k^j|x_k^c) = \sum_{\ell=1}^{M} \pi_\ell \mathcal{N}(y_k^j; x_k^c + \mu^\ell, sX^\ell + R), \tag{3.1}$$

where $\pi_\ell \in [0,1]$ are the mixture weights, i.e., the prior probability that a measurement belongs to ℓ^{th} sub-object. $s \in \mathbb{R}$ is the positive constant scaling factor which is added to spread contribution of the object extension [9]. In some practical applications, it could be more realistic assumption to model the measurement sources as uniformly distributed on the object extent. In that case, the scaling factor is used to approximate the uniform distribution using a Gaussian distribution [11].

The state vector x_k holding the relevant variables of the model can be defined as follows,

$$x_k \triangleq [(x_k^c)^T \ (\tilde{x}_k)^T]^T, \tag{3.2}$$

where x_k^c is the position of the centroid and $\tilde{x}_k$ denotes the additional state variables.

In the Bayesian framework, one can define appropriate priors to compute the posterior density of unknown variables. Our aim is to find a recursive update for the posterior density

$$p(x_k|\mathbf{Y}_{1:k}), \tag{3.3}$$

where this posterior density is intractable since we do not know the true associations between the measurements and sub-objects.

To solve this assignment problem, we first define an association variable r_k^j for each measurement y_k^j. r_k^j indicates the index of the sub-object that the measurement y_k^j belongs and it is assumed to be multinomial distributed as described in Section 2.4.1.

For the target extent model described so far, we will use particle filtering methods to approximate the posterior density of the target kinematic state and the association variables.

3.2 Inference

Under the assumption of known extent matrices $X^{1:M}$ and mean vectors $\mu^{1:M}$, the system can be represented by a multinomial CLGS model. In this case, our aim is to approximate the joint posterior density of the association variables and the kinematic state recursively,

$$p(x_k, \mathbf{r}_k|\mathbf{Y}_{1:k}), \tag{3.4}$$

where $\mathbf{r}_k = [r_k^1 \dots r_k^{m_k}]^T$. The system equations are given as follows,

$$x_{k+1} = Fx_k + e_k, \tag{3.5a}$$

$$y_k^j = Hx_k + v_k^{r_k^j}, \tag{3.5b}$$

where the process noise and the conditional measurement noise can be expressed respectively as

$$e_k \sim \mathcal{N}(0, Q), \tag{3.6a}$$

$$v_k^{r_k^j} \sim \mathcal{N}(\mu^{r_k^j}, sX^{r_k^j} + R). \tag{3.6b}$$

The posterior density 3.4 will be approximated using the system definition provided in 3.5 and different particle filtering approaches.

3.2.1 Adaptation of the Algorithms for Multi-Measurement Systems

The particle filtering approaches introduced so far are derived for a single measurement system. Therefore, we need to perform modifications in their measurement update steps to adapt them into a multi-measurement system.

The first algorithm that we will consider is the BPF. The general flow of the BPF in a multi-measurement multinomial CLGS can be described as follows.

For each time instant k, the association variables $\mathbf{r}_k$ and the kinematic state should be sampled from their prior densities,

$$x_k^{(i)} \sim p(x_k|x_{k-1}^{(i)}), \tag{3.7a}$$

$$\mathbf{r}_k^{(i)} \sim \pi. \tag{3.7b}$$

To update the importance weights, the likelihood of the multiple measurements are calculated as

$$p(\mathbf{Y}_k|x_k^{(i)}, \mathbf{r}_k^{(i)}) = \prod_{j=1}^{m_k} p(y_k^j|x_k^{(i)}, r_k^{(i),j}), \tag{3.8}$$

where we assume that each measurement obtained at time k is conditionally independent of the others. Using the likelihood above, the weights are updated as follows,

$$w_k^{(i)} \propto p(\mathbf{Y}_k|x_k^{(i)}, \mathbf{r}_k^{(i)}) w_{k-1}^{(i)}. \tag{3.9}$$

After calculating the weights, resampling is performed, if necessary. The summary of the BPF is given in Algorithm 3.1.

Algorithm 3.1: Summary of the BPF for multi-measurement case

1: Initialization at $k = 0$:
2: **for all particles** $i = 1, \ldots, N$ **do**
3: Sample $x_0^{(i)} \sim p_0$
4: Set initial weights $w_0^{(i)} = \frac{1}{N}$
5: **end for**
6: Iterations:
7: **for time** $k = 1, \ldots, T$ **do**
8: **for all particles** $i = 1, \ldots, N$ **do**
9: Sample the kinematic state, $x_k^{(i)} \sim p(x_k|x_{k-1}^{(i)})$
10: Sample the association variables, $\mathbf{r}_k^{(i)} \sim \pi$
11: Update the weights according to (3.9)
12: **end for**
13: Normalize the weights
14: Resample, if necessary
15: **end for**

The second algorithm that we will consider is the CMPF. CMPF factorizes the posterior density into,

$$p(x_k, \mathbf{r}_{1:k}|\mathbf{Y}_{1:k}) = p(x_k|\mathbf{r}_{1:k}, \mathbf{Y}_{1:k})p(\mathbf{r}_{1:k}|\mathbf{Y}_{1:k}), \tag{3.10}$$

where $p(x_k|\mathbf{r}_{1:k}, \mathbf{Y}_{1:k})$ can be expressed as a conditional Gaussian density. The general flow of the CMPF in a multi-measurement multinomial CLGS can be described as follows.

For each time instant k, the association variables $\mathbf{r}_k$ are sampled from the importance density,

$$\mathbf{r}_k^{(i)} \sim q(\cdot). \tag{3.11}$$

Here, if we want to use the bootstrap proposal density, the importance density $q(\cdot)$ is chosen to be equal to the prior density π. Alternatively, one can use the optimal proposal density to increase the performance as described in Section 2.3. The optimal proposal density is given by

$$\pi_{opt}^{(i),j} \triangleq p(r_k^{(i),j}|y_{1:k}^j, r_{k-1}^{(i),j}) \propto p(y_k^j|r_k^{(i),j}, y_{1:k-1}^j)p(r_k^{(i),j}|r_{k-1}^{(i),j}), \tag{3.12}$$

where the predictive likelihood $p(y_k^j|r_k^{(i),j}, y_{1:k-1}^j)$ is expressed further as

$$\begin{aligned} p(y_k^j|r_k^{(i),j}, y_{1:k-1}^j) &= \int p(y_k^j|x_k^{(i)}, r_k^{(i),j}, y_{1:k-1}^j) p(x_k^{(i)}|r_k^{(i),j}, y_{1:k-1}^j) dx_k \\ &= \int \mathcal{N}(y_k^j; Hx_k^{(i)} + \mu^{r_k^j}, R + sX^{r_k^j}) \mathcal{N}(x_k^{(i)}; x_{k|k-1}^{(i)}, P_{k|k-1}^{(i)}) dx_k \\ &= \mathcal{N}(y_k^j; H\hat{x}_{k|k-1}^{(i)} + \mu^{r_k^j}, HP_{k|k-1}^{(i)} H^T + R + sX^{r_k^j}). \end{aligned} \tag{3.13}$$

The optimal proposal density is calculated using the following expressions for each sub-object ℓ,

$$\widetilde{\pi}_{opt,\ell}^{(i),j} = \mathcal{N}\big(y_k^j; H\hat{x}_{k|k-1}^{(i)} + \mu^\ell, HP_{k|k-1}^{(i)} H^T + sX^\ell + R\big)\pi_\ell, \tag{3.14a}$$

$$\pi_{opt,\ell}^{(i),j} = \frac{\widetilde{\pi}_{opt,\ell}^{(i),j}}{\sum_{n=1}^{M} \widetilde{\pi}_{opt,n}^{(i),j}}, \tag{3.14b}$$

where $\hat{x}_{k|k-1}^{(i)}$ and $P_{k|k-1}^{(i)}$ are calculated in the time update step. Next, the particle weights are updated according to the following equation,

$$w_k^{(i)} \propto \prod_{j=1}^{m_k} \frac{p(y_k^j|r_k^{(i),j}, y_{1:k-1}^j)\pi_{r_k^{(i),j}}}{q(r_k^{(i),j})} w_{k-1}^{(i)}. \tag{3.15}$$

Time update of the continuous state is not affected by multiple measurements.

Measurement update of the continuous state is performed consecutively for each measurement y_k^j,

$$\hat{x}_k^{(i)} = \hat{x}_{k|k-1}^{(i)} + K_k(y_k^j - H\hat{x}_{k|k-1}^{(i)} - \mu^{r_k^j}), \tag{3.16a}$$

$$P_k^{(i)} = (I - K_k H)P_{k|k-1}^{(i)}, \tag{3.16b}$$

$$K_k = P_{k|k-1}^{(i)} H^T (HP_{k|k-1}^{(i)} H^T + sX^{r_k^j} + R)^{-1}, \tag{3.16c}$$

where we assign $\hat{x}_{k|k-1}^{(i)} \leftarrow \hat{x}_k^{(i)}$, $P_{k|k-1}^{(i)} \leftarrow P_k^{(i)}$ and iterate for $j = 1, \ldots, m_k$.

As a last step, we perform resampling, if necessary. The method is summarized in Algorithm 3.2.

Algorithm 3.2: Summary of the CMPF for multi-measurement case

Initialization at $k = 0$:

for all particles $i = 1, \ldots, N$ **do**

Set initial weights $w_0^{(i)} = \frac{1}{N}$

Set initial kinematic state of each particle $\hat{x}_0^{(i)} = \bar{x}_0$, $\hat{P}_0^{(i)} = \bar{P}_0$

end for

Iterations:

for time $k = 1, \ldots, T$ **do**

for all particles $i = 1, \ldots, N$ **do**

Perform time update of the kinematic state

Sample the association variables, $\mathbf{r}_k^{(i)} \sim q(\cdot)$

Update the weights according to (3.15)

end for

Normalize the weights

for all particles $i = 1, \ldots, N$ **do**

for all measurements $j = 1, \ldots, m_k$ **do**

Perform measurement update of the kinematic states according to (3.16)

Assign $\hat{x}_{k|k-1}^{(i)} \leftarrow \hat{x}_k^{(i)}$, $P_{k|k-1}^{(i)} \leftarrow P_k^{(i)}$ and iterate

end for

end for

Resample, if necessary.

end for

The third algorithm that we will consider is the DMPF. DMPF factorizes the posterior density $p(x_{0:k}, \mathbf{r}_k|\mathbf{Y}_{1:k})$ as follows,

$$p(x_{0:k}, \mathbf{r}_k|\mathbf{Y}_{1:k}) = p(\mathbf{r}_k|x_{0:k}, \mathbf{Y}_{1:k})p(x_{0:k}|\mathbf{Y}_{1:k}). \tag{3.17}$$

At any time k, the mode probability variable $\alpha(\cdot)_{k|k-1}^{(i),j}$ should be calculated for each measurement for a given time. In the measurement update step, γ and α

variables are updated consecutively,

$$\gamma_k^{(i),j}(\ell) = g_\ell(y_k^j|x_k^{(i)}) f(x_k|x_{k-1}^{(i)}) \alpha_{k|k-1}^{(i),j}(\ell), \tag{3.18a}$$

$$\alpha_{k|k}^{(i),j}(\ell) = \frac{\gamma_k^{(i),j}(\ell)}{\sum_{m=1}^{M} \gamma_k^{(i),j}(m)}. \tag{3.18b}$$

Assuming that the bootstrap proposal density is used, the weights are updated using the following expressions,

$$w_k^{(i)} \propto \frac{\sum_{\ell=1}^{M} \prod_{j=1}^{m_k} \gamma_k^{(i),j}(\ell)}{p(x_k|x_{0:k-1}^{(i)}, \mathbf{Y}_{1:k})} w_{k-1}^{(i)}. \tag{3.19}$$

As a last step, we perform resampling, if necessary. The method is summarized in Algorithm 3.3.

Algorithm 3.3: Summary of the DMPF for multi-measurement case

Initialization at $k = 0$:
for all particles $i = 1, \ldots, N$ **do**
  Sample the initial kinematic state $x_0^{(i)} \sim p_0$
  Set initial weights $w_0^{(i)} = \frac{1}{N}$
end for
Iterations:
for time $k = 1, \ldots, T$ **do**
  for all particles $i = 1, \ldots, N$ **do**
    for all measurements $j = 1, \ldots, m_k$ **do**
      Compute $\alpha_{k|k-1}^{(i),j}(\cdot)$ according to (2.30)
    end for
    Sample the kinematic state, $x_k^{(i)} \sim p(x_k | x_{0:k-1}^{(i)}, \mathbf{Y}_{1:k})$
    for all measurements $j = 1, \ldots, m_k$ **do**
      Compute $\gamma_k^{(i),j}(\cdot)$ according to (3.18a)
      Compute $\alpha_{k|k}^{(i),j}(\cdot)$ according to (3.18b)
    end for
    Update the weights according to (3.19)
  end for
  Normalize the weights
  Resample, if necessary
end for

3.3 Performance Evaluation

We test the algorithms on a simulation of a moving multi-ellipse object which consists of 3-ellipses (see Figure 3.2). The object follows a linear path and the number of measurements at each scan is Poisson distributed with an average of 7 measurements. Constant velocity model is used as the kinematic model of the object which will be explained in the following subsection.

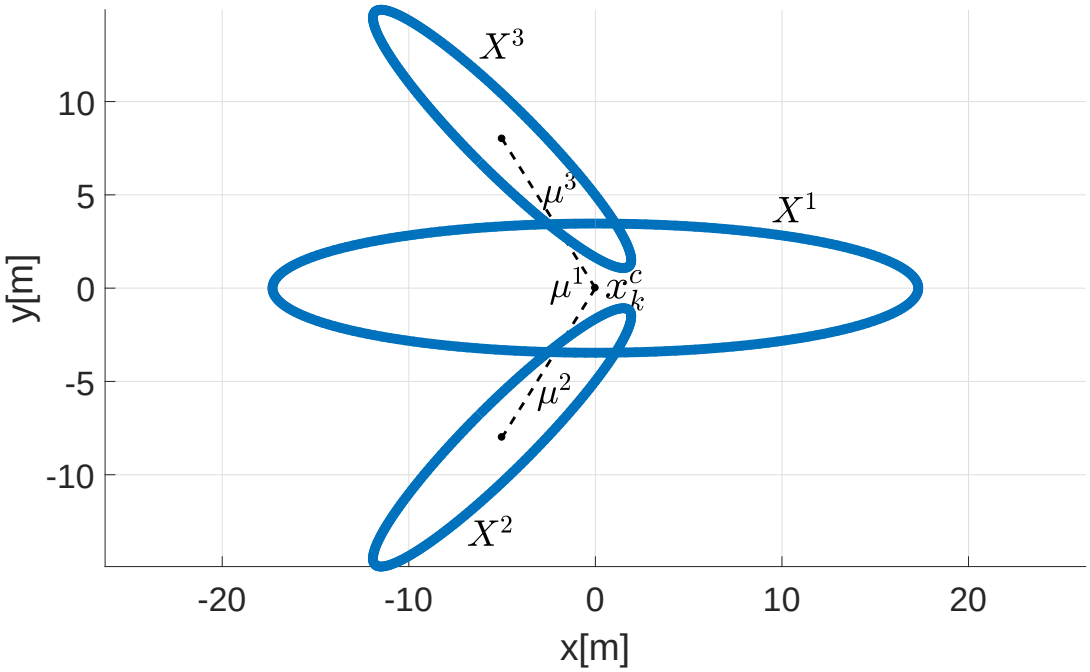

Figure 3.2: Target extent model with 3 ellipses.

3.3.1 2-D Constant Velocity Model

Non-maneuvering 2-dimensional constant velocity model is one of the most commonly used models in target tracking problems [26]. The target speed is assumed to be constant through a linear path and the system noise accounts for possible accelerations that disturbs the speed. The state consists of two dimensional positions and velocities

$$x = \begin{bmatrix} \mathrm{x} \\ \mathrm{y} \\ \mathrm{v_x} \\ \mathrm{v_y} \end{bmatrix}. \tag{3.20}$$

The system matrices in (3.5) are defined as

$$F = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}. \tag{3.21}$$

The process noise covariance matrix Q is chosen as

$$Q = \sigma^2 \begin{bmatrix} T^3/3 & 0 & T^2/2 & 0 \\ 0 & T^3/3 & 0 & T^2/2 \\ T^2/2 & 0 & T & 0 \\ 0 & T^2/2 & 0 & T \end{bmatrix},$$

where T is the sampling time. The sampling time is set to $T = 0.1s$ for the rest of the simulations performed in this study.

The above-defined system is the basic 2-D non-maneuvering constant velocity model which can be extended by including other kinematic parameters such as heading, jerk or acceleration.

3.3.2 Simulation

The system is simulated in 100 Monte Carlo (MC) runs using the same measurement realization to observe estimation consistency and variance. The numeric results presented here is an average of the MC runs.

As a performance metric, RMSE (2.8) will be used. In addition to RMSE, computation time of the algorithms will be compared. The abbreviations of the methods used in simulations are as follows: BPF, CMPF-B (CMPF with bootstrap proposal density), CMPF-O (CMPF with optimal proposal density) and DMPF.

Since the BPF samples both the association variables $\mathbf{r}_k$ and the kinematic state x_k, it requires an excessive number of particles to provide satisfying results when compared with the other two algorithms. Therefore, the BPF is run with $N = 20\,000$ particles while the CMPF-B, the CMPF-O, and the DMPF are run with 200 particles.

The mode probability vector π is chosen as in equation (2.19) which is a reasonable choice when the measurement rates of the sub-objects are unknown. The sampling time is set to $T = 1$s, the system noise variance is set to $\sigma^2 = 1$ and the measurement noise covariance is selected to be $R = \text{diag}([10, 10])\text{m}^2$. The scaling factor is taken as $s = 1$. Initial kinematics and their covariance matrix are

chosen as $x_0 = [0, 0, 10\text{m/s}, 0]^T$, $P_0 = \text{diag}[400, 400, 100, 100]$ respectively. All simulations are run in Matlab$^{(R)}$ R2016b on a standard laptop with an Intel$^{(R)}$ Core$^{(TM)}$ i5-7200U 2.50 GHz platform with 8 GB of RAM running Windows.

The position estimates of all four algorithms are presented in Figures 3.3, 3.4, 3.5 and 3.6.

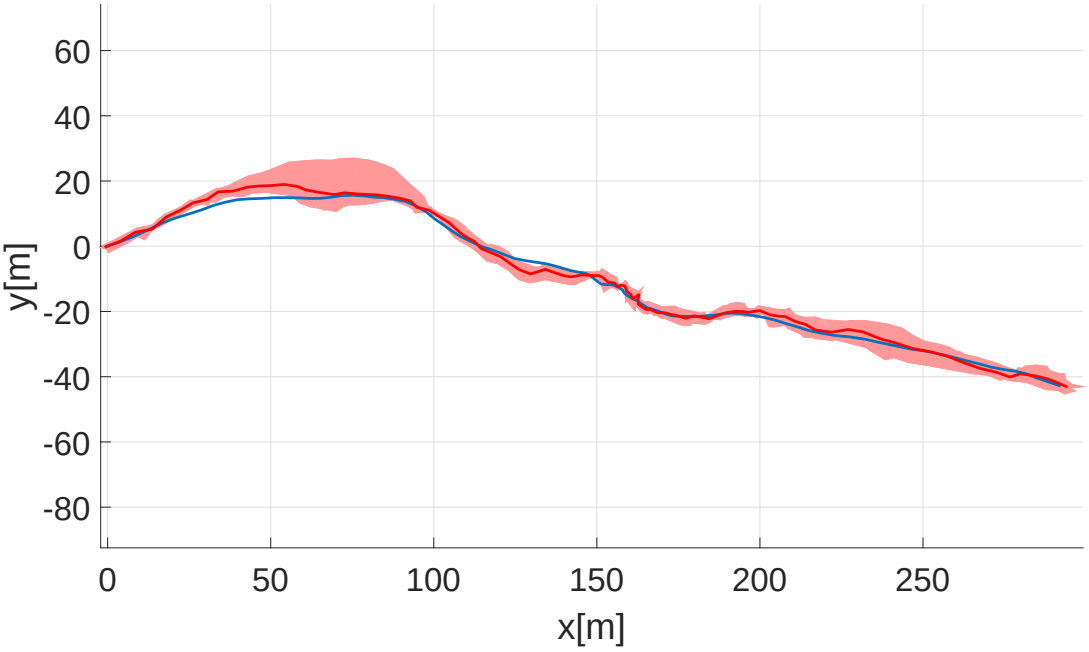

Figure 3.3: Position estimates of the BPF for 100 MC runs with 20 000 particles. The red and blue lines show the average MC run result and the true value respectively. The transparent area shows the upper and lower bound of the MC estimates.

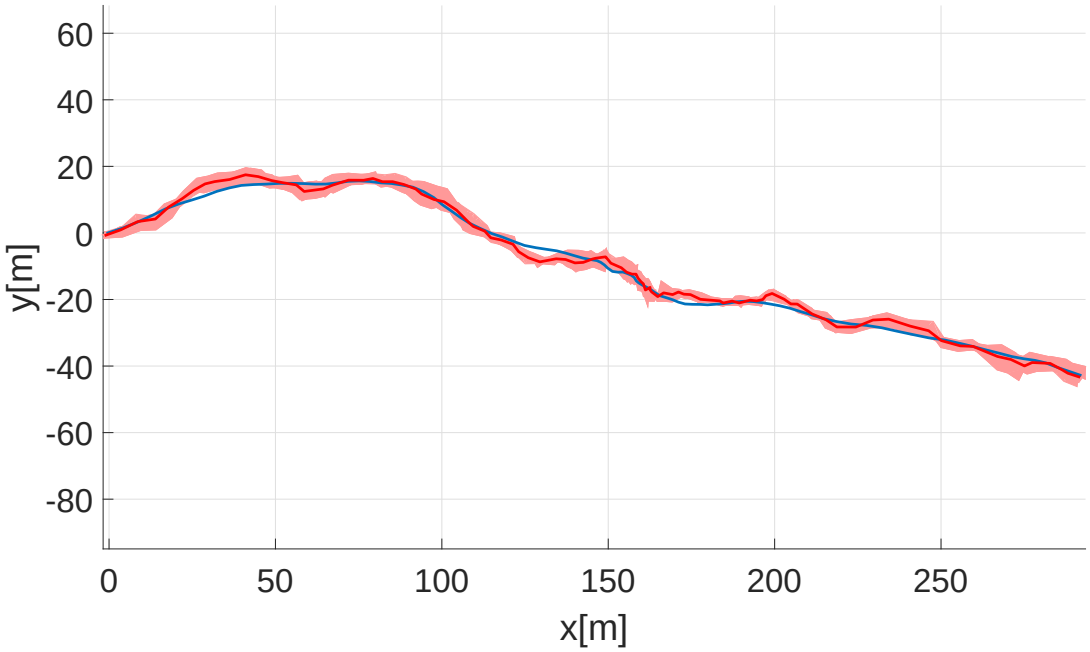

Figure 3.4: Position estimates of the CMPF-B for 100 MC runs with 200 particles. The red and blue lines represent the average MC run result and the true value respectively. The transparent area shows the upper and lower bound of the MC estimates.

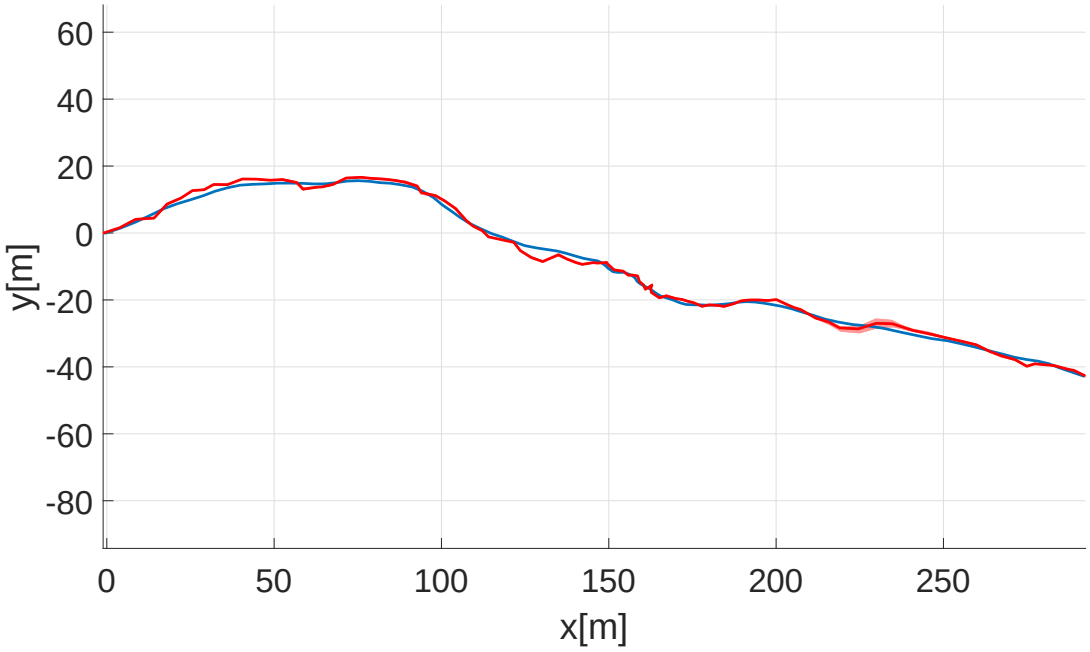

Figure 3.5: Position estimates of the CMPF-O for 100 MC runs with 200 particles. The red and blue lines represent the average MC run result and the true value respectively. The transparent area shows the upper and lower bound of the MC estimates.

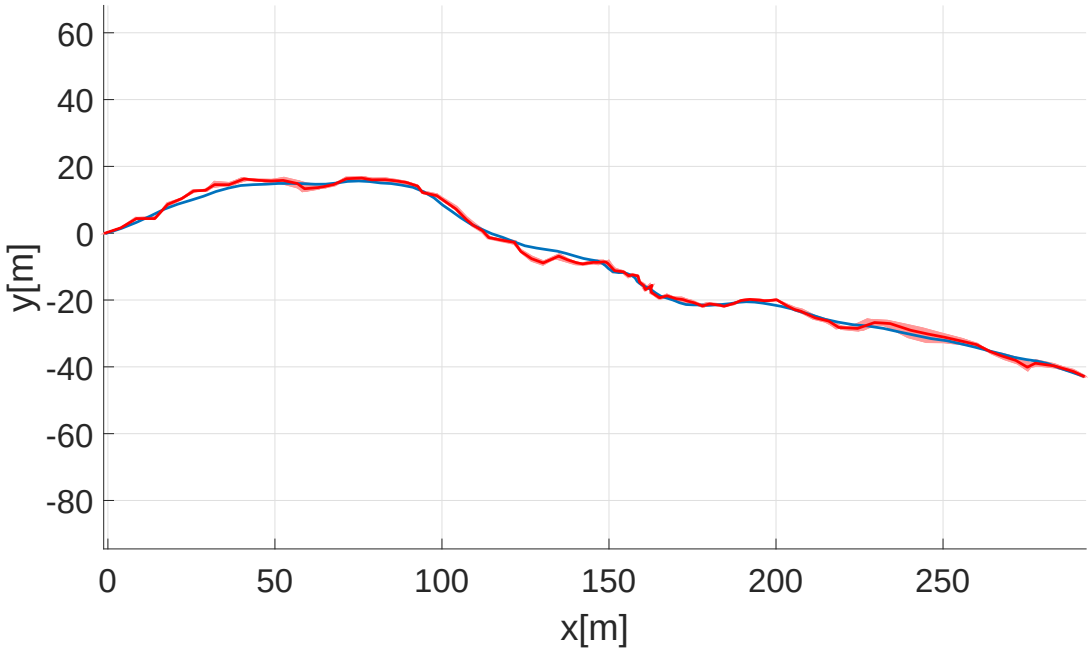

Figure 3.6: Position estimates of the DMPF for 100 MC runs with 200 particles. The red and blue lines represent the average MC run result and the true value respectively. The transparent area shows the upper and lower bound of the MC estimates.

The average RMSE values for the position estimates are presented in Table 3.1.

Table 3.1: RMSE values of the four algorithms

		RMSE[m]
BPF	(with 20K particles)	2.91
CMPF-B	(with 200 particles)	1.59
CMPF-O	(with 200 particles)	1.33
DMPF	(with 200 particles)	1.42

The BPF has the largest MC variance and RMSE value even if it runs with 100 times more particles than the other algorithms. Therefore, one can claim that using marginalization (if it is possible) has a positive impact on the performance of the algorithm.

We obtained the lowest MC variance and RMSE value with the CMPF-O.

The DMPF has the closest estimation performance to the CMPF-O in the sense of RMSE and MC variance.

One can also observe the effect of using the optimal proposal density instead of the bootstrap proposal density by comparing the results of the CMPF-B and the CMPF-O. The optimal proposal density has a positive impact on the performance of the algorithm.

A second simulation is performed to observe the relationship between the number of particles (N) and RMSE values. In the simulation, all algorithms are run with $N = 200, 500, 1000$ particles. The results are presented in Figure 3.7. RMSE of the CMPF-B, the CMPF-O, and the DMPF converge to a small number as the number of particles increases.

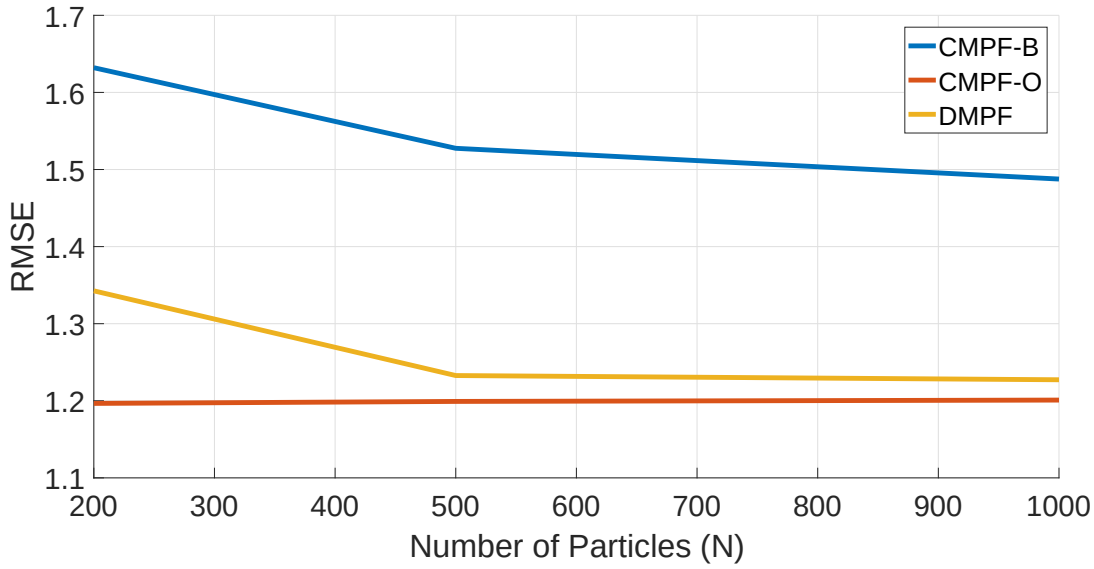

Figure 3.7: RMSE results versus the number of particles of the three algorithms. The blue line denotes the CMPF-B, the orange line denotes the CMPF-O and the yellow line denotes the DMPF.

Table 3.2: Average computation times per update of the algorithms

		Computation Time[s]
BPF	(with 20K particles)	0.089
CMPF-B	(with 200 particles)	0.31
CMPF-O	(with 200 particles)	0.32
DMPF	(with 200 particles)	0.0019

Although RMSE values and the figures state that the CMPF-O has the best estimation results among the others, it has one drawback. Since the algorithm includes optimal proposal density computations and KF update equations performed per particle, it requires vast computation effort when compared with the DMPF. Average computation times per update are given in Table 3.2. The DMPF has the smallest computation time since its implementation does not include any complex equations.

3.4 Conclusion

We have investigated possible marginalization strategies to find an efficient particle filter for a simplified ETT problem where the extent parameters are assumed to be known. Four different PFs are implemented and their performances are compared in the simulations. Overall, the best estimation performance is obtained by using the CMPF-O. However, the CMPF-O requires significant computational power. The DMPF has a close estimation performance to the CMPF-O and it requires far less computational power. Therefore, using the DMPF can be beneficial in practical applications.

CHAPTER 4

NON-ELLIPSOIDAL EXTENDED TARGET TRACKING WITH UNKNOWN EXTENT

Under the assumption of known extent, the problem of kinematic state estimation in ETT is addressed using different PF techniques in the previous chapter. In this chapter, we focus on the problem of joint estimation of the kinematic and the extent states. The multi-ellipse approach will be used to model the target extent as described in the previous chapter. A marginalized particle filtering method which is based on variational Bayes is proposed. The method finds the required analytical conditional expressions approximately by minimizing the KL divergence between the true and the approximate densities. These approximate analytic expressions are used in particle filtering to perform marginalization.

4.1 Target Extent Model

Here, we will use the same target extent model as in 3.1 except that the extent matrices, $\{X_k^\ell\}_{\ell=1}^M$ and mean vectors, $\{\mu_k^\ell\}_{\ell=1}^M$ are unknown. In order to make the target centroid x_k^c observable, one of the mean vectors should be assumed to be known. μ_k^1 is assumed to be zero to define the first sub-object as the main body of the target (see: Figure 4.1). The measurement equation in (3.1) can be re-expressed as

$$p(y_k^j|x_k^c, X_k^{1:M}, \mu_k^{1:M}) = \sum_{\ell=1}^{M} \pi_\ell \mathcal{N}(y_k^j; x_k^c + \mu_k^\ell, sX_k^\ell + R). \tag{4.1}$$

The augmented state vector which holds the relevant variables can be defined as

$$x_k \triangleq [(x_k^c)^T (\mu_k^1)^T \ \ldots \ (\mu_k^M)^T \ (\tilde{x}_k)^T]^T, \tag{4.2}$$

where $\mu_k^{1:M}$ denotes the mean vector (see: Figure 4.1). Furthermore, the posterior density should also be redefined so that it covers the extent state,

$$p(x_k, X_k^{1:M} | \mathbf{Y}_{1:k}). \tag{4.3}$$

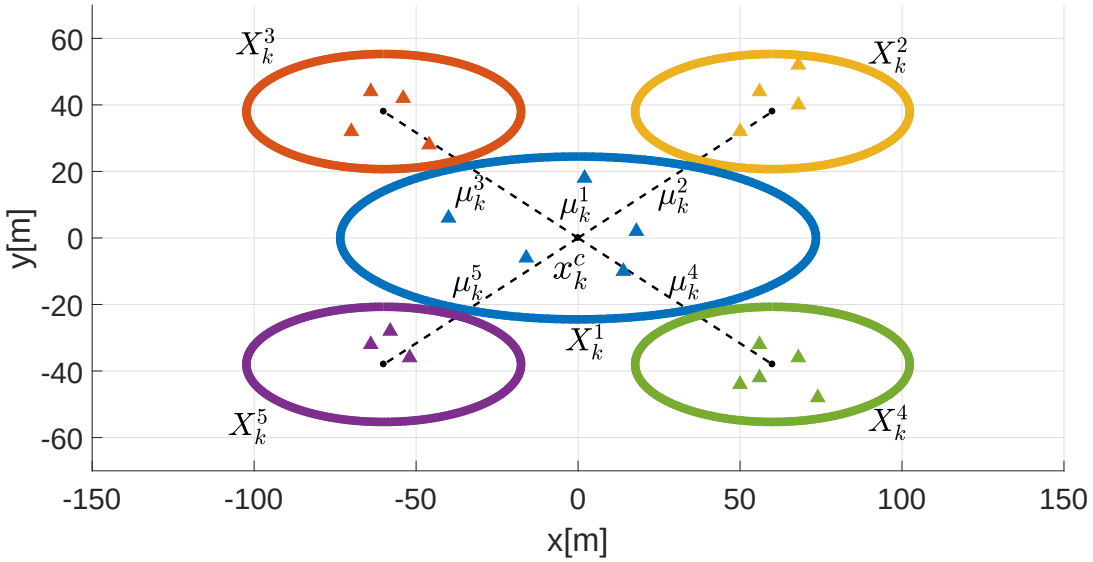

Figure 4.1: An example of target extent representation with 5 ellipses under the assumption of unknown extent.

However, this posterior density is intractable because of two reasons:

1) Given a set of measurements, we do not know which measurement belongs to which sub-object. This creates a combinatorial problem which grows exponentially with time k.

2) Assume that, we know the correct associations between the measurements and the sub-objects; it is still not possible to find a closed form expression for the posterior because of the additive noise covariance term R in (4.1), which violates the conjugacy between the prior and the likelihood (see: [27, Section -II]).

We will solve the first problem using SMC techniques in a marginalized fashion. The second problem will be solved by using variational inference.

We use the association variables $\mathbf{r}_{1:k}$ in order to indicate the assignments between measurements and sub-objects. Therefore, we include $\mathbf{r}_{1:k}$ to the posterior density, $p(x_k, X_k^{1:M}, \mathbf{r}_{1:k}|\mathbf{Y}_{1:k})$.

Consider the following factorization of the posterior density

$$p(x_k, X_k^{1:M}, \mathbf{r}_{1:k}|\mathbf{Y}_{1:k}) = p(x_k, X_k^{1:M}|\mathbf{r}_{1:k}, \mathbf{Y}_{1:k})p(\mathbf{r}_{1:k}|\mathbf{Y}_{1:k}). \tag{4.4}$$

It is possible to approximate the first factor as an analytical expression, and the second factor by a set of weighted samples,

$$p(\mathbf{r}_{1:k}|\mathbf{Y}_{1:k}) \approx \sum_{i=1}^{N} w_{0:k}^{(i)} \delta_{\mathbf{r}_{1:k}^{(i)}}(\mathbf{r}_{1:k}). \tag{4.5}$$

A summary of a generic MPF, where discrete states are sampled and the posterior density of continuous states are analytically expressed, is given in Algorithm 4.1.

In the following section, we will provide the details of the measurement update of the continuous state (line:13, in Algorithm 4.1). The time update of the continuous state, sampling and weight update steps (lines:[10-12], in Algorithm 4.1) will be described in Section 4.3.

4.2 Measurement Update

Suppose at time k, the predicted density for the kinematic and the extent states are given as follows,

$$\begin{aligned} &p(x_k^{(i)}, X_k^{1:M,(i)}|\mathbf{r}_{1:k-1}^{(i)}, \mathbf{Y}_{1:k-1}) \\ &\qquad = \mathcal{N}(x_k^{(i)}; \hat{x}_{k|k-1}^{(i)}, P_{k|k-1}^{(i)}) \prod_{\ell=1}^{M} \mathcal{IW}(X_k^{\ell,(i)}; v_{k|k-1}^{\ell,(i)}, V_{k|k-1}^{\ell,(i)}), \end{aligned} \tag{4.6}$$

where $\hat{x}_{k|k-1}^{(i)}$ and $P_{k|k-1}^{(i)}$ are the mean and the covariance of the Gaussian state vector $x_k^{(i)}$ of the i^{th} particle, respectively. The variables $v_{k|k-1}^{(i)}$ and $V_{k|k-1}^{(i)}$ are

Algorithm 4.1: Summary of the MPF

Initialization at time $k = 0$:
for all particles $i = 1, \ldots, N$ **do**
  Set initial weights $w_0^{(i)} = \frac{1}{N}$
  Set initial continuous state statistics of each particle
end for
Iterations:
for time $k = 1, \ldots, T$ **do**
  for all particles $i = 1, \ldots, N$ **do**
    Perform time update of the continuous states
    Sample discrete states $\mathbf{r}_k^{(i)} \sim q(\mathbf{r}_k^{(i)} | \mathbf{r}_{1:k-1}^{(i)}, \mathbf{Y}_{1:k})$
    Compute the likelihood
    Update the weights $w_k^{(i)}$
    Perform measurement update of the continuous states
  end for
  Normalize the weights,
  Resample, if necessary
end for

the degrees of freedom variable and the scale matrix of the inverse Wishart distributed extent state $X_k^{(i)}$ of the i^{th} particle. The left-hand side of the (4.6) is conditioned on the association variables and measurements up to time instant $k-1$. For notational simplicity, the superscript (i) will be dropped for the remaining of this section.

The new $\mathbf{r}_k$ samples are generated after the measurements $\mathbf{Y}_k$ at time k are available. Then, the joint posterior density of the kinematic and the extent states can be computed recursively using Bayes' formula,

$$p(x_k, X_k^{1:M}|\mathbf{r}_{1:k}, \mathbf{Y}_{1:k}) = \frac{p(\mathbf{Y}_k|x_k, X_k^{1:M}, \mathbf{r}_{1:k})p(x_k, X_k^{1:M}|\mathbf{r}_{1:k-1}, \mathbf{Y}_{1:k-1})}{p(\mathbf{Y}_k|\mathbf{Y}_{1:k-1})}. \quad (4.7)$$

The measurement likelihood at time k can be expressed as follows,

$$\begin{aligned} p(\mathbf{Y}_k|x_k, X_k^{1:M}, \mathbf{r}_{1:k}) &= \prod_{j=1}^{m_k} p(y_k^j|x_k, X_k^{r_k^j}, r_{1:k}^j) \\ &= \prod_{j=1}^{m_k} \mathcal{N}(y_k^j; H^{r_k^j}x_k, sX_k^{r_k^j} + R), \end{aligned} \quad (4.8)$$

where the measurements are assumed to be conditionally independent of the others. The expression on the left hand side also can be expressed further as

$$H^{r_k^j}x_k \triangleq x_k^c + \mu_k^{r_k^j}. \quad (4.9)$$

For the likelihood given above, it is not possible to obtain a compact analytical expression for the posterior in (4.7), hence, an exact update is not possible. Note that, the required update can be interpreted as follows: In reference to Figure 4.1, the unknown means μ_k^ℓ's, unknown centroid x_k^c, and unknown extents X_k^ℓ's must be all updated given their last estimates and the measurements $\mathbf{Y}_k$ with their association variables $\mathbf{r}_k$. In the next subsection, we carefully derive an approximate method by making use of indicator functions to update the statistics of each sub-object with the measurements selected by the association variables.

4.2.1 Conditional Variational Inference

Using the variational approximation, one can find an approximate analytical solution for the posterior density [7],

$$p(x_k, X_k^{1:M}|\mathbf{r}_{1:k}, \mathbf{Y}_{1:k}) \approx q_x(x_k, X_k^{1:M}). \tag{4.10}$$

The approximate density $q_x(x_k, X_k^{1:M})$ is chosen to be a product of factorized densities,

$$q_x(x_k, X_k^{1:M}) = q_x(x_k) \prod_{\ell=1}^{M} q_{X^\ell}(X_k^\ell), \tag{4.11}$$

where $q_x(x_k)$ and $q_{X^{1:M}}(X_k^{1:M})$ are the approximate posterior densities for x_k and $X_k^{1:M}$ respectively. An additional instrumental variable is required to be defined to address the problem caused by the additive measurement noise covariance term R. This variable is called as noise-free measurement [27], and denoted by z_k^j and $\mathbf{Z}_k = \{z_k^j\}_{j=1}^{m_k}$. The measurement likelihood can be expressed using $\mathbf{Z}_k$ as

$$\mathcal{N}(y_k^j; H^{r_k^j} x_k, s X_k^{r_k^j} + R) = \int \mathcal{N}(y_k^j; z_k^j, R)\mathcal{N}(z_k^j; H^{r_k^j} x_k, s X_k^{r_k^j}) dz_k^j. \tag{4.12}$$

The equation (4.12) can be interpreted as the marginalization of the following joint density,

$$p(y_k^j, z_k^j | x_k, X_k^{r_k^j}, r_k^j) = \mathcal{N}(y_k^j; z_k^j, R)\mathcal{N}(z_k^j; H^{r_k^j} x_k, s X_k^{r_k^j}), \tag{4.13}$$

over variable z_k^j for each measurement y_k^j. Since we also want to estimate the instrumental variable, $\mathbf{Z}_k$, it should also be included in the posterior,

$$p(x_k, X_k^{1:M}, \mathbf{Z}_k|\mathbf{r}_{1:k}, \mathbf{Y}_{1:k}) \approx q_x(x_k) \prod_{\ell=1}^{M} q_{X^\ell}(X_k^\ell) q_{\mathbf{Z}}(\mathbf{Z}_k), \tag{4.14}$$

where $q_{\mathbf{Z}}(\mathbf{Z}_k)$ is the approximate density of the instrumental variable $\mathbf{Z}_k$. The idea of variational approximation [7, Ch. 10] is to seek factorized densities whose product minimizes the following cost function.

$$\hat{q}_x, \hat{q}_{X^{1:M}}, \hat{q}_{\mathbf{Z}} = \arg \min_{q_x, q_{X^{1:M}}, q_{\mathbf{Z}}} \mathrm{KL}\big(q(x_k, X_k^{1:M}, \mathbf{Z}_k) || p(x_k, X_k^{1:M}, \mathbf{Z}_k|\mathbf{r}_{1:k}, \mathbf{Y}_{1:k})\big). \tag{4.15}$$

where KL divergence is defined as

$$\mathrm{KL}\big(q||p\big) \triangleq \int q \log\left(\frac{q}{p}\right) dx. \tag{4.16}$$

We use fixed-point iterations to solve the problem (4.15). At each iteration, only one factorized density is updated while the rest is kept constant to their last estimated values. The solution of the optimization problem (4.15) is provided as [7, Ch. 10]:

$$\log \hat{q}_x(x_k) = E_{\hat{q}_{X^{1:M}}, \hat{q}_{\mathbf{Z}}}\big[\log p(x_k, X_k^{1:M}, \mathbf{Z}_k, \mathbf{Y}_k | \mathbf{r}_{1:k}, \mathbf{Y}_{1:k-1})\big] + c_x, \tag{4.17a}$$

$$\log \hat{q}_{X^\ell}(X_k^\ell) = E_{\hat{q}_x, \hat{q}_{\mathbf{Z}}, \hat{q}_{X^{-\ell}}}\big[\log p(x_k, X_k^{1:M}, \mathbf{Z}_k, \mathbf{Y}_k | \mathbf{r}_{1:k}, \mathbf{Y}_{1:k-1})\big] + c_{X^\ell}, \tag{4.17b}$$

$$\log \hat{q}_{\mathbf{Z}}(\mathbf{Z}_k) = E_{\hat{q}_x, \hat{q}_{X^{1:M}}}\big[\log p(x_k, X_k^{1:M}, \mathbf{Z}_k, \mathbf{Y}_k | \mathbf{r}_{1:k}, \mathbf{Y}_{1:k-1})\big] + c_{\mathbf{Z}}, \tag{4.17c}$$

where c_x, c_{X^ℓ} and $c_{\mathbf{Z}}$ are the constant terms with respect to the corresponding variables. The joint density $p(x_k, X_k^{1:M}, \mathbf{Z}_k, \mathbf{Y}_k | \mathbf{r}_{1:k}, \mathbf{Y}_{1:k-1})$ in the equation (4.17) can be written explicitly as

$$\begin{aligned}
&p(x_k, X_k^{1:M}, \mathbf{Z}_k, \mathbf{Y}_k | \mathbf{r}_{1:k}, \mathbf{Y}_{1:k-1}) \\
&\quad = p(\mathbf{Y}_k | \mathbf{Z}_k, \mathbf{r}_{1:k}) p(\mathbf{Z}_k | x_k, X_k^{1:M}, \mathbf{r}_{1:k}) \\
&\qquad \times p(x_k, X_k^{1:M} | \mathbf{r}_{1:k}, \mathbf{Y}_{1:k-1}) \\
&\quad = \left(\prod_{j=1}^{m_k} \mathcal{N}(y_k^j; z_k^j, R)\right) \left(\prod_{j=1}^{m_k} \mathcal{N}(z_k^j; H^{r_k^j} x_k, s X_k^{r_k^j})\right) \\
&\qquad \times \mathcal{N}(x_k; \hat{x}_{k|k-1}, P_{k|k-1}) \prod_{\ell=1}^{M} \mathcal{IW}(X_k^\ell; v_{k|k-1}^\ell, V_{k|k-1}^\ell).
\end{aligned} \tag{4.18}$$

The full derivations for the $(\xi+1)^{\text{th}}$ iterations of approximate posteriors will be presented in the following subsections.

4.2.1.1 Determination of $q_x^{(\xi+1)}(\cdot)$

We take expectation of (4.18) with respect to $\mathbf{Z}_k$ and $X_k^{1:M}$. The parts that does not depend on x_k are omitted and written as a constant term c_x.

$$\begin{aligned}
\log q_x^{(\xi+1)}(x_k) = &E_{q_{X^{1:M}}^{(\xi)}, q_{\mathbf{Z}}^{(\xi)}}[\log p(\mathbf{Z}_k | x_k, X_k^{1:M}, \mathbf{r}_{1:k})] \\
&+ \log \mathcal{N}(x_k; \hat{x}_{k|k-1}.P_{k|k-1}) + c_x.
\end{aligned} \tag{4.19}$$

We can further express the expectation term as

$$\begin{aligned}
&\log p(\mathbf{Z}_k|x_k, X_k^{1:M},\mathbf{r}_{1:k}) \\
&\qquad = \log \prod_{j=1}^{m_k} \mathcal{N}(z_k^j; H^{r_k^j} x_k, sX_k^{r_k^j}) \\
&\qquad = \log \prod_{j=1}^{m_k} \left[\frac{1}{(2\pi)^{n/2}|sX_k^{r_k^j}|^{1/2}} e^{-\frac{1}{2}(z_k^j - H^{r_k^j}x_k)^T (sX_k^{r_k^j})^{-1}(z_k^j - H^{r_k^j}x_k)} \right] \\
&\qquad = -\frac{m_k n}{2}\log(2\pi) - \frac{1}{2}\sum_{j=1}^{m_k} \log|sX_k^{r_k^j}| \\
&\qquad\quad - \frac{1}{2}\sum_{j=1}^{m_k} \left[(z_k^j - H^{r_k^j}x_k)^T (sX_k^{r_k^j})^{-1}(z_k^j - H^{r_k^j}x_k) \right], \qquad (4.20)
\end{aligned}$$

which can be rewritten using trace (see Appendix A) as

$$\begin{aligned}
\log p(\mathbf{Z}_k|x_k, X_k^{1:M}, \mathbf{r}_{1:k}) = &-\frac{m_k n}{2}\log(2\pi) - \frac{1}{2}\sum_{j=1}^{m_k} \log|sX_k^{r_k^j}| \\
&- \frac{1}{2}\sum_{j=1}^{m_k} \mathrm{tr}\left[(z^j - H^{r_k^j}x_k)(z^j - H^{r_k^j}x_k)^T (sX_k^{r_k^j})^{-1} \right]. \qquad (4.21)
\end{aligned}$$

Taking expectation of (4.21) with respect to the variables $X_k^{1:M}$ and $\mathbf{Z}_k$ results in,

$$\begin{aligned}
&E_{q_{X^{1:M}}^{(\xi)}, q_{\mathbf{Z}}^{(\xi)}}[\log p(\mathbf{Z}_k|x_k, X_k^{1:M}, \mathbf{r}_{1:k})] \\
&= -\frac{1}{2}\sum_{j=1}^{m_k} \mathrm{tr}\left[\left(E_{q_Z^{(\xi)}}[z_k^j] - H^{r_k^j}x_k \right) \left(E_{q_Z^{(\xi)}}[z_k^j] - H^{r_k^j}x_k \right)^T E_{q_X^{(\xi)}}\left[(sX_k^{r_k^j})^{-1} \right] \right] + c_x \\
&= -\frac{1}{2}\mathrm{tr}\left[m_k^1 \left((\overline{z}_k^1 - H^1 x_k)(\overline{z}_k^1 - H^1 x_k)^T E_{q_{X^1}^{(\xi)}}\left[(sX_k^1)^{-1} \right] \right) \right] \\
&\qquad\qquad \vdots \\
&\quad -\frac{1}{2}\mathrm{tr}\left[m_k^M \left((\overline{z}_k^M - H^M x_k)(\overline{z}_k^M - H^M x_k)^T E_{q_{X^M}^{(\xi)}}\left[(sX_k^M)^{-1} \right] \right) \right] + c_x. \qquad (4.22)
\end{aligned}$$

The above expression can be rewritten in terms of Gaussian densities as follows,

$$
\begin{aligned}
&E_{q^{(\xi)}_{X^{1:M}}, q^{(\xi)}_{\mathbf{Z}}}[\log p(\mathbf{Z}_k|x_k, X_k^{1:M}, \mathbf{r}_{1:k})] \\
&\qquad = \log \mathcal{N}\left(\overline{z}_k^1; H^1 x_k, \frac{\left(E_{q^{(\xi)}_{X^1}}\left[(sX_k^1)^{-1}\right]\right)^{-1}}{m_k^1} \right) \\
&\qquad\qquad \vdots \\
&\qquad + \log \mathcal{N}\left(\overline{z}_k^M; H^M x_k, \frac{\left(E_{q^{(\xi)}_{X^M}}\left[(sX_k^M)^{-1}\right]\right)^{-1}}{m_k^M} \right) + c_x.
\end{aligned}
\tag{4.23}
$$

Knowing that m_k^ℓ is the measurement number labeled to the subobject ℓ, we can express $\overline{z}_k$ as

$$
\overline{z}_k^\ell \triangleq \frac{1}{m_k^\ell} \sum_{j=1}^{m_k} \left\{ E_{q_Z^{(\xi)}}[z_k^j] \right\} \mathbb{1}(r_k^j = \ell) \quad \text{for } \ell = 1...M. \tag{4.24}
$$

Finally, (4.19) becomes,

$$
\begin{aligned}
\log q_x^{(\xi+1)}(x_k) = &\sum_{\ell=1}^{M} \log \mathcal{N}\left(\overline{z}_k^\ell; H^\ell x_k, \frac{\left(E_{q^{(\xi)}_{X^\ell}}\left[(sX_k^\ell)^{-1}\right]\right)^{-1}}{m_k^\ell} \right) \\
&+ \log \mathcal{N}(x_k; \hat{x}_{k|k-1}.P_{k|k-1}) + c.
\end{aligned}
\tag{4.25}
$$

We obtain a Gaussian density by taking the exponential of the both sides of (4.25), normalizing and using the KF measurement update equations [27],

$$
q_x^{(\xi+1)}(x_k) = \mathcal{N}(x_k; x_{k|k}^{(\xi+1)}, P_{k|k}^{(\xi+1)}), \tag{4.26}
$$

where the parameters are,

$$
x_{k|k}^{(\xi+1)} = P_{k|k}^{(\xi+1)} \left(P_{k|k-1}^{-1} x_{k|k-1} + \sum_{\ell=1}^{M} m_k^\ell (H^\ell)^T E_{q^{(\xi)}_{X^\ell}}[(sX_k^\ell)^{-1}] \overline{z}_k^\ell \right), \tag{4.27a}
$$

$$
P_{k|k}^{(\xi+1)} = \left(P_{k|k-1}^{-1} + \sum_{\ell=1}^{M} m_k^\ell (H^\ell)^T E_{q^{(\xi)}_{X^\ell}}[(sX_k^\ell)^{-1}] H^\ell \right)^{-1}. \tag{4.27b}
$$

4.2.1.2 Determination of $q_{X^\ell}^{(\xi+1)}(\cdot)$

We take expectation of (4.18) with respect to x_k, $\mathbf{Z}_k$ and $X_k^{-\ell}$ where $X_k^{-\ell}$ denotes the variables $\{X_k^1, \ldots, X_k^{\ell-1}, X_k^{\ell+1}, \ldots, X_k^M\}$. The parts that does not depend on X_k^ℓ are omitted and written as a constant term c_{X^ℓ}.

$$\begin{aligned}\log q_{X^\ell}^{(\xi+1)}(X_k^\ell) =& E_{q_x^{(\xi)}, q_{\mathbf{Z}}^{(\xi)}, q_{X^{-\ell}}^{(\xi)}}[\log p(\mathbf{Z}_k | x_k, X_k^{1:M}, \mathbf{r}_{1:k})] \\ &+ \log \mathcal{IW}(X_k^\ell; v_{k|k-1}^\ell, V_{k|k-1}^\ell) + c_{X^\ell}, \end{aligned} \tag{4.28}$$

where we can further express the expectation term as

$$\begin{aligned}&E_{q_x^{(\xi)}, q_{\mathbf{Z}}^{(\xi)}, q_{X^{-\ell}}^{(\xi)}}[\log p(\mathbf{Z}_k | x_k, X_k^{1:M}, \mathbf{r}_{1:k})] \\ &\quad = -\frac{m_k^\ell}{2} \log |s X_k^\ell| - \frac{1}{2} \mathrm{tr} \Bigg[\sum_{j=1}^{m_k} \Bigg(E_{q_x^{(\xi)}, q_{\mathbf{Z}}^{(\xi)}, q_{X^{-\ell}}^{(\xi)}} \Big[(z_k^j - H^\ell x_k) \\ &\qquad \times (z_k^j - H^\ell x_k)^T \Big] \Bigg) (s X_k^\ell)^{-1} \mathbb{1}(r_k^j = \ell) \Bigg] + c_{X^\ell}. \end{aligned} \tag{4.29}$$

Then, we rewrite (4.28) as

$$\begin{aligned}\log q_{X^\ell}^{(\xi+1)}(X_k^\ell) =& \log \mathcal{IW}(X_k^\ell; v_{k|k-1}^\ell, V_{k|k-1}^\ell) - \frac{m_k^\ell}{2} \log |s X_k^\ell| \\ &- \frac{1}{2} \mathrm{tr} \Bigg[\sum_{j=1}^{m_k} \Bigg(E_{q_x^{(\xi)}, q_{\mathbf{Z}}^{(\xi)}, q_{X^{-\ell}}^{(\xi)}} \Big[(z_k^j - H^\ell x_k)(z_k^j - H^\ell x_k)^T \Big] \Bigg) \\ &\times (s X_k^\ell)^{-1} \mathbb{1}(r_k^j = \ell) \Bigg] + c_{X^\ell}. \end{aligned} \tag{4.30}$$

The right hand side of (4.30) is sum of two logarithms of inverse Wishart densities. As a result, we combine them and obtain a new one,

$$q_{X^\ell}^{(\xi+1)}(X_k^\ell) = \mathcal{IW}(X_k^\ell; v_{k|k}^{\ell,(\xi+1)}, V_{k|k}^{\ell,(\xi+1)}), \tag{4.31}$$

where the parameters are,

$$v_{k|k}^{\ell,(\xi+1)} = v_{k|k-1}^\ell + m_k^\ell, \tag{4.32a}$$

$$V_{k|k}^{\ell,(\xi+1)} = V_{k|k-1}^\ell + \frac{1}{s} \sum_{j=1}^{m_k} \Bigg(E_{q_x^{(\xi)}, q_Z^{(\xi)}, q_{X^{-\ell}}^{(\xi)}} \big[(z_k^j - H^\ell x_k)(z_k^j - H^\ell x_k)^T \big] \Bigg) \mathbb{1}(r_k^j = \ell). \tag{4.32b}$$

4.2.1.3 Determination of $q_Z^{(\xi+1)}(\cdot)$

We take expectation of (4.18) with respect to x_k and $X_k^{1:M}$. The parts that does not depend on $\mathbf{Z}_k$ are omitted and written as a constant term $c_{\mathbf{Z}}$.

$$\log q_{\mathbf{Z}}^{(\xi+1)}(\mathbf{Z}_k) = \log p(\mathbf{Y}_k|\mathbf{Z}_k, \mathbf{r}_{1:k}) + E_{q_x^{(\xi)}, q_{X^{1:M}}^{(\xi)}}[\log p(\mathbf{Z}_k|x_k, X_k^{1:M}, \mathbf{r}_{1:k})] + c_{\mathbf{Z}}, \tag{4.33}$$

where the expectation term can be further expressed as,

$$\begin{aligned} E_{q_x^{(\xi)}, q_{X^{1:M}}^{(\xi)}}&[\log p(\mathbf{Z}_k|x_k, X_k^{1:M}, \mathbf{r}_{1:k})] \\ &= \sum_{j=1}^{m_k} -0.5tr\Bigg[\left(z_k^j - H^{r_k^j}\left(E_{q_x^{(\xi)}}[x_k]\right)\right)\left(z_k^j - H^{r_k^j}\left(E_{q_x^{(\xi)}}[x_k]\right)\right)^T \\ &\quad \times E_{q_{X^{r_k^j}}^{(\xi)}}\left[(sX_k^{r_k^j})^{-1}\right]\Bigg] + c_{\mathbf{Z}}. \end{aligned} \tag{4.34}$$

Finally, (4.33) can be rewritten as

$$\begin{aligned} \log q_{\mathbf{Z}}^{(\xi+1)}(\mathbf{Z}_k) = \sum_{j=1}^{m_k}\Bigg[&\log p(y_k^j; z_k^j, r_{1:k}^j) \\ &+ \log p\left(z_k^j;\ H^{r_k^j} E_{q_x^{(\xi)}}[x_k] E_{q_{X^{r_k^j}}^{(\xi)}}\left[(sX_k^{r_k^j})^{-1}\right]^{-1}\right)\Bigg] + c_{\mathbf{Z}}. \end{aligned} \tag{4.35}$$

We obtain a Gaussian density by taking the exponential of the both sides of (4.35), normalizing and using the KF measurement update equations,

$$\log q_{\mathbf{Z}}^{(\xi+1)}(\mathbf{Z}_k) = \prod_{j=1}^{m_k} \mathcal{N}(z_k^j; \hat{z}_k^{j,(\xi+1)}, \Sigma_k^{z,(\xi+1)}), \tag{4.36}$$

where the parameters are

$$\hat{z}_k^{j,(\xi+1)} = \Sigma_k^{z,r_k^j,(\xi+1)}\left(E_{q_{X^{r_k^j}}^{(\xi)}}\left[(sX_k^{r_k^j})^{-1}\right] H^{r_k^j} E_{q_x^{(\xi)}}[x_k] + R^{-1} y_k^j\right), \tag{4.37a}$$

$$\Sigma_k^{z,\ell,(\xi+1)} = \left(E_{q_{X^\ell}^{(\xi)}}\left[(sX_k^\ell)^{-1}\right] + R^{-1}\right)^{-1}. \tag{4.37b}$$

The following equations are used to compute the relevant expectations:

$$
\begin{aligned}
&E_{q_x^{(\xi)}}[x_k] = x_{k|k}^{(\xi)}, \quad E_{q_{\mathbf{Z}}^{(\xi)}}[z_k^j] = \hat{z}_k^{j,(\xi)}, \\
&E_{q_{X^\ell}^{(\xi)}}[(sX_k^\ell)^{-1}] = v_{k|k}^\ell \left(sV_{k|k}^{\ell,(\xi)}\right)^{-1}, \\
&E_{q_x^{(\xi)}, q_{\mathbf{Z}}^{(\xi)}, q_{X^{-\ell}}^{(\xi)}} \left[(z_k^j - H^{r_k^j} x_k)(z_k^j - H^{r_k^j} x_k)^T\right] \\
&\quad = \left(\hat{z}_k^{j,(\xi)} - H^{r_k^j} x_{k|k}^{(\xi)}\right)\left(\hat{z}_k^{j,(\xi)} - H^{r_k^j} x_{k|k}^{(\xi)}\right)^T \\
&\quad + H^\ell P_{k|k}^{(\xi)} (H^\ell)^T + \Sigma_k^{z,\ell,(\xi)}.
\end{aligned}
$$

The expected value, $E_{q_{X^\ell}^{(\xi)}}[(sX_k^\ell)^{-1}]$ is found using the theorem given in [15, Theorem 3.4.1]. The theorem states that if X is an inverse Wishart random variable with the following parameters $\mathcal{IW}(X; v, V)$, then X^{-1} is a Wishart random variable with the following parameters $\mathcal{W}(X^{-1}; v, V^{-1})$. Furthermore, the expected value of the Wishart density is equal to $v \times V^{-1}$ [15, Theorem 3.3.15].

The initial conditions for the quantities are chosen as follows:

$$
\begin{aligned}
&\hat{z}_k^{j,(0)} = y_k^j, \quad \Sigma_k^{z,\ell,(0)} = sX_{k|k-1}^\ell, \quad x_{k|k}^{(0)} = \hat{x}_{k|k-1}, \\
&P_{k|k}^{(0)} = P_{k|k-1}, \quad v_{k|k}^{\ell,(0)} = v_{k|k-1}^\ell, \quad V_{k|k}^{\ell,(0)} = V_{k|k-1}^\ell.
\end{aligned}
$$

Now, using the expressions derived so far, we can set up a variational iteration to find the estimate of approximate posteriors q_x, q_{X^ℓ} and $q_{\mathbf{Z}}$. In a similar fashion to [27], $\mathbf{Z}_k$ is marginalized out from the joint density, and an approximation for $p(x_k, X_k^{1:M} | \mathbf{r}_{1:k}, \mathbf{Y}_{1:k})$ is obtained.

4.3 Prediction, Sampling, and Weight Update

Once the measurement update of the continuous state is complete, a resampling is performed (if necessary), and we proceed to the next time step. In the next time step,

- time update of the continuous state,
- sampling the association variables,

- weight update

tasks should be performed.

4.3.1 Time Update

Suppose at time $k-1$, we have the following posterior density for the kinematic and extent states:

$$p(x_{k-1}^{(i)}, X_{k-1}^{1:M,(i)} | \mathbf{r}_{1:k-1}^{(i)}, \mathbf{Y}_{1:k-1}) = \mathcal{N}(x_{k-1}^{(i)}; \hat{x}_{k-1}^{(i)}, P_{k-1}^{(i)}) \times \prod_{\ell=1}^{M} \mathcal{IW}(X_{k-1}^{\ell,(i)}; v_{k-1}^{\ell,(i)}, V_{k-1}^{\ell,(i)}), \tag{4.39}$$

where the superscript $^{(i)}$ denotes the statistics of the ith particle. In the time update, the prediction density is computed by updating the sufficient statistics of the Gaussian and inverse Wishart components according to system dynamics. The dynamics of the state vector x_k is described with the state space model given below,

$$x_{k+1} = Fx_k + e_k, \quad e_k \sim \mathcal{N}(0, Q). \tag{4.40}$$

Then, the prediction density $\mathcal{N}(x_{k|k-1}^{(i)}; \hat{x}_{k|k-1}^{(i)}, P_{k|k-1}^{(i)})$ is obtained by updating the sufficient statistics (mean and covariance) of the Gaussian components in accordance with the system dynamics

$$\hat{x}_{k|k-1}^{(i)} = F\hat{x}_{k-1|k-1}^{(i)}, \tag{4.41a}$$

$$P_{k|k-1}^{(i)} = FP_{k-1|k-1}^{(i)}F^T + Q. \tag{4.41b}$$

Sufficient statistics of the the inverse Wishart density is updated by defining a forgetting factor γ_k

$$v_{k|k-1}^{(i),\ell} = \gamma_k v_{k-1|k-1}^{(i),\ell}, \tag{4.42a}$$

$$V_{k|k-1}^{(i),\ell} = \gamma_k V_{k-1|k-1}^{(i),\ell}. \tag{4.42b}$$

The use of the forgetting factor γ_k results in the maximum entropy estimate of the prediction density for systems with unknown but slowly varying transition dynamics [33, Theorem 1].

4.3.2 Sampling

The association variables can be sampled from their prior density π which is defined in (2.17) as follows:

$$r_k^{(i),j} \sim q(\cdot) = \pi \quad \text{for } i = 1, \ldots, N. \tag{4.43}$$

A more efficient approach would be choosing the proposal density as the optimal proposal density,

$$r_k^{(i),j} \sim q(\cdot) = \pi_{opt}^{(i),j} \quad \text{for } i = 1, \ldots, N. \tag{4.44}$$

Derivation of the optimal proposal density is given in the following subsection.

4.3.2.1 Optimal Proposal Density

The association variables can be sampled from the optimal proposal density [2] which is defined as

$$\pi_{opt}^{(i),j} \triangleq p(r_k^{(i),j} | y_{1:k}^j, r_{k-1}^{(i),j}) \propto p(y_k^j | r_k^{(i),j}, y_{1:k-1}^j) p(r_k^{(i),j} | r_{k-1}^{(i),j}). \tag{4.45}$$

For each sub-object ℓ the optimal proposal density is evaluated using the following approximate expressions,

$$\begin{aligned} \widetilde{\pi}_{opt,\ell}^{(i),j} &= \mathcal{N}\big(y_k^j; H^\ell \hat{x}_{k|k-1}^{(i)}, H^\ell P_{k|k-1}^{(i)} (H^\ell)^T + R + s\hat{X}_{k|k-1}^{(i),\ell}\big)\pi_\ell, \\ \pi_{opt,\ell}^{(i),j} &= \frac{\widetilde{\pi}_{opt,\ell}^{(i),j}}{\sum_{n=1}^{M} \widetilde{\pi}_{opt,n}^{(i),j}}, \end{aligned} \tag{4.46a}$$

where $\hat{x}_{k|k-1}^{(i)}$ and $P_{k|k-1}^{(i)}$ are given in (4.41). The approximation above is described in likelihood computation discussions given in the next subsection.

4.3.3 Weight Update

Once the samples from the association variables are obtained, the updated weights are calculated and normalized according to the following equations,

$$w_k^{(i)} \propto w_{k-1}^{(i)} \prod_{j=1}^{m_k} \frac{p(y_k^j | r_k^{(i),j}, y_{0:k-1}^j) \pi_{r_k^{(i),j}}}{q(r_k^{(i),j})}. \tag{4.47}$$

The likelihood $p(y_k^j|r_k^{(i),j}, y_{0:k-1}^j)$, a.k.a. predictive likelihood, in (4.47) can be found by computing the following integral:

$$\begin{aligned} p(y_k^j|r_k^{(i),j}, y_{0:k-1}^j) &= \iint p(y_k^j|x_k^{(i)}, X_k^{(i),r_k^j}, r_k^{(i),j}, y_{1:k}^j) \\ &\quad \times p(x_k^{(i)}, X_k^{(i),r_k^j}|r_k^{(i),j}, y_{1:k}^j)dx_k dX_k \qquad (4.48a) \\ &= \iint \mathcal{N}(y_k^j; H^{r_k^j} x_k^{(i)}, R + sX_k^{(i),r_k^j}) \mathcal{N}(x_k^{(i)}; \hat{x}_{k|k-1}^{(i)}, P_{k|k-1}^{(i)}) \\ &\quad \times \mathcal{IW}(X_k^{(i),r_k^j}; V_{k|k-1}^{(i),r_k^j}, v_{k|k-1}^{(i),r_k^j})dx_k dX_k. \qquad (4.48b) \end{aligned}$$

There is no compact analytical expression for the integral given above, hence an approximation is necessary. [9] and [27] provide different approximations for the above integral.

When the degrees of freedom is large, the inverse Wishart density becomes peaky around its mean. In that case, the likelihood can be approximated as

$$p(y_k^j|r_k^{(i),j}, y_{0:k-1}^j) \approx \mathcal{N}(y_k^j; H^{r_k^j} \hat{x}_{k|k-1}^{(i)}, H^{r_k^j} P_{k|k-1}^{(i)} (H^{r_k^j})^T + R + s\hat{X}_{k|k-1}^{(i),r_k^j}). \quad (4.49)$$

Although simple, we used the approximation given by (4.49) in our simulations to demonstrate the robustness of the proposed algorithm to likelihood approximations. The approximations presented in [9] and [27] can also be used at the expense of computation time.

After having computed the importance weights, the measurement update is performed as described in Section 4.2. Lastly, resampling is performed if necessary. The pseudo-code of the proposed method is given in Algorithm 4.2.

Algorithm 4.2: Summary of Variational inference based MPF

Initialization at $k = 0$:
for all particles $i = 1, \ldots, N$ **do**
  Set initial weights $w_0^{(i)} = \frac{1}{N}$
  Set initial values for parameters $\hat{x}_0^{(i)}, P_0^{(i)}, v_0^{(i)}, V_0^{(i)}$
end for
Iterations:
for time $k = 1, \ldots, T$ **do**
  for all particles $i = 1, \ldots, N$ **do**
    Perform time update for the continuous states according to (4.41) and (4.42)
    for all measurements $j = 1, \ldots, m_k$ **do**
      Sample the association variables, $r_k^{(i),j} \sim q(\cdot)$
      Compute the likelihood, $p(y_k^j | r_k^{(i),j}, y_{1:k-1}^j)$
    end for
    Update the weights according to (4.47)
  end for
  Normalize the weights
  for all particles $i = 1, \ldots, N$ **do**
    Perform measurement update for the continuous state using variational Bayes
  end for
  Resample, if necessary
end for

4.4 Performance Evaluation

In this section, the performance of the proposed method is evaluated in three different simulations. The object follows a linear path and constant velocity model is used for the position of the object as described in Section 3.3.1. In the first two simulations, the proposed method is compared with the random matrix-based method proposed in [24]. The third simulation is performed only with the proposed method. The method proposed here is denoted with VPF and the one in [24] is denoted with JL.

4.4.1 The Method of JL

A standard random matrix approach proposed by [23] performs extended target tracking using single ellipse represented with SPD matrix. The joint density of the extent and kinematic states is approximated with Gaussian inverse Wishart density and estimated using a Bayesian approach. The approach proposed by [24] differs from that of [23] in the prediction step of the extent state and the update step of both kinematic and extent states. Moreover, the new approach is applicable for non-ellipsoidal shapes where single ellipse is not enough for approximation. Similar to the method proposed in this paper, multiple ellipses are used while representing the target extent. Each sub-object ℓ is represented by an SPD random matrix X_k^ℓ. Unlike our method, each sub-object has separate kinematics represented with a random vector x_k^ℓ. The system model can be expressed with,

$$x_k^\ell = \phi_k^\ell x_{k-1}^\ell + w_k^\ell, \quad w_k^\ell \sim \mathcal{N}\left(0, D_k^\ell \otimes X_k^\ell\right) \tag{4.50a}$$

$$z_k = \widetilde{H}_k^\ell x_k^\ell + v_k^\ell, \quad v_k^\ell \sim \mathcal{N}\left(0, B_k^\ell X_k^\ell (B_k^\ell)^T\right) \tag{4.50b}$$

$$p[X_k^\ell | X_{k-1}^\ell] = \mathcal{W}\left(X_k^\ell; \delta_k^\ell, A_k^\ell X_{k-1}^\ell (A_k^\ell)^T\right) \tag{4.50c}$$

where $\phi_k^\ell = F_k^\ell \otimes I_d$ and $\widetilde{H}_k^\ell = H_k^\ell \otimes I_d$. Because that the origins of measurements are unknown, all the association possibilities should be considered with a likelihood defined in [24, Sec. III-B.]. There are $(n_k^\ell)^{n_k}$ number of possibilities where n_k^ℓ is the sub-object number and n_k is the measurement number. The

Table 4.1: Summary of the method JL

State	$\hat{x}_{k\|k-1} = (F_k \otimes I_d)\hat{x}_{k-1}$
	$P_{k\|k-1} = F_k P_{k-1} F_k^T + D_k$
	$\hat{x}_k = \hat{x}_{k\|k-1} + (K_k \otimes I_d)G_k$
	$P_k = P_{k\|k-1} - K_k S_{k\|k-1} K_k^T$
	$S_{k\|k-1} = H_k P_{k\|k-1} H_k^T + 1/n_k \|B_k\|^{2/d}$
	$K_k = P_{k\|k-1} H_k^T S_{k\|k-1}^{-1}$
	$G_k \triangleq \bar{y}_k - (H_k \otimes I_d)\hat{x}_{k\|k-1}$
	$\bar{y}_k = \frac{1}{n_k} \sum_{r=1}^{n_k} y_k^r$
	$E[x_k \| Y_{1:k}] = \hat{x}_k$
Extension	$\hat{X}_{k\|k-1} = \frac{\delta_k}{\lambda_{k-1}}(\hat{v}_{k\|k-1} - 2d - 2) A_k \hat{X}_{k-1} A_k^T$
	$\lambda_{k-1} = (\hat{v}_{k-1} - 2d - 2$
	$\hat{v}_{k\|k-1} = \frac{2\delta_k(\lambda_{k-1}+1)(\lambda_{k-1}-1)(\lambda_{k-1}-2)}{\lambda_{k-1}^2(\lambda_{k-1}+\delta_k)} + 2d + 4$
	$\hat{X}_k = \hat{X}_{k\|k-1} + N_{k\|k-1} + B_k^{-1} \bar{Y}_k B_k^{-T}$
	$\hat{v}_k = \hat{v}_{k\|k-1} + n_k$
	$N_{k\|k-1} = S_{k\|k-1}^{-1} G_k G_k^T$
	$\bar{Y}_k = \sum_{r=1}^{n_k} (y_k^r - \bar{y}_k)(y_k^r - \bar{y}_k)^T$
	$E[X_k \| Y_{1:k}] = \bar{X}_k = \hat{X}_k / (\hat{v}_k - 2d - 2)$

update equations for the algorithm can be summarized in Table 4.1. Detailed derivations and explanations can be found in [24].

4.4.2 Simulation

100 MC runs are performed for the simulations. Different measurement realizations are generated at each run. The numeric results presented here is an average of the MC runs. The VPF uses $N = 100$ particles and a forgetting factor of $\gamma_k = 0.95$ in the simulations. π vector is chosen as in equation (2.19) which assigns equal probabilities for each mode. This is a reasonable choice when the measurement rates of the sub-objects are unknown.

As a performance metric for the position estimate, RMSE as defined in (2.8)

will be used. Performance measure for the target extent estimation is chosen as intersection over union (IOU), which is originally used in computer vision [1] and later employed in ETT [13]. Let A and B denote the true and the estimated extents respectively, then IOU is defined as the fraction of the areas of intersection and union of the extents A and B,

$$\mathrm{IOU}(A,B) = \frac{\mathrm{area}(A \cap B)}{\mathrm{area}(A \cup B)} \in [0,1]. \tag{4.51}$$

Note that IOU takes values between 0 and 1 where 0 corresponds to the worst possible match and 1 corresponds to the best possible match.

In the first simulation, we consider the scenario S1 given in [24] which is called as the line scenario since the measurements originate from 9 constant points forming perpendicular two lines as illustrated in Figure 4.2. The values of the measurements are given as

$$y^1 = \begin{bmatrix} -20 \\ 40 \end{bmatrix},\ y^2 = \begin{bmatrix} -20 \\ 20 \end{bmatrix},\ y^3 = \begin{bmatrix} -20 \\ 0 \end{bmatrix},\ y^4 = \begin{bmatrix} -20 \\ -20 \end{bmatrix},$$
$$y^5 = \begin{bmatrix} -20 \\ -40 \end{bmatrix},\ y^6 = \begin{bmatrix} -40 \\ 0 \end{bmatrix},\ y^7 = \begin{bmatrix} 0 \\ 0 \end{bmatrix},\ y^8 = \begin{bmatrix} 20 \\ 0 \end{bmatrix},\ y^9 = \begin{bmatrix} 40 \\ 0 \end{bmatrix}.$$

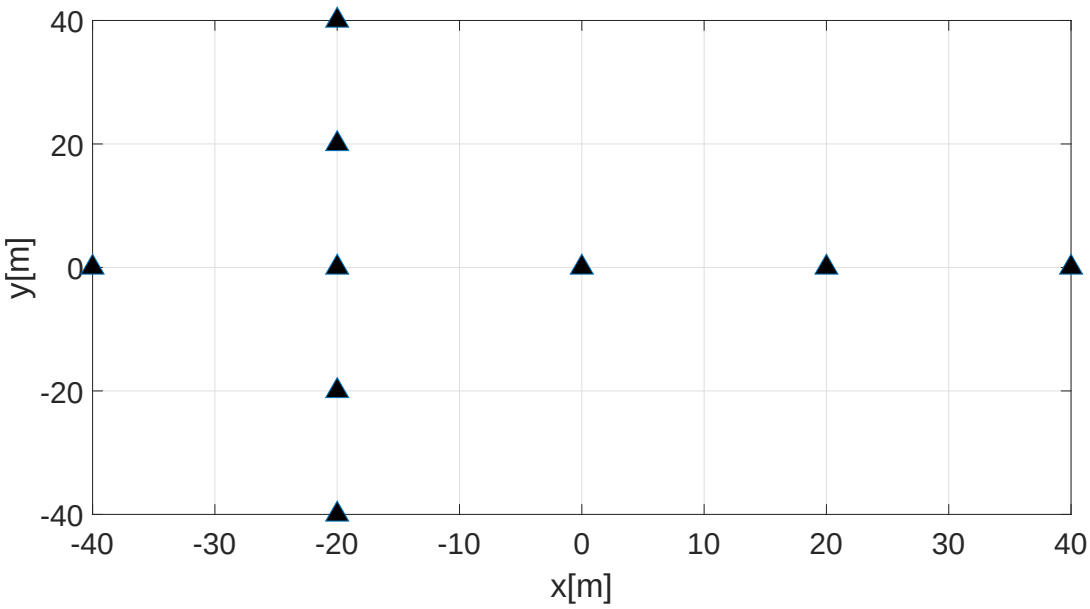

Figure 4.2: Measurement model of the line scenario.

The target moves on a linear trajectory with the measurement model provided above and estimation performances of two algorithms are compared. For this purpose, all the initial conditions of the unknown parameters are kept the same for two algorithms as described in [24]. Since there is no true extent surface here, IOU cannot be computed. Therefore we only provide RMSE values of the two algorithms in Table 4.2. An example of one MC run with snapshots of the corresponding frames $t \in \{0, 8, 16, 24, 32, 40\}$ are shown in Fig. 4.3.

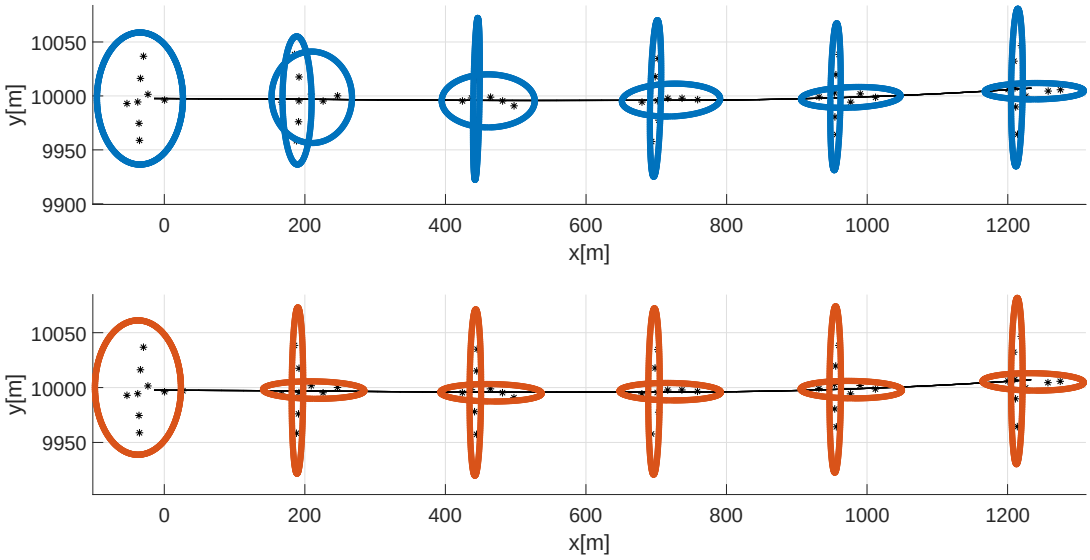

Figure 4.3: Estimation results of a single MC run, for the target represented with two perpendicular lines. The JL estimate is shown with blue, the VPF estimate is shown with orange solid lines. The measurements at each scan are represented with star-shaped dots.

The second simulation is performed with a target which consists of two elliptical sub-objects and it is called as elliptic scenario. The measurements originate from the surface of the ellipses whose true values are given as

$$\mu^1_{true} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \ \mu^2_{true} = \begin{bmatrix} -10 \\ 0 \end{bmatrix}, \ X^1_{true} = \begin{bmatrix} 400 & 0 \\ 0 & 10 \end{bmatrix}, \ X^2_{true} = \begin{bmatrix} 10 & 0 \\ 0 & 400 \end{bmatrix}.$$

The illustration of the ellipses are given in Figure 4.4.

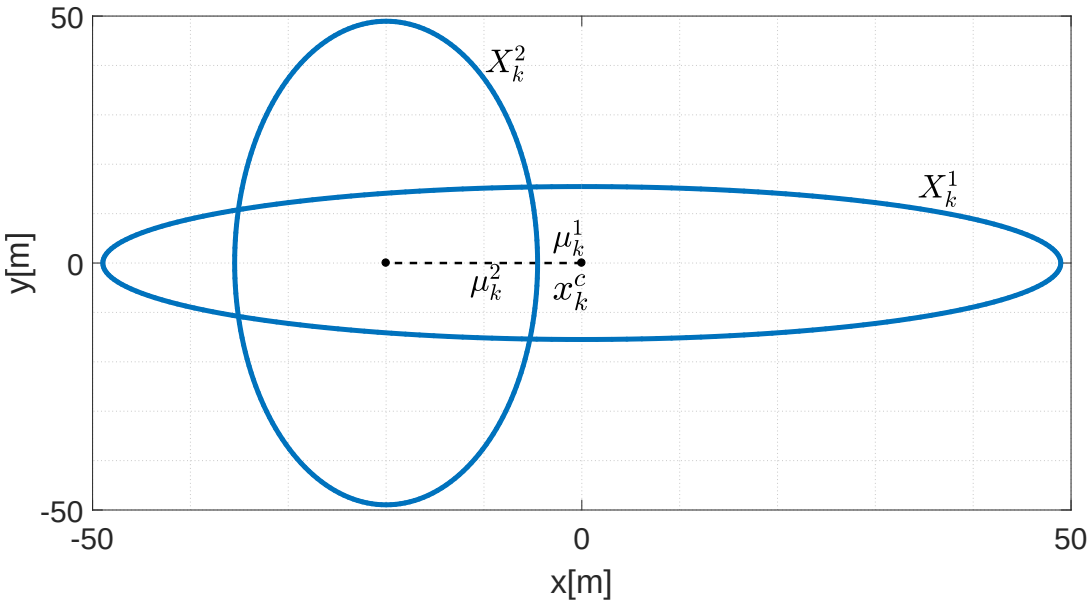

Figure 4.4: Measurement model of the elliptical scenario.

The object moves on a linear trajectory the same as the previous case with the parameters of simulation the S2 in [24]. The measurement number is kept constant at each scan as $N_{meas} = 9$. An example of one MC run with snapshots of the corresponding frames $t \in \{0, 8, 16, 24, 32, 40\}$ are shown in Fig. 4.5. RMSE and IOU results are presented in Table 4.2.

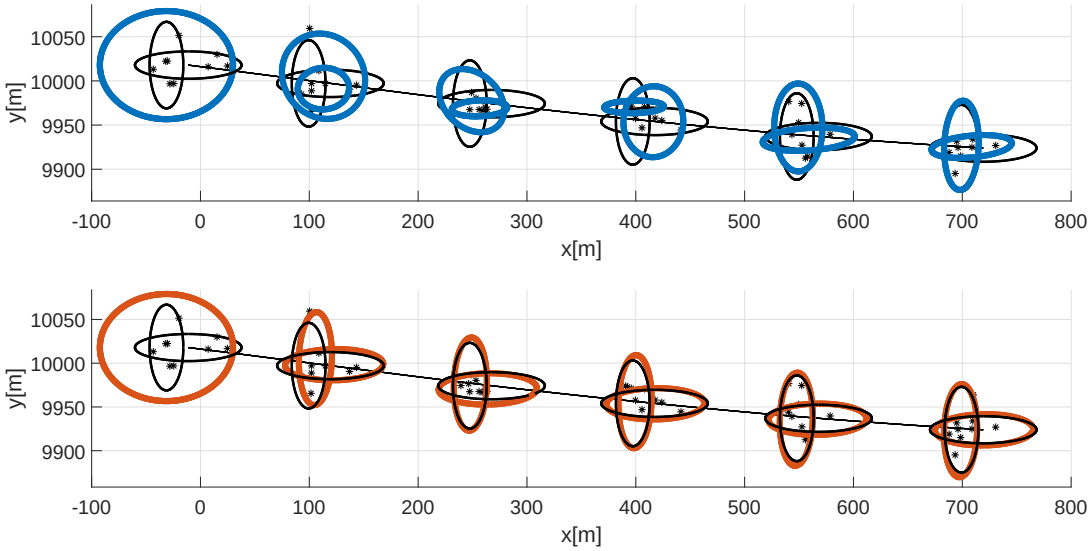

Figure 4.5: Estimation results of a single MC run, for the target with two elliptical sub-objects. The JL estimate is shown with blue, the VPF estimate is shown with orange and the true extent is represented with black solid lines. The measurements at each scan are represented with star-shaped dots.

Table 4.2: RMSE and IOU results of the JL and the VPF in the first two simulations

	Sub-object	**IOU**	**RMSE[m]**	
			Line Scenario	***Elliptical Scenario***
JL	1^{st}	0.62	2.08	5.81
	2^{nd}		1.09	8.52
VPF	1^{st}	0.86	1.79	2.29
	2^{nd}		1.02	2.75

In the first two simulations, the VPF algorithm performs better in terms of both RMSE and IOU. Compared with the JL, the extent estimates of the VPF are more stable, in the sense that no abrupt changes are observed in consecutive frames. The computation times per update of two algorithms are given in the Table 4.3. In contrast to the JL method, the computation time of the proposed algorithm does not increase exponentially with the number of measurements. All simulations are run in Matlab$^{(R)}$ R2016b on a standard laptop with an Intel$^{(R)}$

Core$^{(TM)}$ i5-7200U 2.50 GHz platform with 8 GB of RAM running Windows.

Table 4.3: Average computation times per update of the JL and the VPF

	Computation Time[s]	
	9-Measurements	***15-Measurements***
JL	0.14	8.38
VPF	0.37	0.56

In the third simulation, the performance of the proposed algorithm will be demonstrated using the system model described in 3.3.1 and a target that consists of 5 sub-objects as illustrated in 4.6.

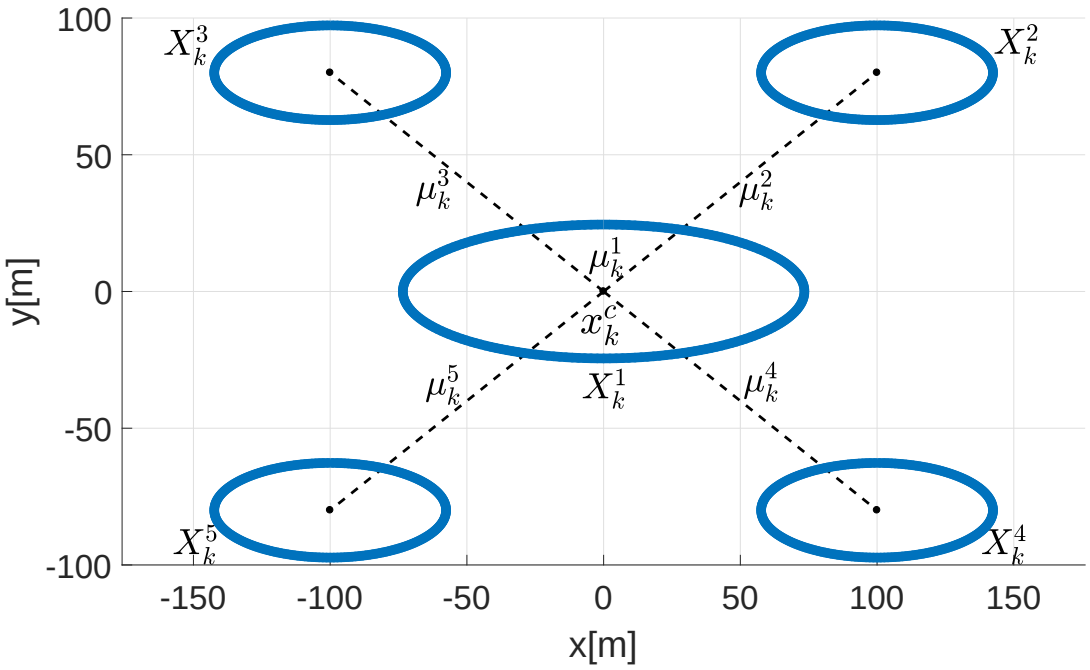

Figure 4.6: Target extent model with 5 ellipses.

The true values of the matrices and vectors representing the model are given as

follows:

$$X_{true}^{1} = \begin{bmatrix} 900 & 0 \\ 0 & 100 \end{bmatrix}, \quad X_{true}^{2,3,4,5} = \begin{bmatrix} 300 & 0 \\ 0 & 50 \end{bmatrix},$$

$$\mu_{true}^{1} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \ \mu_{true}^{2} = \begin{bmatrix} -100 \\ 80 \end{bmatrix}, \ \mu_{true}^{3} = \begin{bmatrix} 100 \\ 80 \end{bmatrix}, \ \mu_{true}^{4} = \begin{bmatrix} -100 \\ -80 \end{bmatrix}, \ \mu_{true}^{5} = \begin{bmatrix} 100 \\ -80 \end{bmatrix}.$$

The number of measurements at each scan is Poisson distributed with an average of 20 measurements. The system noise variance is set to $\sigma^2 = 0.1$ and the measurement noise covariance is selected to be $R = \text{diag}([10, 10])\text{m}^2$. The scale factor is taken as $s = 1$. Number of variational iterations is set to 5. Initial kinematics, initial covariance matrix and initial extents are chosen as

$$x_0 = \begin{bmatrix} 0 \text{ m} \\ 10^4 \text{ m} \\ 200 \text{ m/s} \\ 0 \text{ m/s} \end{bmatrix}, \quad P_0 = \begin{bmatrix} 400 & 0 & 0 & 0 \\ 0 & 400 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 100 \end{bmatrix}, \quad X_0^{1,2,3,4,5} = \begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix} \text{ m}^2.$$

The initial degree of freedom is set to $v_0 = 10$. An example of one MC run with snapshots of the corresponding frames $t \in \{0, 20, 40, 60, 80, 100\}$ are shown in Figure 4.7.

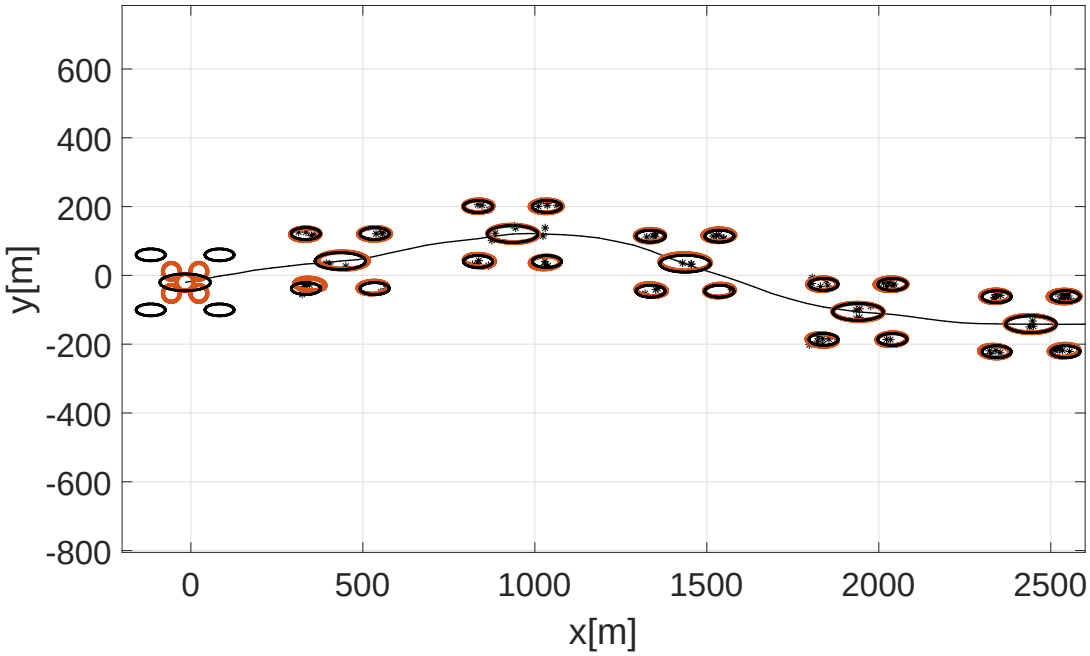

Figure 4.7: Estimation results of a single MC run, for the target with 5 elliptical sub-objects. The VPF estimate is shown with orange, true extent is shown with black solid lines. The measurements at each scan are represented with star-shaped dots.

RMSE and IOU results are presented in Table 4.4.

Table 4.4: RMSE and IOU results of the VPF for third simulation

	VPF					
	Sub-obj:	1^{st}	2^{nd}	3^{rd}	4^{th}	5^{th}
RMSE[m]		1.50	3.48	4.59	3.42	3.96
IOU		0.88	0.86	0.85	0.86	0.85

As shown in Figure 4.7 and in Table 4.4 the algorithm is capable of tracking the extended object with five sub-objects successfully. The positions and extents of all sub-object are estimated accurately. In our experiments, we observed that the performance of the algorithm is not highly sensitive to the selection of π.

4.5 Conclusion

A new approach for tracking non-ellipsoidal targets is proposed, where the target is modeled with multiple ellipses. The resulting inference problem is treated in SMC framework. An efficient particle filter is derived using the marginalization technique. The variational approximation is used to obtain an analytical approximation for the joint density of the multiple unknown extent matrices, target kinematic state, and unknown mean vectors of the extent. This approximation enables, otherwise impossible, marginalization. Furthermore, optimal proposal distribution is used for minimizing the estimation variance and boosting the particle efficiency.

CHAPTER 5

CONCLUSION AND FUTURE WORK

In this research, the problem of extended target tracking is studied in the SMC framework. Various object extents are represented by multiple ellipses. This multi-ellipse representation creates an association problem between the measurements and the sub-objects. This problem is solved by defining discrete association random variables. The joint density of the discrete and continuous states is approximated using PF where different marginalization strategies are possible. In the first part of the thesis, the ETT problem is handled under the assumption of known extent to focus on the association problem. Four different PF methods are investigated and their performances are compared in simulations.

The marginalization technique significantly improves the performance under the same computational capacity. The standard particle filter without marginalization approximates the density of the discrete and continuous states using weighted particles. Thus, it requires an excessive number of particles to provide satisfying results. The marginalized algorithms (CMPF and DMPF), on the other hand, uses analytical expressions for either the continuous state or the discrete state. Therefore, they require a smaller number of particles.

We investigate two different marginalization techniques. In the first one (CMPF), the continuous state is marginalized out by approximating the density of the discrete states using weighted particles. The conditional density of the continuous state is found analytically using Kalman filters. In the second marginalization technique (DMPF), the discrete state is marginalized out by approximating the density of the continuous states using weighted particles. The conditional density of the discrete state is found analytically using HMM filter. The CMPF

algorithm requires more particles as the number of measurements increases. In the DMPF algorithm, the space that we sample remains the same regardless of the increasing number of measurements.

We also illustrate the advantages of using optimal importance sampling in the CMPF algorithm via simulations. Since the optimal proposal density (CMPF-O) also uses the information provided by the latest measurement, it provides better results with a smaller number of particles.

Among the particle filters considered here, the CMPF-O and the DMPF perform best. The CMPF-O provides satisfying results even with a small number of particles however, its computation time is the highest one. The CMPF-O uses Kalman filter measurement update equations and optimal proposal density calculations per particle per time step. Performance of the DMPF is very close to that of the CMPF-O and it requires far less computational power. Therefore using the DMPF in practical applications could be beneficial. To the best of our knowledge, it is the first time that these novel SMC methods are used in ETT problems.

In the second part of the thesis, the complete problem of non-ellipsoidal extended target tracking is considered. A new particle filter based approach for the joint estimation of kinematic and extent states is proposed which uses marginalization and variational Bayes techniques. The continuous state is marginalized out and the density of the association variables is approximated using weighted samples. The density of the continuous state includes the extent and kinematic states which is represented with normal-inverse Wishart density. To be able to perform marginalization, the density of the continuous states should be expressed analytically which is not possible in our case. Therefore, we propose to use the variational Bayes method to obtain approximate analytical expressions. Our proposed algorithm is a novel approach which is based on the variational Bayes and marginalized particle filter. Among the non-ellipsoidal ETT methods which are based on the random matrix model, our algorithm is the only one which does not require mixture reduction, merging or clustering steps. The performance of the algorithm is tested in a simulation and compared with an

existing method of [24]. It is shown that the proposed algorithm outperforms the existing solution.

Some possible directions for future work are proposed as follows. The prior probabilities that a measurement is associated with sub-objects are assumed to be known in our model. It would be interesting to estimate these probabilities by defining appropriate prior. The performance of the algorithm can be improved by including maneuvering motion models and clutter rejection techniques. Another different but important task is to estimate the number of sub-objects.

REFERENCES

[1] B. Alexe, T. Deselaers, and V. Ferrari. Measuring the objectness of image windows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2189–2202, Nov 2012.

[2] N. d. F. . N. G. Arnaud Doucet, editor. *Sequential Monte Carlo Methods in Practice*. Springer, 2001.

[3] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, Feb 2002.

[4] M. Baum and U. D. Hanebeck. Random hypersurface models for extended object tracking. In *2009 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, pages 178–183, Dec 2009.

[5] M. Baum and U. D. Hanebeck. Shape tracking of extended objects and group targets with star-convex rhms. In *14th International Conference on Information Fusion*, pages 1–8, July 2011.

[6] M. Baum, V. Klumpp, and U. D. Hanebeck. A Novel Bayesian Method for Fitting a Circle to Noisy Points. In *Proceedings of the 13th International Conference on Information Fusion (Fusion 2010)*, Edinburgh, United Kingdom, July 2010.

[7] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[8] A. Doucet, S. Godsill, and C. Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3):197–208, Jul 2000.

[9] M. Feldmann, D. Franken, and W. Koch. Tracking of extended objects and group targets using random matrices. *IEEE Transactions on Signal Processing*, 59(4):1409–1420, April 2011.

[10] N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings F - Radar and Signal Processing*, 140(2):107–113, April 1993.

[11] K. Granström and M. Baum. Extended object tracking: Introduction, overview and applications. *CoRR*, abs/1604.00970, 2016.

[12] K. Granström and C. Lundquist. On the use of multiple measurement models for extended target tracking. In *Proceedings of the 16th International Conference on Information Fusion*, pages 1534–1541, July 2013.

[13] K. Granström, C. Lundquist, and U. Orguner. Tracking rectangular and elliptical extended targets using laser measurements. In *14th International Conference on Information Fusion*, pages 1–8, July 2011.

[14] K. Granström, P. Willett, and Y. Bar-Shalom. An extended target tracking model with multiple random matrices and unified kinematics. In *2015 18th International Conference on Information Fusion (Fusion)*, pages 1007–1014, July 2015.

[15] A. Gupta and D. Nagar. *Matrix Variate Distributions*. PMS Series. Addison-Wesley Longman, Limited, 1999.

[16] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P. J. Nordlund. Particle filters for positioning, navigation, and tracking. *IEEE Transactions on Signal Processing*, 50(2):425–437, Feb 2002.

[17] Q. Hu, H. Ji, and Y. Zhang. Tracking of maneuvering non-ellipsoidal extended target with varying number of sub-objects. *Mechanical Systems and Signal Processing*, 99:262 – 284, 2018.

[18] M. Isard and A. Blake. Condensation—conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, Aug 1998.

[19] A. Jazwinski. *Stochastic processes and filtering theory*. Number 64 in Mathematics in science and engineering. Acad. Press, New York, NY [u.a.], 1970.

[20] S. Julier and J. Uhlmann. A new extension of the Kalman filter to nonlinear systems. In *Proc. of AeroSense: The 11th Int. Symp. on Aerospace/Defense Sensing, Simulations and Controls*, 1997.

[21] K. Kanazawa, D. Koller, and S. J. Russell. Stochastic simulation algorithms for dynamic probabilistic networks. *CoRR*, abs/1302.4965, 2013.

[22] G. Kitagawa. Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics*, 5(1):1–25, 1996.

[23] J. W. Koch. Bayesian approach to extended object and cluster tracking using random matrices. *IEEE Transactions on Aerospace and Electronic Systems*, 44(3):1042–1059, July 2008.

[24] J. Lan and X. R. Li. Tracking of maneuvering non-ellipsoidal extended object or target group using random matrix. *IEEE Transactions on Signal Processing*, 62(9):2450–2463, May 2014.

[25] J. Lan and X. R. Li. Tracking of extended object or target group using random matrix: New model and approach. *IEEE Transactions on Aerospace and Electronic Systems*, 52(6):2973–2989, December 2016.

[26] X. R. Li and V. P. Jilkov. Survey of maneuvering target tracking. part i. dynamic models. *IEEE Transactions on Aerospace and Electronic Systems*, 39(4):1333–1364, Oct 2003.

[27] U. Orguner. A variational measurement update for extended target tracking with random matrices. *IEEE Transactions on Signal Processing*, 60(7):3827–3834, July 2012.

[28] B. Ristic, S. Arulampalam, and N. Gordon. *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House, 2004.

[29] T. Schön, F. Gustafsson, and P. J. Nordlund. Marginalized particle filters for mixed linear/nonlinear state-space models. *IEEE Transactions on Signal Processing*, 53(7):2279–2289, July 2005.

[30] D. D. Sworder and J. E. Boyd. *Estimation Problems in Hybrid Systems*. Cambridge University Press, 1999.

[31] N. Wahlström and E. Özkan. Extended target tracking using Gaussian processes. *IEEE Transactions on Signal Processing*, 63(16):4165–4178, Aug 2015.

[32] E. Özkan, F. Lindsten, C. Fritsche, and F. Gustafsson. Recursive maximum likelihood identification of jump Markov nonlinear systems. *IEEE Transactions on Signal Processing*, 63(3):754–765, Feb 2015.

[33] E. Özkan, V. Šmídl, S. Saha, C. Lundquist, and F. Gustafsson. Marginalized adaptive particle filtering for nonlinear models with unknown time-varying noise parameters. *Automatica*, 49(6):1566 – 1575, 2013.

APPENDIX A

MULTIVARIATE GAUSSIAN DENSITY

The notation $\mathcal{N}(\mu, \Sigma)$ with a mean vector μ and a covariance matrix Σ represents the multivariate Gaussian density which is given explicitly as

$$\mathcal{N}(x; \mu, \Sigma) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right). \tag{A.1}$$

The Gaussian density can also be expressed using the trace operator. The trace of any $n \times n$ matrix can be expressed as,

$$\text{tr}[A] = \sum_{i=1}^{n} a_{ii}, \tag{A.2}$$

where a_{ij} is the element where intersection point of i^{th} row and j^{th} column. The trace operator holds the following properties:

(1) Invariance under matrix products, $\quad tr[ABC] = \text{tr}[CAB] = \text{tr}[BCA]$

(2) if the matrix product AB results in a scalar value, then $\quad \text{tr}[AB] = AB.$

Now we can use these properties for rewriting the multivariate Gaussian density as

$$\begin{aligned}\mathcal{N}(x; \mu, \Sigma) &= \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right) \\ &= \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}\text{tr}\left[(x-\mu)(x-\mu)^T\Sigma^{-1}\right]\right).\end{aligned} \tag{A.3}$$

The equation above is satisfied first using the property (2) as $(x-\mu)^T\Sigma^{-1}(x-\mu) = \text{tr}[(x-\mu)^T\Sigma^{-1}(x-\mu)]$. Then with property (1) we replace the matrices inside the trace and obtain, $\text{tr}\left[(x-\mu)^T\Sigma^{-1}(x-\mu)\right] = \text{tr}\left[(x-\mu)(x-\mu)^T\Sigma^{-1}\right]$.

APPENDIX B

INVERSE WISHART DENSITY

The inverse Wishart density is denoted with $\mathcal{IW}(X; v, V)$ and expressed explicitly as

$$\mathcal{IW}(X; v, V) = \frac{|V|^{v/2}}{|X|^{\frac{v+d+1}{2}} 2^{\frac{vd}{2}} \Gamma_d(\frac{v}{2})} \exp\left[-\frac{1}{2}\mathrm{tr}(VX^{-1})\right], \tag{B.1a}$$

$$\Gamma_d(v/2) = \pi^{d(d-1)/4} \prod_{i=1}^{d} \Gamma(\frac{v+1-i}{2}), \tag{B.1b}$$

where V is the SPD scale matrix, v is the scalar degree of freedom, d is the dimension of X and $\Gamma(z) = \int_0^\infty t^{z-1}e^{-t}dt$ represents the standard gamma function. The expected value of the density is provided as

$$E\{X\} = \frac{V}{v-d-1}, \quad v > d+1. \tag{B.2}$$