The
University
Of
Sheffield.

Sequential Monte Carlo Methods for Crowd and Extended Object Tracking and Dealing with Tall Data

Allan De Freitas

A thesis submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy

Department of Automatic Control and Systems Engineering
University of Sheffield
January 2017

This thesis is dedicated to Trinette and my family
for their endless love and encouragement

ABSTRACT

The Bayesian methodology is able to deal with a number of challenges in object tracking, especially with uncertainties in the system dynamics and sensor characteristics. However, model complexities can result in non-analytical expressions which require computationally cumbersome approximate solutions. In this thesis computationally efficient approximate methods for object tracking with complex models are developed.

One such complexity is when a large group of objects, referred to as a crowd, is required to be tracked. A crowd generates multiple measurements with uncertain origin. Two solutions are proposed, based on a box particle filtering approach and a convolution particle filtering approach. Contributions include a theoretical derivation for the generalised likelihood function for the box particle filter, and an adaptive convolution particle filter able to resolve the data association problem without the measurement rates. The performance of the two filters is compared over a realistic scenario for a large crowd of pedestrians.

Extended objects also generate a variable number of multiple measurements. In contrast with point objects, extended objects are characterised with their size or volume. Multiple object tracking is a notoriously challenging problem due to complexities caused by data association. An efficient box particle filter method for multiple extended object tracking is proposed, and for the first time it is shown how interval based approaches can deal efficiently with data association problems and reduce the computational complexity of the data association. The performance of the method is evaluated on real laser rangefinder data.

Advances in digital sensors have resulted in systems being capable of accumulating excessively large volumes of data. Three efficient Bayesian inference methods are developed for object tracking when excessively large numbers of measurements may otherwise cause standard algorithms to be inoperable. The underlying mechanics of these methods are adaptive subsampling and the expectation propagation algorithm.

ACKNOWLEDGMENTS

First and foremost, I would like to thank my Lord and Saviour for granting me the opportunity and talent to conduct the research which lead to this thesis.

I would like to sincerely thank my supervisor Prof. Lyudmila Mihaylova for her continual support and enthusiasm. Her constant availability, combined with a willingness to share her knowledge, has tremendously benefited my postgraduate experience in Sheffield.

I would also like to express my gratitude to my second supervisor, Prof. Visakan Kadirkamanathan, and to Dr. Amadou Gning, Dr. François Septier and Prof. Simon Godsill, for their scientific support and guidance through out the project.

I give thanks to Prof. Fredrik Gustafsson, Dr. Carsten Fritsche and the team at Linköping University, as well as Dr. Fredrik Gunnarsson and the team at Ericsson, for the insightful discussions and hospitality during my research exchange in Linköping.

My time as a PhD student would not have been as motivating and stimulating without the discussions and knowledge exchange with my fellow colleagues in Sheffield and the whole TRAX project consortium.

On a personal note, I would like to thank my wife, Trinette, for embarking on this adventure in the UK with me and for always being a great source of comfort. I am also grateful to my parents, grandmother, siblings and in-laws for always believing in me and for their unconditional support in my life.

Finally, I would like to acknowledge the financial support from the European Union under the Marie Curie Actions in the Seventh Framework Program [FP7 2013-2017] TRAcking in compleX sensor systems (TRAX) with grant agreement no. 607400.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

Box PF	Box Particle Filter
BP PF	Border Parameterised Particle Filter
CDF	Cumulative Density Function
CSP	Constraints Satisfaction Problem
CP	Constraints Propagation
CPF	Convolution Particle Filter
EKF	Extended Kalman Filter
EP	Expectation Propagation
ESS	Effective Sample Size
GPS	Global Positioning System
KF	Kalman Filter
KS	Kolmogorov-Smirnov
LiDAR	Light Detection and Ranging
MCMC	Markov Chain Monte Carlo
MC	Monte Carlo

MH	Metropolis-Hastings
NP	Natural Parameters
OSPA	Optimal SubPattern Assignment
PDF	probability density function
PF	Particle Filter
PHD	Probability Hypothesis Density
RFS	Random Finite Set
RMSE	Root Mean Square Error
SIR	Sequential Importance Resampling
SIS	Sequential Importance Sampling
SMCMC	Sequential Markov Chain Monte Carlo
SMC	Sequential Monte Carlo
UKF	Unscented Kalman Filter

Chapter 1

INTRODUCTION

A digital sensor converts observations of physical quantities into a digital signal. The digital signals are required to be processed to extract meaningful information contained within. Consider the scenario where an object or multiple objects of interest are observed by a digital sensor or a network of digital sensors. The task of estimating characteristics that describe the object or objects, e.g. the location of an object, from the data collected is referred to as object tracking. Object tracking methods have been utilised in many different applications, including cell tracking in biology [40], pedestrian tracking in surveillance [115], and aircraft tracking with radar in defence [22]. A wide variety of object tracking methods exist and have been heavily researched [84, 30, 78]. The majority of methods are focused on a Bayesian framework. This is a probabilistic framework which is a natural way of taking uncertainties in the motion and sensor characteristics into account. Typically, complexities in the probabilistic models lead to sub-optimal or computationally expensive solutions. Recently, this has been compounded by advancements in technology which have lead to the availability of a wide range of sophisticated digital sensors. The result is an abundance of sensors capable of transmitting large quantities of data. The focus of this thesis is on the development of novel computationally efficient methods for object tracking when the probabilistic models are afflicted with three specific types of complexity:

1. A large number of objects are required to be tracked by a sensor or network of sensors.

2. Complexities in the received digital signal, e.g. highly non-linear relationships between the signal and the characteristics of an object, or false measurements

that do not originate from the object.

3. The presence of large amounts of data. There may be many low cost sensors thus resulting in a large amount of data to process, or data “rich” sensors which provide large amounts of data about the objects of interest and the environment.

1.1 Outline

The structure of the thesis is outlined below:

Chapter 1 provides the purpose and motivation for the research presented in this thesis, followed by the outline and key contributions of the thesis, and finally the author’s relevant publications.

Chapter 2 introduces the object tracking problem and how it can be approached within a Bayesian framework. A review of several fundamental algorithms is presented. An overview of key methods used for multiple object tracking, focusing on small groups and large groups of objects, is included.

Chapter 3 begins with an introduction to the problem of tracking a large number of objects which follow a certain pattern of motion, referred to as a crowd, and the inference process is formulated within a Bayesian framework. This is followed by an introduction to interval analysis and the box particle filter (Box PF) for point object tracking. Building on that foundation, an efficient Box PF for crowd tracking is developed. Next a brief introduction to the convolution particle filter (CPF) is presented. This is followed by the development of an efficient CPF for crowd tracking. Finally, the performance of the methods is evaluated for two different cases. The first case corresponds to the fully matched scenario where the models used by the methods directly match that used by the simulator, and the second case is an unmatched scenario of a realistic crowd moving through a bottleneck.

Chapter 4 focuses on the related problem of extended object tracking. In comparison with Chapter 3, this chapter considers the scenario where multiple extended objects, which may appear or disappear, are required to be tracked. A Box PF formulation for this challenging problem is presented. Finally, a performance comparison of the developed Box PF, with the border parameterised particle filter (BP PF), is presented over a challenging real dataset based on laser rangefinder measurements.

Chapter 5 begins with an overview of recent advances made in sequential Markov chain Monte Carlo (SMCMC) for object tracking. The focus then shifts to reducing the computational burden in situations where an exhaustively large amount of measurements are observed. The computational complexity for processing the measurements increases significantly with an increase in data. Two novel approaches for reducing this computational burden are presented. The methods based on these approaches achieve computational efficiency while maintaining accurate estimates. The first method achieves this through the introduction of adaptive subsampling in the SMCMC framework; the second and third methods, by merging the expectation propagation (EP) algorithm with the SMCMC and particle filter (PF) frameworks. The performance of the proposed methods is explored through three detailed examples.

Chapter 6 concludes the thesis with a synopsis of the presented methods and a discussion of the key results. Avenues for future research are proposed.

1.2 Key Contributions

Here the significant contributions of the thesis are outlined according to chapters of appearance and are linked to the author's relevant publications listed in Section 1.3.

Chapter 3 - The contributions in this chapter revolve around the development of a Box PF and CPF for crowd tracking [P1]:

- For the first time a generalised likelihood function for crowd tracking in clutter is introduced for the development of the Box PF for crowd tracking. Previous formulations of the Box PF are related to point target tracking.

- The formulated crowd tracking Box PF represents the posterior state probability distribution function (pdf) by a mixture of uniform distributions. The number of components in the mixture distribution is shown to grow with time. An efficient approximation of the Box PF, based on the introduction of the relaxed intersection, is proposed to cope with the growth of mixture components.

- A method to jointly estimate the crowd and clutter measurement rates within the Box PF framework is proposed.
- Development of an adaptive CPF for crowd tracking which is able to resolve the data association problem without the measurement rates.
- Extensive comparisons of the developed filters with a state of the art PF with both a matched rectangular simulator and a realistic crowd simulator.

Chapter 4 - The contributions of this chapter focus on the extension of the Box PF for multiple extended object tracking [P5]:

- A theoretical proof of the generalised likelihood for multiple extended objects in the presence of clutter is given based on a binomial expansion.
- The derived generalised likelihood is incorporated into the Bayesian framework, including a birth/disappearance model for the derivation of the Box PF for multiple extended objects.
- The formulated multiple extended object tracking Box PF represents the posterior state pdf by a mixture of uniform distributions. The number of components in the mixture distribution is shown to grow with time. An efficient approximation of the Box PF, based on the introduction clustering, the relaxed intersection and resampling, is proposed to curb the growth of mixture components.
- Extensive comparisons of the developed Box PF with a state of the art PF on real data from laser rangefinder sensors.

Chapter 5 - The contributions of this chapter focus on the development of methods for efficient object tracking with tall data:

- Adaptive subsampling has been shown to be a promising technique for the processing of tall data in static systems. In this thesis adaptive subsampling was integrated into a SMCMC framework. [P3], [P6].

- In contrast to subsampling approaches, divide and conquer approaches process batches of data in parallel. The Expectation Propagation algorithm is a powerful tool which is integrated into SMCMC [P6] and PF [P2] frameworks to enable parallel processing of the data.

- Extensive comparisons of the developed filters with state of the art filters for simulations scenarios including: multiple object tracking with a single sensor [P6], and object tracking in a sensor network [P2].

1.3 Publications

The author's publications with relevance to this thesis are outlined below:

Peer Reviewed Journal Publications

[P1] A. De Freitas, L. Mihaylova, A. Gning, D. Angelova, V. Kadirkamanathan, "Autonomous crowds tracking with box particle filtering and convolution particle filtering", Automatica, vol. 69, pp. 380-394, July 2016.

Peer Reviewed Conference Proceedings

[P2] A. De Freitas, L. Mihaylova,"Dealing with Massive Data with a Distributed Expectation Propagation Particle Filter for Object Tracking", Proceedings of the 19th International Conference on Information Fusion, July 2016, pp. 457-463.

[P3] A. De Freitas, F. Septier, L. Mihaylova, S. Godsill, "How Can Subsampling Reduce Complexity in Sequential MCMC Methods and Deal with Big Data in

Target Tracking?", Proceedings of the 18th International Conference on Information Fusion, July 2015, pp. 134 - 141.

[P4] N. Petrov, L. Mihaylova, A. De Freitas, "Crowd tracking with box particle filtering", Proceedings of the 17th International Conference on Information Fusion, July 2014, pp. 1 - 7.

Journal Publications Under Review

[P5] A. De Freitas, L. Mihaylova, A. Gning, M. Schikora, M. Ulmke, D. Angelova, W. Koch, "A Box Particle Filter Method for Tracking of Multiple Extended Objects", Submitted to IEEE Transactions on Signal Processing, 2016.

[P6] A. De Freitas, F. Septier, L. Mihaylova, "Sequential Markov Chain Monte Carlo for Bayesian Filtering with Massive Data", Submitted to IEEE Transactions on Signal Processing, 2016.

Conference Papers Under Review

[P7] A. De Freitas, C. Fritsche, L. Mihaylova, F. Gunnarsson, "A Novel Measurement Processing Approach to the Parallel Expectation Propagation Unscented Kalman Filter", Submitted to 20th Int. Conf. on Information Fusion, 2017.

Chapter 2

LITERATURE REVIEW

Object tracking consists of the inference of the unknown characteristics of an object from measurements collected by a single sensor or multiple sensors. In this chapter, an overview of methods for object tracking is presented. Methods which focus specifically on the problem of tracking multiple objects, and handling large amounts of sensor data, are reviewed in Sections 2.3 and 2.4, respectively.

2.1 The Classical Approach

In the classical approach, a discrete state space model is used to model the motion of the object and model the relationship between the measurements and the unknown characteristics of interest. The unknown characteristics of interest are referred to as the hidden states, and are represented by a hidden Markov process, $\boldsymbol{x}_k \in \mathbb{R}^{n_{\boldsymbol{x}}}$, with $n_{\boldsymbol{x}}$ the dimension of the state vector, $k = 0, ..., T \in \mathbb{N}$ represents the discrete time index, and T is the final time step. A transition equation describes how the hidden states evolve over time:

$$\boldsymbol{x}_k = f(\boldsymbol{x}_{k-1}, \boldsymbol{\eta}_k), \tag{2.1}$$

where $\boldsymbol{\eta}_k$ represents a stochastic variable modelling the noise disturbances in the state dynamics, and $f(\,\cdot\,)$ is typically a non-linear function. At each discrete time step k, a set of measurements are available, $\boldsymbol{z}_k \in \mathbb{R}^{n_{\boldsymbol{z}}}$, with $n_{\boldsymbol{z}}$ the dimension of the measurement vector. An observation equation describes the relationship between the measurements and the hidden states:

$$\boldsymbol{z}_k = h(\boldsymbol{x}_k, \boldsymbol{\xi}_k), \tag{2.2}$$

where $\boldsymbol{\xi}_k$ represents a stochastic variable modelling the noise disturbances in the measurements, and $h(\,\cdot\,)$ is typically a non-linear function. A probabilistic equivalent of (2.1) and (2.2) is given by

$$\begin{aligned} \boldsymbol{x}_k &\sim p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}), \\ \boldsymbol{z}_k &\sim p(\boldsymbol{z}_k|\boldsymbol{x}_k), \end{aligned} \tag{2.3}$$

where $\sim$ is the sampling operator, $p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$ and $p(\boldsymbol{z}_k|\boldsymbol{x}_k)$ represent the transition pdf and likelihood function, respectively. The most pertinent information about the hidden state $\boldsymbol{x}_k$, given all the measurements up to and including the current time step, $\boldsymbol{z}_{1:k} = (\boldsymbol{z}_1, \ldots \boldsymbol{z}_k)$, is given by the filtering posterior state pdf, $p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k})$.

The problem of sequentially updating the filtering posterior state pdf can be overcome when the filtering posterior state pdf at the previous time step, $p(\boldsymbol{x}_{k-1}|\boldsymbol{z}_{1:k-1})$, is known. This is achieved through a two step procedure [8] referred to as the optimal Bayesian solution. The first step utilises the transition density to obtain a predicted filtering posterior state pdf:

$$p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1}) = \int p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})p(\boldsymbol{x}_{k-1}|\boldsymbol{z}_{1:k-1})d\boldsymbol{x}_{k-1}. \tag{2.4}$$

The second step utilises the likelihood function to obtain the filtering posterior state pdf:

$$p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k}) = \frac{p(\boldsymbol{z}_k|\boldsymbol{x}_k)p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1})}{p(\boldsymbol{z}_k|\boldsymbol{z}_{1:k-1})}, \tag{2.5}$$

where $p(\boldsymbol{z}_k|\boldsymbol{z}_{1:k-1})$ is a normalisation factor. Unfortunately, the optimal Bayesian solution is rarely available in an analytical form. The most notable case is referred to as the Kalman filter (KF) [68]. The KF is the optimal Bayesian solution when the state space model is linear and perturbed by Gaussian noise. Under these conditions, it can be shown that the filtering posterior state pdf takes on the form of a Gaussian distribution. Thus the KF simply updates the mean and covariance of the filtering posterior state pdf given the measurements at each time step. It is also attractive due to its computationally efficient implementation.

However, complex systems are typically non-linear and/or contain non-Gaussian noise. There have been several extensions of the KF for these cases. The two most notable techniques are referred to as the extended Kalman filter (EKF) [11] and unscented Kalman filter (UKF) [123]. In general, techniques based on the EKF take the approach of linearising the state space model. This then allows for the straightforward application of the KF. The UKF takes a different approach by using a deterministically selected set of sample points that capture the mean and covariance of the Gaussian distribution. These points are then propagated through the non-linear state space model. Unfortunately both of these techniques have limitations in the degree of non-linearity with which they can operate. They are also still based on the underlying assumption of the presence of Gaussian noise in the state space model which leads to a uni-modal filtering posterior state pdf.

Alternative approaches which are more robust to non-linearities and non-Gaussian noise have been developed and are discussed in the following sections.

2.1.1 Monte Carlo Techniques

In Monte Carlo (MC) simulation, the filtering posterior state pdf can be approximated by an unweighted set of samples

$$\hat{p}(\boldsymbol{x}_k|\boldsymbol{z}_{1:k}) = \frac{1}{N}\sum_{j=1}^{N}\delta(\boldsymbol{x}_k - \boldsymbol{x}_k^{(j)}), \tag{2.6}$$

where $\delta(\,\cdot\,)$ is the Dirac delta function, N is the number of samples, (j) the sample index, and $\boldsymbol{x}_k^{(j)} \sim p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k})$. Unlike the KF based approaches, MC based approaches are capable of inference in state space models which are non-linear and/or perturbed by non-Gaussian noise. The problem is in obtaining the samples from the filtering posterior state pdf. Two distinct approaches have been developed as solutions to this problem and are further discussed.

Sequential Monte Carlo Approach

The sequential MC (SMC) approach [38] refers to all the techniques which rely on importance sampling as a mechanism for obtaining samples from the filtering posterior state pdf. In this case a weighted set of samples represent the full posterior state pdf

$$\hat{p}(\boldsymbol{x}_{0:k}|\boldsymbol{z}_{1:k}) = \sum_{j=1}^{N} w_k^{(j)} \delta(\boldsymbol{x}_{0:k} - \boldsymbol{x}_{0:k}^{(j)}), \tag{2.7}$$

where $w_k^{(j)}$ represents the normalised weights, i.e $\sum_{j=1}^{N} w_k^{(j)} = 1$. Importance sampling is a technique for obtaining samples from the unknown posterior state pdf indirectly by sampling a known proposal distribution. The purpose of the weights is to correct for the mismatch between the proposal distribution and the posterior state pdf,

$$w_k^{(j)} \propto \frac{p(\boldsymbol{x}_{0:k}^{(j)}|\boldsymbol{z}_{1:k})}{q(\boldsymbol{x}_{0:k}^{(j)}|\boldsymbol{z}_{1:k})} \tag{2.8}$$

where $\boldsymbol{x}_{0:k}^{(j)} \sim q(\boldsymbol{x}_{0:k}|\boldsymbol{z}_{1:k})$, and $q(\,\cdot\,)$ represents the proposal distribution.

If the proposal distribution can be factored into the following form

$$\begin{aligned} q(\boldsymbol{x}_{0:k}|\boldsymbol{z}_{1:k}) &= q(\boldsymbol{x}_k|\boldsymbol{x}_{0:k-1}, \boldsymbol{z}_{1:k}) q(\boldsymbol{x}_{0:k-1}|\boldsymbol{z}_{1:k-1}), \\ &= q(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, \boldsymbol{z}_k) q(\boldsymbol{x}_{0:k-1}|\boldsymbol{z}_{1:k-1}) \end{aligned} \tag{2.9}$$

then it has been shown [8] that the weights can be sequentially updated according to

$$w_k^{(j)} \propto w_{k-1}^{(j)} \frac{p(\boldsymbol{z}_k|\boldsymbol{x}_k^{(j)}) p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^{(j)})}{q(\boldsymbol{x}_k^{(j)}|\boldsymbol{x}_{k-1}^{(j)}, \boldsymbol{z}_k)}. \tag{2.10}$$

With a sequential update of the weights, it is thus possible to obtain a weighted approximation of the filtering posterior state pdf

$$\hat{p}(\boldsymbol{x}_k|\boldsymbol{z}_{1:k}) = \sum_{j=1}^{N} w_k^{(j)} \delta(\boldsymbol{x}_k - \boldsymbol{x}_k^{(j)}). \tag{2.11}$$

This approach is referred to as the sequential importance sampling (SIS) algorithm and is illustrated in Algorithm 1.

Algorithm 1 Sequential Importance Sampling

```
Initialise particle set: $\left\{\boldsymbol{x}_0^{(j)}\right\}_{j=1}^N$ and weights: $\left\{w_0^{(j)} = \frac{1}{N}\right\}_{j=1}^N$
for $k = 1, ..., T$ do
    for $j = 1, ..., N$ do
        Sample $\boldsymbol{x}_k^{(j)} \sim q_k(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^{(j)}, \boldsymbol{z}_k)$
        Update the importance weights according to (2.10).
    end for
    $\widehat{p}(\boldsymbol{x}_k|\boldsymbol{z}_{1:k}) = \sum_{j=1}^N w_k^{(j)}\delta(\boldsymbol{x}_k - \boldsymbol{x}_k^{(j)})$
end for
```

Although theoretically sound, the SIS algorithm has been found to suffer for larger values of T. This is due to the majority of importance weights tending towards a value of zero. This occurs because the SIS algorithm is essentially sampling from a space with dimensionality linked to k. Sampling from a space with increasing dimensionality with a fixed sample size is expected to fail. This problem is referred to as weight degeneracy.

To overcome the weight degeneracy problem, it was proposed [100, 56] to introduce a resampling step in the algorithm. This version of the algorithm is commonly referred to as the sequential importance sampling resampling (SISR) PF. The resampling step can be implemented in several different ways, but it is typically weighted sampling with replacement which results in the removal of the samples, also referred to as particles, with low weights and duplication of particles with high weights. However, this procedure can also result in a phenomenon called sample impoverishment. Sample impoverishment refers to when certain particles are too highly favoured resulting in a large number of duplicated particles. In the extreme case, the particles may all be duplicates of a single particle. To prevent sample impoverishment, it was proposed [75] to only apply the resampling step when severe weight degeneracy occurs. A commonly used measure of weight degeneracy is the effective sample size (ESS), also referred to as the number of given by

$$ESS = \frac{1}{\sum_{j=1}^{N}\left(w_k^{(j)}\right)^2}. \tag{2.12}$$

The ESS value depicts the number of informative particles. Thus, the ESS should ideally be equal to the total number of particles. The SISR PF is illustrated by Algorithm 2. It is worth mentioning that after resampling, the SISR PF approximation of the filtering posterior state pdf is equivalent to (2.6).

Algorithm 2 Sequential Importance Sampling Resampling Particle Filter

```
Initialise particle set: $\left\{\boldsymbol{x}_0^{(j)}\right\}_{j=1}^{N}$ and weights: $\left\{w_0^{(j)} = \frac{1}{N}\right\}_{j=1}^{N}$
for $k = 1, ..., T$ do
    for $j = 1, ..., N$ do
        Sample $\boldsymbol{x}_k^{(j)} \sim q_k(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^{(j)}, \boldsymbol{z}_k)$
        Update the importance weights according to (2.10).
    end for
    Check for weight degeneracy using a measure such as (2.12).
    if Weight degeneracy detected then
        Resample $\left\{\boldsymbol{x}_k^{(j)}\right\}_{j=1}^{N}$, and reset the weights: $\left\{w_k^{(j)} = \frac{1}{N}\right\}_{j=1}^{N}$
    end if
    $\widehat{p}(\boldsymbol{x}_k|\boldsymbol{z}_{1:k}) = \sum_{j=1}^{N} w_k^{(j)}\delta(\boldsymbol{x}_k - \boldsymbol{x}_k^{(j)})$
end for
```

The successful implementation of the SISR PF is largely dependent on two assumptions. The first assumption is that the filtering posterior state pdf can be sufficiently approximated by a discrete set of weighted samples, i.e $p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k}) \approx \hat{p}(\boldsymbol{x}_k|\boldsymbol{z}_{1:k})$. This is dependent on a variety of factors such as the number of particles, N, and the initialisation of the SISR PF. The second assumption is that samples from the proposal distribution with importance sampling corrections represent samples from the filtering posterior state pdf. The only real criterion for this to be valid would be that the support of the proposal distribution and filtering posterior state pdf should coincide. However, to minimise the amount of weight degeneracy and sample impoverishment, a proposal distribution which minimises the variance of the importance weights should be selected [39]. This is given by

$$q(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^{(j)}, \boldsymbol{z}_k) = p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^{(j)}, \boldsymbol{z}_k). \tag{2.13}$$

Substituting this into (2.10) results in the following weight update

$$w_k^{(j)} \propto w_{k-1}^{(j)} \int p(\boldsymbol{z}_k|\boldsymbol{x}_k) p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^{(j)}) d\boldsymbol{x}_k. \tag{2.14}$$

This requires sampling from the ideal proposal distribution and the integration in the aforementioned equation. However, an analytical form rarely exists for the solution of this integral. There are several different alternative approaches for selecting the proposal distribution. Generally these variations are based either on sub-optimal proposal distributions, such as [56] which proposed utilising the state transition density as the importance distribution, known as the sequential importance resampling (SIR) PF, or approximations of the optimal proposal distribution, such as [116].

Sequential Markov Chain Monte Carlo Approach

Although SMC approaches such as the PF are used in a wide variety of applications, the issues introduced by importance sampling can lead to poor performance. This was one of the primary motivations for the development of SMCMC approaches which do not require importance sampling.

Markov chain Monte Carlo (MCMC) methods work by constructing a Markov chain with a desired distribution, also referred to as the target distribution, as the equilibrium distribution. Two popular MCMC methods used for sampling from a multivariate probability distribution, $\pi(\boldsymbol{x})$, are the Metropolis-Hastings (MH) algorithm [62] and Gibbs sampling [46]. The MH algorithm first generates a sample from a known proposal distribution, $\boldsymbol{x}^* \sim q(\,\cdot\,|\boldsymbol{x}^{m-1})$. The proposed sample is accepted as the current state of the chain, $\boldsymbol{x}^m$, if the following condition is satisfied

$$u < \frac{\pi(\boldsymbol{x}^*) q(\boldsymbol{x}^{m-1}|\boldsymbol{x}^*)}{\pi(\boldsymbol{x}^{m-1}) q(\boldsymbol{x}^*|\boldsymbol{x}^{m-1})}, \tag{2.15}$$

where u represents a sample from a uniform random variable, $u \sim U_{[0,1]}$. The previous state of the chain is stored as the current state, $\boldsymbol{x}^m = \boldsymbol{x}^{m-1}$, when the proposed sample does not meet this criterion. The MH algorithm is illustrated in Algorithm 3.

Algorithm 3 Metropolis-Hastings Algorithm

```
Initialise Markov chain: $\boldsymbol{x}_0 \sim q_0(\,\cdot\,)$
for $m = 1,...,N$ do
    Propose $\boldsymbol{x}^* \sim q\left(\boldsymbol{x}|\boldsymbol{x}^{m-1}\right)$
    Compute $\rho = \min\left(1, \frac{\pi(\boldsymbol{x}^*)q\left(\boldsymbol{x}^{m-1}|\boldsymbol{x}^*\right)}{\pi(\boldsymbol{x}^{m-1})q(\boldsymbol{x}^*|\boldsymbol{x}^{m-1})}\right)$
    Accept $\boldsymbol{x}^m = \boldsymbol{x}^*$ with probability $\rho$, else $\boldsymbol{x}^m = \boldsymbol{x}^{m-1}$.
end for
$\widehat{\pi}(\boldsymbol{x}) = \frac{1}{N}\sum_{j=1}^{N}\delta(\boldsymbol{x} - \boldsymbol{x}^{(j)})$
```

In contrast, the Gibbs sampler is not based on an accept or reject mechanism. Instead, an approximation is obtained by sampling from the marginal distributions of the target distribution with the conditioned variables fixed at their current values. The Gibbs sampler is illustrated in Algorithm 4.

Algorithm 4 Gibbs Sampling

```
Initialise Markov chain: $\boldsymbol{x}_0 \sim q_0(\,\cdot\,)$, where $\boldsymbol{x} = (x_1, x_2, ..., x_{N_d})^\top$.
for $m = 1,...,N$ do
    $x_1^m \sim \pi\left(x_1|x_2^{m-1}, ..., x_{N_d}^{m-1}\right)$
    $x_2^m \sim \pi\left(x_2|x_1^{m}, ..., x_{N_d}^{m-1}\right)$
    $\vdots$
    $x_{N_d}^m \sim \pi\left(x_{N_d}|x_1^{m}, ..., x_{N_d-1}^{m}\right)$
end for
$\widehat{\pi}(\boldsymbol{x}) = \frac{1}{N}\sum_{j=1}^{N}\delta(\boldsymbol{x} - \boldsymbol{x}^{(j)})$
```

The accuracy of the approximation of the target distribution by MCMC methods increases with an increase in the number of MCMC iterations, N. However, increasing N increases the computational costs of the methods. It is also suggested to ignore a number of initial samples, to remove the bias from the initial starting point, referred to as the burn-in period.

In [69] it was proposed to assign the filtering posterior state pdf as the equilibrium distribution, and to use the MH algorithm to obtain samples that approximate the distribution. This allows for the iterative update of an approximation of the filtering posterior distribution by representing $p(\boldsymbol{x}_{k-1}|\boldsymbol{z}_{1:k-1})$ with a set of unweighted particles,

$$p(\boldsymbol{x}_{k-1}|\boldsymbol{z}_{1:k-1}) \approx \frac{1}{N}\sum_{j=1}^{N}\delta(\boldsymbol{x}_{k-1} - \boldsymbol{x}_{k-1}^{(j)}). \tag{2.16}$$

Substituting this result in (2.4) and (2.5) results in the following approximation for the filtering posterior state pdf:

$$\hat{p}(\boldsymbol{x}_k|\boldsymbol{z}_{1:k}) = \frac{1}{N}p(\boldsymbol{z}_k|\boldsymbol{x}_k)\sum_{j=1}^{N_p}p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^{(j)}). \tag{2.17}$$

This procedure is illustrated by Algorithm 5.

Algorithm 5 Sequential Markov Chain Monte Carlo

```
Initialise particle set: $\{\boldsymbol{x}_0^{(j)}\}_{j=1}^{N}$
for $k = 1,...,T$ do
    for $m = 1,...,N + N_b$ do
        Propose $\boldsymbol{x}_k^* \sim q\left(\boldsymbol{x}_k|\boldsymbol{x}_k^{m-1}\right)$
        Compute $\rho = \min\left(1, \frac{p(\boldsymbol{x}_k^*|\boldsymbol{z}_{1:k})q\left(\boldsymbol{x}_k^{m-1}|\boldsymbol{x}_k^*\right)}{p(\boldsymbol{x}_k^{m-1}|\boldsymbol{z}_{1:k})q\left(\boldsymbol{x}_k^*|\boldsymbol{x}_k^{m-1}\right)}\right)$
        Accept $\boldsymbol{x}_k^m = \boldsymbol{x}_k^*$ with probability $\rho$, else $\boldsymbol{x}_k^m = \boldsymbol{x}_k^{m-1}$ .
    end for
end for
$\widehat{p}(\boldsymbol{x}_k|\boldsymbol{z}_{1:k}) = \frac{1}{N}\sum_{j=N_b+1}^{N+N_b}\delta(\boldsymbol{x}_k - \boldsymbol{x}_k^{(j)})$
```

2.2 *The Random Set Statistics Approach*

Tracking of multiple individual objects introduces several challenges which the classical Bayesian approach does not implicitly address. In a multiple object scenario, the received measurements are typically unlabelled, i.e there is no knowledge of which object generated each measurement. This is referred to as the data association problem. Additional complexity in the measurements may exist in the form of clutter, which are measurements which do not contain any information about the objects, and conversely, measurements from objects may not be present due to occlusions or missed detections. The number of objects at each time step may also vary due to the objects leaving or entering the range of the sensors used to observe the objects. The random set statistics approach was developed to directly address these challenges.

The random set statistics approach makes use of random finite sets (RFS). In a RFS, the number of points is random as well as the points themselves. The points are also distinct and unordered. A RFS can be completely specified by a discrete distribution that characterises the number of points, also referred to as the cardinality, and a family of symmetric joint distributions that characterise the distribution of the points conditional on the cardinality [79]. Suppose $\boldsymbol{x}_{n,k}$ represents the state of object n at discrete time step k taking a value in the state space $\mathcal{X} \subseteq \mathbb{R}^{n_{\boldsymbol{x}}}$, and $\boldsymbol{z}_{m,k}$ represents the mth measurement at discrete time step k taking a value in the observation space $\mathcal{Z} \subseteq \mathbb{R}^{n_{\boldsymbol{z}}}$. Then the multi-object and multi-measurement state and observation RFSs are defined as:

$$\begin{aligned} \boldsymbol{X}_k &= \{\boldsymbol{x}_{1,k}, ..., \boldsymbol{x}_{N_k,k}\} \in \mathcal{F}(\mathcal{X}), \\ \boldsymbol{Z}_k &= \{\boldsymbol{z}_{1,k}, ..., \boldsymbol{z}_{M_k,k}\} \in \mathcal{F}(\mathcal{Z}), \end{aligned} \tag{2.18}$$

where $\mathcal{F}(\mathcal{X})$ and $\mathcal{F}(\mathcal{Z})$ are finite subspaces containing space and measurement vectors, respectively of $\mathcal{X}$ and $\mathcal{Z}$, and N_k and M_k represent the number of objects and measurements at time k, respectively.

Analogous to (2.4) and (2.5), the multi-object Bayes recursion can be formulated as [121]

$$\pi(\boldsymbol{X}_k|\boldsymbol{Z}_{1:k-1}) = \int \Pi(\boldsymbol{X}_k|\boldsymbol{X}_{k-1})\pi(\boldsymbol{X}_{k-1}|\boldsymbol{Z}_{1:k-1})\delta\boldsymbol{X}_{k-1}, \tag{2.19}$$

$$\pi(\boldsymbol{X}_k|\boldsymbol{Z}_{1:k}) = \frac{\Phi(\boldsymbol{Z}_k|\boldsymbol{X}_k)\pi(\boldsymbol{X}_k|\boldsymbol{Z}_{1:k-1})}{\Phi(\boldsymbol{Z}_k|\boldsymbol{X}_k)\pi(\boldsymbol{X}_k|\boldsymbol{Z}_{1:k-1})\delta\boldsymbol{X}_k}, \tag{2.20}$$

where $\pi(\,\cdot\,)$ represents the multi-object filtering posterior distribution, $\Pi(\,\cdot\,)$ denotes the multi-object transition density, and $\Phi(\,\cdot\,)$ is the multi-object likelihood density. Due to the introduction of RFSs, the notion of integration and densities is different compared to the classical approach. A full description of RFS operations and densities can be found in [79]. The multi-object filtering posterior distribution is typically intractable. As such, approximations of the multi-object filtering posterior distribution have been developed.

The Probability Hypothesis Density Filter

Since propagating the full multi-object posterior density in time is intractable, it has been proposed [77] to propagate the first order statistical moment of the multi-object posterior state, also referred to as the probability hypothesis density (PHD), as an approximation. The PHD is a density function defined on the single object states $\boldsymbol{x} \in \mathcal{X}$. The predictive step for the intensity function, v_k, is given by

$$v_{k|k-1}(\boldsymbol{x}) = \int p_{S,k|k-1}(\boldsymbol{\zeta}) f_{k|k-1}(\boldsymbol{x}|\boldsymbol{\zeta}) v_{k-1}(\boldsymbol{\zeta}) d\boldsymbol{\zeta} + \gamma_k(\boldsymbol{x}), \tag{2.21}$$

where $p_{S,k|k-1}(\boldsymbol{\zeta})$ is the survival probability of the object at time k given the previous state $\boldsymbol{\zeta}$, $f_{k|k-1}(\,\cdot\,|\boldsymbol{\zeta})$ is the single object transition density at time k given the previous state $\boldsymbol{\zeta}$, and $\gamma_k(\,\cdot\,)$ is the intensity of spontaneous birth. The update equation is given by

$$v_k(\boldsymbol{x}) = \left[1 - p_{D,k}(\boldsymbol{x}) + \sum_{z \in Z_k} \frac{p_{D,k}(\boldsymbol{x}) g_k(\boldsymbol{z}|\boldsymbol{x})}{\kappa_k(\boldsymbol{z}) + \int p_{D,k}(\boldsymbol{\zeta}) g_k(\boldsymbol{z}|\boldsymbol{\zeta}) d\boldsymbol{\zeta}}\right] v_{k|k-1}(\boldsymbol{x}), \tag{2.22}$$

where $p_{D,k}(\boldsymbol{x})$ represents the probability of detection given a state $\boldsymbol{x}$ at time k, $g_k(\,\cdot\,|\boldsymbol{x})$ is the likelihood of a measurement given a state $\boldsymbol{x}$ at time k, and $\kappa_k(\,\cdot\,)$ is the intensity of clutter measurements. The following list describes the assumptions in the model which make (2.21) and (2.22) valid [81]:

- The measurements from each object are independent of the other objects.
- The birth RFS and survival RFS are independent.

- The clutter RFS is Poisson and independent of the object states.
- The prior and predicted multiobject RFSs are Poisson.

There is typically no analytical form for the PHD recursion, with the exception being for linear Gaussian multi-object models. For general non-linear multi-object models, an option is to use an SMC implementation of the PHD recursion [121], or alternatively, approximate the PHD with a weighted mixture of Gaussian distributions and use a KF based approach [120].

2.3 Multiple Object Tracking

In multiple object tracking it is required to jointly track a group of objects. A group of objects can be further defined as either a small or large group. The focus in small group tracking is on the tracking of each individual object in the group. In the case where the objects are traveling within a group formation, it may also be of interest to infer the group structure in addition. In large group tracking, the number of objects is considered too large to track each object. This may be due to limited information from sensors i.e. limited sensor resolution, the degeneration of techniques when considering a large number of objects, or due to an overwhelming computational burden. Techniques used for large group tracking assume that the motion of the group is coordinated. In this case, the focus of large group tracking techniques shifts away from tracking the individual objects to tracking the group as a whole as well as inferring other characteristics of the group. This includes estimation of the shape, size and orientation [84]. Figure 2.1 represents a taxonomy of the methods applied to the tracking of multiple objects.

2.3.1 Small Group Tracking

Independent Object Tracking Approaches

Techniques based on the assumption that the motion and measurements from each object are independent are referred to as standard multiple object tracking techniques.

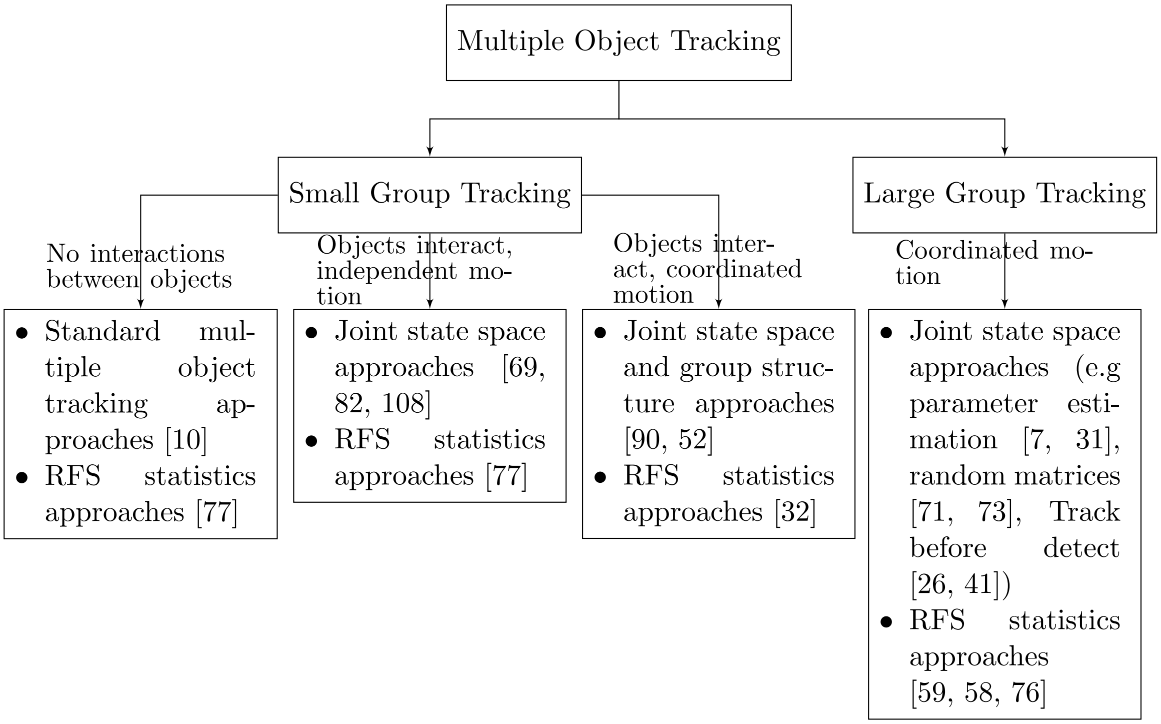

Multiple Object Tracking
Small Group Tracking
Large Group Tracking
No interactions between objects
Objects interact, independent motion
Objects inter-act, coordinated motion
Coordinated motion
• Standard multiple object tracking approaches [10]
• RFS statistics approaches [77]
• Joint state space approaches [69, 82, 108]
• RFS statistics approaches [77]
• Joint state space and group structure approaches [90, 52]
• RFS statistics approaches [32]
• Joint state space approaches (e.g parameter estimation [7, 31], random matrices [71, 73], Track before detect [26, 41])
• RFS statistics approaches [59, 58, 76]

Figure 2.1: Taxonomy of multiple object tracking, adapted from [84].

In this case, each object can be individually tracked by an appropriate filter. There are several traditional approaches which consist of a mechanism for resolving the data association, and tracking the individual objects conditioned on the data association. In this case, the dimensionality of the state vector remains fixed and the number of individual filters may vary depending on if the algorithm is capable of handling a varying number of objects. Classical techniques include: the global nearest neighbour filter [37], the joint probabilistic data association filter [10], and the multiple hypothesis tracking filter [97].

Joint State Space Approaches

An alternative approach is to consider a joint state space model. In this approach the state vector consists of the hidden states for all of the objects. The disadvantage of using a joint state space is that the performance of the techniques identified in Section 2.1.1 are related to the dimensionality of the state space. A notable and well studied case is the PF, which has been shown to require an exponential increase in the number of particles to match performance with an increase in dimensionality of the state space [113]. This places an inherent limit on the number of individual objects which can be tracked.

There has been substantial interest in extending the general PF to applications which consist of a high dimensional state vector. An example is the application of data assimilation in the field of geosciences. The models used are typically non-linear and can consist of millions of dimensions. The success of SMC approaches has been limited due to weight degeneracy. Research has focused mainly on optimising the proposal density [25, 2]. In [2] the equal weight PF was proposed. The strategy of the equal weight PF is to adapt the proposal density in such a manner that the generated particles are equipped with almost equal weights. Several other general approaches were proposed in [116, 93] and more recently [111, 28]. Practically, the optimal proposal distribution can improve filter performance but it does not overcome weight degeneracy in high dimensional models [112]. This is because it does not address the underlying problem that the importance sampling paradigm is inefficient in high dimensional models. It has been proposed to incorporate MCMC algorithms within the PF framework to further increase filter performance in high dimensional systems. In [50] it was proposed to introduce an MH step after resampling. This helps rejuvenate the set of particles by reducing path degeneracy. In [55] it was proposed to combine MCMC with the introduction of intermediate distributions between the prior and likelihood, referred to as a bridging densities. This results in the gradual introduction of the likelihood, which when combined with MCMC techniques, has shown promising results. In [34, 35] the concept of bridging is extended to describe distributions between the prior and posterior, referred to as the particle flow PF. This approach theoretically overcomes the weight degeneracy problem. However, the practical implementation of these methods require many approximations. Another set of approaches are referred to as local PFs [85, 96]. These approaches partition the state space so that individual PFs can be run on the lower dimensional spaces. These techniques rely on the assumption that the state space can be appropriately factorised. Similarly, in [19], an approach referred to as the space-time PF is presented. In this approach the model is assumed to have a weaker factorisation in comparison with local PFs. The likelihood is gradually introduced as the filter moves along both the state space and time index. Another interesting PF variant is the Box PF which combines interval analysis with the PF [1, 54]. In this approach each particle is referred to as a box and has a non-zero volume in the state space. This approach has only been applied to low dimensional systems yet has been identified [84] as a promising candidate for higher dimensional models. The standard SMCMC approach has been shown to be more robust to higher dimensional systems when compared with standard PF approaches. In [109, 44] further improvements have been made to the SMCMC approach to specifically improve performance in high dimensional systems. This is achieved through the introduction of Langevin diffusion and Hamiltonian dynamics.

In object tracking, there are two cases which require a joint state space model. The first case is when the individual object states are related, i.e. dropping the assumption of object motion independence. This allows for more elaborate and accurate models for the motion. This is normally taken into account through the object state dynamics. In [69] and [115] this was done through the introduction of Markov random fields and the social force model, respectively. The second case is when the likelihood is in a form which requires the joint evaluation of all the measurements. This typically has the advantage of implicitly solving the data association problem with the disadvantage of requiring a joint state space. A common likelihood which meets this criteria is the generalised likelihood proposed in [49]. It is applicable in models which consist of a Poisson point process for the measurements from the objects and clutter.

An additional challenge associated with the joint state space approach is that the size of the state vector is generally required to be fixed, fixing the number of objects. To overcome the restricted size of the state vector, it was proposed in [108] to set the dimensionality of the state vector to correspond to the maximum number of objects with the introduction of an indicator state which describes if an object is within the region being monitored by the sensor. Another option would be to utilise reversible jump techniques [5] which allow for a variable size of the state vector in an MCMC based approach. The authors in [69, 64] propose the introduction of a variable related to the dimensionality of the state space for each particle, allowing for a variable size of state space in PF applications. The Random set statistics approach [77] utilises RFSs to overcome these challenges at the expense of increasing the complexity of Bayes filter.

Small Groups with Coordinated Motion

An interesting extension is when the motion of the individual objects is coordinated. In addition to unique motion models which capture the relationships between the motion of the individual objects (e.g inspired by complex biological systems [33]), further information can also be inferred about the group of individual objects. An example would be the structure of a coordinated group. Multiple coordinated groups may also be present in the same scenario. This introduces new challenges, such as the merging of two or more coordinated groups, or conversely, a coordinated group could split into multiple coordinated groups. The classical sequential Bayesian formulation in (2.4) and (2.5) can be manipulated to include the sequential inference of an additional parameter describing the structure of a group. In [90] the group structure was sequentially updated with the states of the objects using a SMCMC method. In [52] the group structure was represented as a random graph which was sequentially updated with the states of the objects using a SMC method. The group structure was inferred in [32] using a Gaussian mixture PHD filter which identifies the group objects and utilises a random graph to represent the group structure.

2.3.2 Large Group Tracking

In terms of measurement modelling, large group tracking is essentially identical to the tracking of extended objects. Large groups and extended objects are characterised by their size, in contrast to point objects ([84, 57]) where the whole is approximated with a single point. They generate multiple measurements, where the number varies in time. While tracking point objects has been widely studied, and efficient solutions are developed, the problem of large group and extended object tracking is still challenging and requires new efficient approaches. The methods for large groups and extended object tracking can be broadly classified into several categories: RFS statistics methods (the PHD filter [80, 59], Cardinality PHD filter [76], multi-Bernoulli Filters [18] etc.), SMC and MCMC methods [84], and analytical type of methods [15, 16].

There are also results with different types of data: radar [7], image and video [31], laser range sensors [58], LiDAR data (radioactive clouds [108]) and others.

Various models for the representation of the shape of a group or extended object have been explored. In [7], the shape of an extended object is modelled as an ellipse and the parameters of the ellipse are directly related to the measurements. The concept of a spatial distribution over the group or object extent was introduced in [49], where the parameters specify the region of the spatial distribution. This concept has also been applied in a track-before-detect setting for extended object tracking [26]. In [71], the extent parameter is represented by a random matrix. In [17], the shape of an extended object is described by an implicit function instead of in a parametric form. Similarly, the shape contour describing the extent of an object is modelled with a Gaussian Process in [122].

A specific large group of interest is a crowd of pedestrians with complex coordinated motion. Recent results for the modelling, simulation and visual analysis of crowds are presented in [3] from the point of view of computer vision, transportation systems and surveillance. The social force model [82] has been used to model behaviour of pedestrians, including the evacuation of people through bottlenecks. The social force model has also been combined with some filtering techniques for multiple-target tracking in [91].

2.4 Dealing with Tall Data

Datasets containing massive amounts of data points, referred to as tall datasets [13], can cause excessive computational burdens to traditional statistical methods. Several approaches have been developed to reduce this computational expense, e.g. for regression and classification problems, a random forest approach [47] has been proposed.

In object tracking applications, information rich sensors result in a large amount of data that is required to be processed. In certain applications it may be possible to reduce the amount of data through feature extraction [127]. This is only useful when portions of the data are redundant or non-informative. The focus in this thesis is in dealing with a large amount of informative and non-redundant data. Tracking applications are typically time sensitive. In a Bayesian framework, the filtering posterior state pdf is updated at each time step. This limits the amount of processing time that a tracking algorithm has to process the measurements received at each time step. Naturally, the more measurements received, the more processing time is required to evaluate them.

Research on efficient implementations of SMC methods have focused on making the structure of the PF parallel [27], particularly the resampling step [74], which can then be used in distributed processing applications [95]. However, this typically requires approximations to achieve a solution and still requires the evaluation of all the measurements. Other related research focuses on likelihoods which are complex and do not have an analytical form, such as the approximate Bayesian calculation PF [36] and the CPF [99]. These algorithms require the comparison of all the measurements with sampled measurements.

In MCMC simulation, there have been several different approaches proposed for dealing with large amounts of data. The proposed methods can be categorised as either parallel or sequential strategies.

In terms of parallel strategies, there are two general approaches which have been proposed. The first approach is referred to as blocking. These techniques focus on parallelising specific steps in the MCMC approach. In [114] it was proposed to parallelise the computation of the likelihood. This is restrictive in terms of the model used, and requires a large amount of communication between the processors. The second approach is referred to as divide and conquer. Techniques based on divide and conquer focus on subdividing the measurements and running separate MCMC samplers in parallel on each subdivided set of measurements. The samples from the separate MCMC samplers, referred to as local samples, are then combined to obtain samples from the complete posterior state pdf, referred to as global samples. The divide and conquer techniques differ in how the local samples are combined to obtain the global samples. In [105], global samples are obtained as a weighted average of the local samples. This approach is only theoretically valid under a Gaussian assumption. In [89], the local posterior from the separate MCMC samplers is approximated as Gaussian or with a Gaussian kernel density estimation. Global samples can then be obtained through the product of the local densities. This idea is further developed in [124] by representing the discrete kernel density estimation as a continuous Weierstrass transform. In [87], the combination is based on the geometric median of the local posteriors which are approximated with Weiszfeld's algorithm by embedding the local posteriors in a reproducing kernel Hilbert space. Divide and conquer techniques typically struggle in applications where the local posteriors substantially differ, and if they do not satisfy Gaussian assumptions. In [126, 45] a divide and conquer strategy was proposed which attempts to overcome the challenge of differing local posteriors, and relaxing the Gaussian assumption to a more general assumption of a posterior state pdf from the exponential family. The approach is based on the EP algorithm. In this iterative approach, the separate MCMC samplers exchange sufficient statistics, resulting in each individual MCMC sampler converging to the global posterior.

Sequential strategies rely on subsampling mechanisms, such as pseudo likelihoods [6, 94] or confidence intervals [12, 72], to perform inference using MCMC techniques based only on a certain subsample of all the measurements.

2.5 Summary

In this chapter a literature review covering a wide range of aspects related to object tracking in complex systems was presented. This began with the general Bayesian formulation for object tracking, where the solution is encapsulated by the posterior state pdf. Since model complexities typically result in a lack of availability of an analytical solution, a wide variety of approximate methods have been proposed. These methods approximate the posterior state pdf with a set of discrete samples. The two most common approaches rely on SIS or MCMC to obtain samples that represent the posterior state pdf.

The focus then shifted specifically towards multiple object tracking, where different approaches for small group and large group tracking were considered. This also included a brief overview of an alternative tracking formulation, based on random finite sets. In addition, problems associated with a large number of measurements, referred to as tall data, were explored and approaches specifically developed for dealing with tall data were reviewed.

The computational complexity of object tracking approaches is generally linked to the complexity of the models. For example, in the particle flow filter it is required to numerically integrate complex partial differential equations. In sampling based approaches (such as the PF and SMCMC), the computational complexity increases with the dimensionality of the state space. The Box PF is a computationally efficient variant of the PF which is based on an approximation of the posterior state pdf with a weighted uniform mixture. The computational gain is based on the ability of the Box PF to utilise a significantly smaller number of uniform components relative to the number of particles utilised in standard approaches. This is at the expense of the introduction of specialised bounded arithmetic, referred to as interval analysis. The advantage is that, compared to other basis function filters, such as the Gaussian mixture filter [4], interval analysis can aid in overcoming nonlinearities in the state space model. The CPF is also an efficient variant of the PF that does not require an explicit expression for the likelihood function. However, neither of these filters have been previously considered for the challenging problem of dynamic crowd tracking considered in the following chapter.

Advances in processing of tall data for static MCMC simulation, and the flexible structure of SMCMC for object tracking form the basis of the developments in a later chapter.

Chapter 3

EFFICIENT PARTICLE APPROACHES FOR CROWD TRACKING

In this chapter, the problem of collectively tracking a large number of objects, referred to as a crowd, is considered. Section 3.1 begins with the problem formulation. Inference based on the Bayesian framework for the proposed state space model is presented in Section 3.2. Section 3.3 introduces the fundamentals of interval analysis and the Box PF for tracking point objects without clutter. The Box PF is a combination of interval analysis with the standard PF framework. Two novel methods are developed in Sections 3.4 and 3.6, based on the Box PF and CPF frameworks, respectively, to achieve efficient inference. Finally, numerical studies are presented for a generic group object simulator and realistic crowd simulator in Section 3.7.

3.1 State Space Modelling of a Crowd

By extending the general state space model described in Section 2.1, the characteristics of a crowd of objects and the observed scene that are required to be inferred at each time step k, can be represented by an augmented state vector:

$$\boldsymbol{\zeta}_k = \left(\boldsymbol{\lambda}_k^\top, \boldsymbol{X}_k^\top, \boldsymbol{\Theta}_k^\top\right)^\top, \tag{3.1}$$

where $\boldsymbol{X}_k$ is the kinematic vector of the centre of the crowd, and $\boldsymbol{\Theta}_k$ is the parameter vector which characterises the crowd extent. Multiple measurements are received from the crowd and from clutter at each time step, thus the state vector includes $\boldsymbol{\lambda}_k$ which is the measurement rate vector. The notation $(\,\cdot\,)^\top$ is the transpose operator.

Without loss of generality, the kinematic vector consists of the position coordinates and the velocity of the centre of the crowd, and the extent of the crowd is appoximated by a rectangle. The resulting kinematic vector has the following form:

$$\boldsymbol{X}_k = (x_k, \dot{x}_k, y_k, \dot{y}_k)^\top \tag{3.2}$$

where (x_k, y_k), are the position coordinates, and $(\dot{x}_k, \dot{y}_k)$ are the respective velocity components of the crowd centroid. The parameter vector is given by:

$$\boldsymbol{\Theta}_k = (a_k, b_k)^\top \tag{3.3}$$

where a_k and b_k represent the lengths of the sides of the rectangle in the x and y dimensions, respectively. The measurement rate vector is represented by:

$$\boldsymbol{\lambda}_k = (\lambda_{T,k}, \lambda_{C,k})^\top, \tag{3.4}$$

where $\lambda_{T,k}$ and $\lambda_{C,k}$ represents the crowd and clutter measurement rates, respectively.

Crowd Dynamics Model

The motion of the centre of the crowd is modelled by a correlated velocity model. The correlated velocity model is related to the Singer model [110] and jerk model [83] with the difference being that the velocity component is correlated in time and that the second and other higher order derivatives of position are negligible. The evolution model for the kinematic state of the crowd is represented mathematically by

$$\boldsymbol{X}_k = \boldsymbol{A}\boldsymbol{X}_{k-1} + \boldsymbol{\eta}_k, \tag{3.5}$$

where $\boldsymbol{\eta}_k$ represents the system dynamics noise. The state transition matrix is given by

$$\boldsymbol{A} = \begin{bmatrix} 1 & \frac{1}{\alpha}\left(1 - e^{-\alpha T_s}\right) \\ 0 & e^{-\alpha T_s} \end{bmatrix} \otimes \mathrm{I}_2 \tag{3.6}$$

where T_s is the sampling interval, $\otimes$ denotes the Kronecker product, I_n denotes the $n \times n$ identity matrix, and α is the reciprocal of the velocity correlation time constant. The covariance of the system dynamics noise $\boldsymbol{\eta}_k$ can be modelled as

$$\boldsymbol{Q} = 2\alpha\sigma_v^2 \begin{bmatrix} q_{11} & q_{12} \\ q_{12} & q_{22} \end{bmatrix} \otimes \mathrm{I}_2, \tag{3.7}$$

where σ_v^2 is the variance of the velocity of the crowd centroid for a single dimension and

$$\begin{aligned} q_{11} &= \frac{1}{2\alpha^3}\left(4e^{-\alpha T_s} - 3 - e^{-2\alpha T_s} + 2\alpha T_s\right), \\ q_{12} &= \frac{1}{2\alpha^2}\left(e^{-2\alpha T_s} + 1 - 2e^{-\alpha T_s}\right), \\ q_{22} &= \frac{1}{2\alpha}\left(1 - e^{-2\alpha T_s}\right). \end{aligned} \tag{3.8}$$

The evolution for the crowd extent is assumed to be a random walk model, described by

$$\boldsymbol{\Theta}_k = \boldsymbol{\Theta}_{k-1} + \boldsymbol{\eta}_{p,k}, \tag{3.9}$$

where the parameter noise $\boldsymbol{\eta}_{p,k}$ is zero-mean Gaussian distributed with covariance $\boldsymbol{\Sigma}_\theta$.

3.1.1 Observation Model

In this chapter it is assumed that the digital sensor observes the crowd from a vantage point which results in the origin of measurements from within a confined area. However, other scenarios, such as the case where the measurements only come from the border, are considered in a related problem in Chapter 4.

The total number of measurements M_k, obtained at each time step from the sensor consists of the $M_{T,k}$ number of measurements, originating from the crowd and $M_{C,k}$ clutter measurements, i.e. $M_k = M_{T,k} + M_{C,k}$. The number of measurements $M_{T,k}$ originating from the crowd is considered as a Poisson-distributed random variable with mean value of the crowd rate, $\lambda_{T,k}$, i.e., $M_{T,k} \sim Poisson(\lambda_{T,k})$. Similarly, the number of clutter measurements is $M_{C,k} \sim Poisson(\lambda_{C,k})$. The $M_{T,k}$ measurements originating from the crowd are uniformly located in the area represented by the crowd. The $M_{C,k}$ clutter measurements are uniformly located in the region about the crowd.

In crowd tracking, the measurement equation in (2.2), which directly relates the states to the measurements is not available. The approach followed here is to relate the observations indirectly to the states through the sensor characteristics and the object model. The sensor characteristics describe the relationship between the measurement point m, $m = 1, ..., M_k$ and the measurement source in a Cartesian coordinate system and is of the form:

$$\boldsymbol{z}_{m,k} = \tilde{h}(\boldsymbol{y}_{m,k}) + \boldsymbol{\xi}_k, \tag{3.10}$$

where $\tilde{h}(\cdot)$ is the measurement function and $\boldsymbol{y}_{m,k} = (x_{m,k}, y_{m,k})^\top$ denotes the Cartesian coordinates of the measurement source in a two dimensional space. In this chapter the following model is considered:

$$\boldsymbol{z}_{m,k} = \boldsymbol{H}\boldsymbol{y}_{m,k} + \boldsymbol{\xi}_k, \tag{3.11}$$

where $\boldsymbol{H} = \mathrm{I}_2$, and the measurement noise $\boldsymbol{\xi}_k = (\xi_{1,k}, \xi_{2,k})^\top$, is assumed to be Gaussian, with a known covariance matrix $\boldsymbol{R} = diag(\sigma_1^2, \sigma_2^2)$. The vector of an interval measurement is $[\boldsymbol{z}_{m,k}] = ([z_{1,m,k}], [z_{2,m,k}])^\top$, where $[z_{1,m,k}]$ and $[z_{2,m,k}]$ are the intervals of the m-th measurement point. One way to describe these components is by representing the noise terms in equation (3.10) as intervals:

$$\begin{aligned} [\xi_{1,k}] &= [-3\sigma_1, +3\sigma_1], \\ [\xi_{2,k}] &= [-3\sigma_2, +3\sigma_2]. \end{aligned} \tag{3.12}$$

At each time step k, the M_k interval measurements are combined into a set of intervals $[\boldsymbol{Z}_k] = \{[\boldsymbol{z}_{1,k}], \ldots, [\boldsymbol{z}_{m,k}]\}$.

Each measurement originates from either random clutter or the crowd but its origin is unknown. The object model describes the relationship between the states and the measurement sources for the $M_{T,k}$ measurements that originate from the crowd. As previously described, the measurement sources are uniformly distributed across the region which exhibits measurements, and this region is represented by the states through the following probability density:

$$p(\boldsymbol{y}_{m,k}|\boldsymbol{x}_k) = U_{q(\boldsymbol{x}_k)}(\boldsymbol{y}_{m,k}), \tag{3.13}$$

The support of the uniform distribution describes two independent regions which cover the area of the rectangle used to approximate the extent of the crowd:

$$q(\boldsymbol{x}_k) = \begin{cases} x_k - \frac{a_k}{2} \leq x_{m,k} \leq x_k + \frac{a_k}{2}, \\ y_k - \frac{b_k}{2} \leq y_{m,k} \leq y_k + \frac{b_k}{2}. \end{cases} \tag{3.14}$$

3.2 Inference in a Bayesian Framework for Crowd Tracking

Similarly to the steps in (2.4) and (2.5), the posterior state pdf for the extended state vector described by (3.1) can be updated sequentially based on a prediction step,

$$p(\boldsymbol{\zeta}_k|\boldsymbol{Z}_{1:k-1}) = \int p(\boldsymbol{\zeta}_k|\boldsymbol{\zeta}_{k-1})p(\boldsymbol{\zeta}_{k-1}|\boldsymbol{Z}_{1:k-1})d\boldsymbol{\zeta}_{k-1} \tag{3.15}$$

followed by an update step,

$$p(\boldsymbol{\zeta}_k|\boldsymbol{Z}_{1:k}) = \frac{p(\boldsymbol{Z}_k|\boldsymbol{\zeta}_k)p(\boldsymbol{\zeta}_k|\boldsymbol{Z}_{1:k-1})}{p(\boldsymbol{Z}_k|\boldsymbol{Z}_{1:k-1})}. \tag{3.16}$$

For further notational convenience, the marginal state is defined as follows:

$$\boldsymbol{x}_k = \left(\boldsymbol{X}_k^\top, \boldsymbol{\Theta}_k^\top\right)^\top. \tag{3.17}$$

In this application the posterior state pdf can be further factored into the following form:

$$p(\boldsymbol{\zeta}_k|\boldsymbol{Z}_{1:k}) = p(\boldsymbol{x}_k|\boldsymbol{Z}_{1:k}, \boldsymbol{\lambda}_k)p(\lambda_{T,k}|\boldsymbol{Z}_{1:k})p(\lambda_{C,k}|\boldsymbol{Z}_{1:k}). \tag{3.18}$$

This factorisation implicitly states that the crowd and clutter measurement rates are independent of the kinematics and extent of the crowd. This is true for the clutter measurement rate but not necessarily valid for the crowd measurement rate. However, the variance of the prior distribution for the crowd rate is sufficient to represent the variation of the number of measurements over time.

It has been shown that an analytical recursive Bayesian solution exists for the estimation of the mean of a Poisson distribution, based on using the conjugate prior Gamma distribution [60]. The crowd and clutter measurement rates are estimated based on this concept[1], and the focus of this chapter thus lies on the calculation of the marginal posterior distribution for the states representing the kinematics and extent of the crowd, $p(\boldsymbol{x}_k|\boldsymbol{Z}_{1:k}, \boldsymbol{\lambda}_k)$, using novel Box PF and CPF methods.

3.3 Review of Interval Analysis and the Box Particle Filter

Prior to the introduction of the Box PF based method for crowd tracking, an introduction to general interval analysis and the standard Box PF is presented.

3.3.1 Interval Analysis

A real interval, $[x]$, is defined as a closed and connected subset of the set $\mathbb{R}$:

$$[x] = [\underline{x}, \overline{x}] = \{x \in \mathbb{R} \,|\, \underline{x} \leq x \leq \overline{x}\}, \tag{3.19}$$

where $\underline{x}$ and $\overline{x}$ define the lower and upper limits of the subset and are referred to as the infimum and supremum. The length, or size, of an interval, i.e. $\overline{x} - \underline{x}$, is represented by $|[x]|$. In an n-dimensional space, an interval vector (or box) is denoted by $[\boldsymbol{x}]$ with $\boldsymbol{x} \in \mathbb{R}^n$. This is equivalent to the Cartesian product of n intervals, i.e. $[\boldsymbol{x}] = [x_1] \times [x_2] \times ...[x_n]$. Set-theoretic operations, such as the intersection or union of sets, can be directly applied to boxes.

The standard binary operators, e.g. $\{+, -, \backslash, \times\}$, have been extended to boxes [67] and results in a new box, i.e. $[\boldsymbol{z}] = [\boldsymbol{x}] \diamond [\boldsymbol{y}]$, where $\diamond$ denotes a binary operator.

[1] Refer to Appendix A.3 for more information on crowd and clutter measurement rate estimation.

However, passing a box through a more general function, $\boldsymbol{g}(\,\cdot\,)$, may lead to a region which cannot be described by a box. One approach is to approximate this region with a box which encloses the region. A function with output corresponding to this approximation is referred to as an inclusion function, $[\boldsymbol{g}](\,\cdot\,)$, where $\boldsymbol{g}([\boldsymbol{x}]) \subseteq [\boldsymbol{g}]([\boldsymbol{x}])$. The minimal inclusion function has the tightest possible bound on the region. One of the challenges associated with interval analysis is in finding an inclusion function sufficiently close to the minimal inclusion function at a minimal computational expense [67].

Another challenge of interest is in solving a CSP. Consider a prior box $[\boldsymbol{x}] \subset \mathbb{R}^{n_x}$, and a set of constraints on the state space of the form $\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{0}$. The CSP involves finding the region enclosing the set $[\boldsymbol{x}]$, which satisfies the set of constraints. The CSP $\mathcal{H}$ is formulated as:

$$\mathcal{H} : \left(\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{0}, \boldsymbol{x} \in [\boldsymbol{x}]\right). \tag{3.20}$$

The solution set that satisfies $\mathcal{H}$ is defined as

$$\mathcal{S} : \left(\boldsymbol{x} \in [\boldsymbol{x}] \,|\, \boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{0}\right), \tag{3.21}$$

and may not necessarily be a box. In general, finding $\mathcal{S}$ is computationally intractable. Solving the CSP, in an interval framework, involves finding the smallest box, $[\boldsymbol{x}^*]$, which encloses $\mathcal{S}$, i.e. $\mathcal{S} \subseteq [\boldsymbol{x}^*] \subseteq [\boldsymbol{x}]$. Finding an enclosing box, $[\boldsymbol{x}']$, smaller than the prior box, with $[\boldsymbol{x}^*] \subseteq [\boldsymbol{x}'] \subseteq [\boldsymbol{x}]$, can be achieved through an operation referred to as a contraction, with the optimal contraction resulting in $[\boldsymbol{x}'] = [\boldsymbol{x}^*]$. A contractor for $\mathcal{H}$ is any operator which results in contraction of the prior box. A wide variety of contractors exist [67]. Selection of a contractor is dependent on the type of constraints in $\mathcal{H}$. Computationally, the complexity of a contractor can be kept polynomial in time and space.

3.3.2 The Classic Box Particle Filter

The standard PF represents the posterior state pdf with a set of weighted particles, where each particle has a zero volume in the state space. In contrast, the Box PF consists of a weighted set of box particles, where each box particle represents a region with controllable volume in the state space. The Box PF approximates the posterior state pdf with a mixture of uniform pdfs [51]:

$$p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k}) \approx \sum_{p=1}^{N} w_k^{(p)} U_{[\boldsymbol{x}_k^{(p)}]}(\boldsymbol{x}_k). \tag{3.22}$$

where $U_{[\boldsymbol{x}]}(\,\cdot\,)$ is a multivariate uniform distribution defined on the region of the box $[\boldsymbol{x}]$. Initially at time step k, only an expression for the posterior state pdf at time step $k-1$ is available. The first step to finding the posterior state pdf at time step k consists of finding an expression for the predictive posterior state pdf, $p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1})$, via the time update in (2.4):

$$\begin{aligned} p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1}) &\approx \int p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) \sum_{p=1}^{N} w_{k-1}^{(p)} U_{[\boldsymbol{x}_{k-1}^{(p)}]}(\boldsymbol{x}_{k-1}) d\boldsymbol{x}_{k-1} \\ &= \sum_{p=1}^{N} w_{k-1}^{(p)} \int_{[\boldsymbol{x}_{k-1}^{(p)}]} p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) U_{[\boldsymbol{x}_{k-1}^{(p)}]}(\boldsymbol{x}_{k-1}) d\boldsymbol{x}_{k-1}. \end{aligned} \tag{3.23}$$

An inclusion function, $[f](\,\cdot\,)$, exists for the transition function, $f(\,\cdot\,)$, in (2.1) when the noise is bounded, i.e. $[\boldsymbol{\eta}_k]$. If the pth box particle at time step $k-1$ is defined on the region, $\boldsymbol{x}_{k-1} \in [\boldsymbol{x}_{k-1}^{(p)}]$, then the inclusion function can be used to obtain the region of the box particle at time step k, $\boldsymbol{x}_k \in [f]([\boldsymbol{x}_{k-1}^{(p)}], [\boldsymbol{\eta}_k])$.

In the classical Box PF, each of the terms represented by the integral in (3.23) is approximated by a single uniform pdf component [51],

$$\int_{[\mathbf{x}_{k-1}^{(p)}]} p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) U_{[\boldsymbol{x}_{k-1}^{(p)}]}(\boldsymbol{x}_{k-1}) d\boldsymbol{x}_{k-1} \approx U_{[f]([\boldsymbol{x}_{k-1}^{(p)}],[\boldsymbol{\eta}_k])}(\boldsymbol{x}_k). \tag{3.24}$$

Combining (3.23) and (3.24) gives the predictive posterior state pdf:

$$\begin{aligned} p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1}) &\approx \sum_{p=1}^{N} w_{k-1}^{(p)} U_{[f]([\boldsymbol{x}_{k-1}^{(p)}],[\boldsymbol{\eta}_k])}(\boldsymbol{x}_k) \\ &= \sum_{p=1}^{N} w_{k-1}^{(p)} U_{[\boldsymbol{x}_{k|k-1}^{(p)}]}(\boldsymbol{x}_k). \end{aligned} \tag{3.25}$$

The accuracy of the approximation of each pdf term with a single uniform pdf component may not be sufficient. It has been shown [51] that a more accurate representation can be utilised by approximating each term with a mixture of uniform pdfs if required.

Finally, the posterior state pdf, $p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k})$, can be obtained through the application of the measurement update step to the predictive posterior state pdf in (3.25). In the classical Box PF, it is assumed that the likelihood function in (2.3), $p(\boldsymbol{z}_k|\boldsymbol{x}_k)$, can be represented by a mixture of uniform pdfs. Without loss of generality, it is represented in this section with a single uniform pdf, such that the box measurement $[\boldsymbol{z}_k]$ contains all realisations of (2.2). Accordingly, the likelihood function is then represented by, $p(\boldsymbol{z}_k|\boldsymbol{x}_k) = U_{[\boldsymbol{z}_k]}(h(\boldsymbol{x}_k))$. Given this expression, utilising (2.5) results in the following expression for the posterior state pdf:

$$\begin{aligned} p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k}) &= \frac{1}{\alpha_k} p(\boldsymbol{z}_k|\boldsymbol{x}_k) p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1}) \\ &= \frac{1}{\alpha_k} U_{[\boldsymbol{z}_k]}(h(\boldsymbol{x}_k)) \sum_{p=1}^{N} w_{k-1}^{(p)}\, U_{[\boldsymbol{x}_{k|k-1}^{(p)}]}(\boldsymbol{x}_k), \\ &= \frac{1}{\alpha_k} \sum_{p=1}^{N} w_{k-1}^{(p)}\, U_{[\boldsymbol{z}_k]}(h(\boldsymbol{x}_k))\, U_{[\boldsymbol{x}_{k|k-1}^{(p)}]}(\boldsymbol{x}_k), \end{aligned} \tag{3.26}$$

where α_k is a normalising constant. Each of the terms within the summation is also a constant function with a support being the following region,

$$S_p = \left\{ \boldsymbol{x}_k \in [\boldsymbol{x}_{k|k-1}^{(p)}] \mid h(\boldsymbol{x}_k) \in [\boldsymbol{z}_k] \right\}. \tag{3.27}$$

Equation (3.27) represents a constraint as defined in Section 3.3.1, thus leading to a CSP. A contractor can be utilised to reduce the predicted supports $[\boldsymbol{x}_{k|k-1}^{(p)}]$ from the time update pdf, $p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1})$, with the constraints imposed by the measurement box, $[\boldsymbol{z}_k]$, and likelihood function. The contraction leads to the definition of a new set of box particles, denoted $\{[\boldsymbol{x}_k^{(p)}]\}_{p=1}^N$, which approximate the posterior state pdf $p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k})$ at time k. Thus, the posterior state pdf expression in (3.26) can be further developed to take into consideration the contracted box particles:

$$\begin{aligned} p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k}) &= \frac{1}{\alpha_k}\sum_{p=1}^N w_{k-1}^{(p)} \frac{1}{|[\boldsymbol{z}_k]|}\frac{1}{|[\boldsymbol{x}_{k|k-1}^{(p)}]|}||S_p||U_{S_p}(\boldsymbol{x}_k) \\ &\approx \frac{1}{\alpha_k}\sum_{p=1}^N w_{k-1}^{(p)} \frac{1}{|[\boldsymbol{z}_k]|}\frac{1}{|[\boldsymbol{x}_{k|k-1}^{(p)}]|}|[\boldsymbol{x}_k^{(p)}]|U_{|[\boldsymbol{x}_k^{(p)}]|}(\boldsymbol{x}_k) \\ &\propto \sum_{p=1}^N w_{k-1}^{(p)} \frac{|[\boldsymbol{x}_k^{(p)}]|}{|[\boldsymbol{x}_{k|k-1}^{(p)}]|}U_{|[\boldsymbol{x}_k^{(p)}]|}(\boldsymbol{x}_k). \end{aligned} \tag{3.28}$$

Comparing the posterior state pdf expressions in (3.22) and (3.28), the relationship between the weights at time step $k-1$ and k is

$$w_k^{(p)} \propto w_{k-1}^{(p)}\frac{|[\boldsymbol{x}_k^{(p)}]|}{|[\boldsymbol{x}_{k|k-1}^{(p)}]|}. \tag{3.29}$$

In summary, the posterior state pdf is approximated by $\{(w_k^{(p)}, [\boldsymbol{x}_k^{(p)}])\}_{p=1}^N$.

3.4 The Box Particle Filter for Crowd Tracking

The prediction step for the crowd tracking Box PF follows the same spirit as described by equations (3.23) to (3.25). However, when dealing with multiple object originated measurements and clutter measurements, the update step is required to be re-derived. The generalised likelihood for the Poisson rate measurement and clutter model described in Section 3.1.1 is given by [49]

$$, \quad \begin{aligned} p(\boldsymbol{Z}_k, \boldsymbol{\lambda}_k|\boldsymbol{x}_k) &= \prod_{m=1}^{M_k}\left(1+\frac{\lambda_{T,k}}{\rho_k}p(\boldsymbol{z}_{m,k}|\boldsymbol{x}_k)\right) \\ &= \prod_{m=1}^{M_k}\left(1+\frac{\lambda_{T,k}}{\rho_k}\int p(\boldsymbol{z}_{m,k}|\boldsymbol{y}_{m,k})p(\boldsymbol{y}_{m,k}|\boldsymbol{x}_k)d\boldsymbol{y}_{m,k}\right), \end{aligned} \tag{3.30}$$

where $\rho = \frac{\lambda_{C,k}}{A_C}$ represents the clutter density and A_C denotes the area of the region where clutter may be emitted from.

The sensor characteristics are approximated with a uniform pdf,

$$p(\boldsymbol{z}_{m,k}|\boldsymbol{y}_{m,k}) = U_{[\boldsymbol{z}_{m,k}]}\left(\tilde{h}\left(\boldsymbol{y}_{m,k}\right)\right). \tag{3.31}$$

Substituting this equation and (3.13) into (3.30), results in

$$p(\boldsymbol{Z}_k, \boldsymbol{\lambda}_k|\boldsymbol{x}_k) = \prod_{m=1}^{M_k}\left(1 + \frac{\lambda_{T,k}}{\rho_k}\int U_{[\boldsymbol{z}_{m,k}]}\left(\tilde{h}\left(\boldsymbol{y}_{m,k}\right)\right)U_{q(\boldsymbol{x}_k)}(\boldsymbol{y}_{m,k})d\boldsymbol{y}_{m,k}\right) \tag{3.32}$$

The updated marginal posterior distribution for crowd tracking can then be expressed with the equation:

$$\begin{aligned} p(\boldsymbol{x}_k|\boldsymbol{Z}_{1:k}, \boldsymbol{\lambda}_k) &= \frac{1}{\alpha_k}p(\boldsymbol{Z}_k, \boldsymbol{\lambda}_k|\boldsymbol{x}_k)p(\boldsymbol{x}_k|\boldsymbol{Z}_{1:k-1}, \boldsymbol{\lambda}_{k-1}) \\ &= \frac{1}{\alpha_k}\sum_{p=1}^{N} w_{k-1}^{(p)}\prod_{m=1}^{M_k}\Bigg(U_{[\boldsymbol{x}_{k|k-1}^{(p)}]}(\boldsymbol{x}_k) + \\ \frac{\lambda_{T,k}}{\rho_k}&\int U_{[\boldsymbol{x}_{k|k-1}^{(p)}]}(\boldsymbol{x}_k)U_{[\boldsymbol{z}_{m,k}]}\left(\tilde{h}\left(\boldsymbol{y}_{m,k}\right)\right)U_{q(\boldsymbol{x}_k)}(\boldsymbol{y}_{m,k})d\boldsymbol{y}_{m,k}\Bigg). \end{aligned} \tag{3.33}$$

Each of the M_k product terms, $U_{[\boldsymbol{x}_{k|k-1}^{(p)}]}(\boldsymbol{x}_k)U_{[\boldsymbol{z}_{m,k}]}\left(\tilde{h}\left(\boldsymbol{y}_{m,k}\right)\right)U_{q(\boldsymbol{x}_k)}(\boldsymbol{y}_{m,k})$, is also a constant function with a support being the following region $S_{p,m} \subset \mathbb{R}^{n_x}$, where

$$S_{p,m} = \left\{\boldsymbol{x}_k \in [\boldsymbol{x}_{k|k-1}^{(p)}] \mid \boldsymbol{y}_{m,k} \in q(\boldsymbol{x}_k), \tilde{h}\left(\boldsymbol{y}_{m,k}\right) \in [\boldsymbol{z}_{m,k}]\right\}. \tag{3.34}$$

Equation (3.34) represents a constraint as defined in Section 3.3.1, thus leading to a CSP. A contractor can be utilised to reduce the predicted supports $[\boldsymbol{x}_{k|k-1}^{(p)}]$ with the constraints imposed by the interval measurements, $[\boldsymbol{Z}_k]$, sensor characteristics, and object model. The contraction leads to M_k new boxes denoted $[\boldsymbol{x}_k^{(p,m)}]$.

Following the definition of the sets $S_{p,m}$ in (3.34),

$$\begin{aligned}
&U_{[\boldsymbol{x}_{k|k-1}^{(p)}]}(\boldsymbol{x}_k)U_{[\boldsymbol{z}_{m,k}]}\left(\tilde{h}\left(\boldsymbol{y}_{m,k}\right)\right)U_{q(\boldsymbol{x}_k)}(\boldsymbol{y}_{m,k})\\
&=U_{[\boldsymbol{z}_{m,k}]}\left(\tilde{h}\left(\boldsymbol{y}_{m,k}\right)\right)U_{q(\boldsymbol{x}_k)}(\boldsymbol{y}_{m,k})\frac{1}{||[\boldsymbol{x}_{k|k-1}^{(p)}]||}||S_{p,m}||U_{S_{p,m}}(\boldsymbol{x}_k),\\
&\simeq U_{[\boldsymbol{z}_{m,k}]}\left(\tilde{h}\left(\boldsymbol{y}_{m,k}\right)\right)U_{q(\boldsymbol{x}_k)}(\boldsymbol{y}_{m,k})\frac{||[\boldsymbol{x}_k^{(p,m)}]||}{||[\boldsymbol{x}_{k|k-1}^{(p)}]||}U_{[\boldsymbol{x}_k^{(p,m)}]}(\boldsymbol{x}_k), \qquad (3.35)
\end{aligned}$$

since by definition $[\boldsymbol{x}_k^{(p,m)}]$ is the smallest box containing $S_{p,m}$. Substituting (3.35) in (3.33) leads to the following updated expression for the posterior state pdf:

$$\begin{aligned}
p(\boldsymbol{x}_k|\boldsymbol{Z}_{1:k},\boldsymbol{\lambda}_k)=\frac{1}{\alpha_k}\sum_{p=1}^{N}w_{k-1}^{(p)}\prod_{m=1}^{M_k}\Bigg(&U_{[\boldsymbol{x}_{k|k-1}^{(p)}]}(\boldsymbol{x}_k)+\\
\frac{\lambda_{T,k}}{\rho_k}\frac{||[\boldsymbol{x}_k^{(p,m)}]||}{||[\boldsymbol{x}_{k|k-1}^{(p)}]||}U_{[\boldsymbol{x}_k^{(p,m)}]}(\boldsymbol{x}_k)\int U_{[\boldsymbol{z}_{m,k}]}&\left(\tilde{h}\left(\boldsymbol{y}_{m,k}\right)\right)U_{q(\boldsymbol{x}_k)}(\boldsymbol{y}_{m,k})d\boldsymbol{y}_{m,k}\Bigg). \qquad (3.36)
\end{aligned}$$

The integration terms are approximated by a uniform distribution,

$$\int U_{[\boldsymbol{z}_{m,k}]}\left(\tilde{h}\left(\boldsymbol{y}_{m,k}\right)\right)U_{q(\boldsymbol{x}_k)}(\boldsymbol{y}_{m,k})d\boldsymbol{y}_{m,k}=U_{r(\boldsymbol{x}_k)}\left(\boldsymbol{z}_{m,k}\right), \qquad (3.37)$$

where $r(\boldsymbol{x}_k)$ represents an interval dependent on the states and measurement function. The validity of this assumption is explored in Appendix A. The posterior state pdf can thus be expanded accordingly:

$$\begin{aligned}
p(\boldsymbol{x}_k|\boldsymbol{Z}_{1:k},\boldsymbol{\lambda}_k)&=\frac{1}{\alpha_k}\sum_{p=1}^{N}w_{k-1}^{(p)}\prod_{m=1}^{M_k}\left(U_{[\boldsymbol{x}_{k|k-1}^{(p)}]}(\boldsymbol{x}_k)+\frac{\lambda_{T,k}}{\rho_k}\frac{1}{|r(\boldsymbol{x}_k)|}\frac{||[\boldsymbol{x}_k^{(p,m)}]||}{||[\boldsymbol{x}_{k|k-1}^{(p)}]||}U_{[\boldsymbol{x}_k^{(p,m)}]}(\boldsymbol{x}_k)\right)\\
&=\frac{1}{\alpha_k}\sum_{p=1}^{N}w_{k-1}^{(p)}\left(\left(U_{[\boldsymbol{x}_{k|k-1}^{(p)}]}(\boldsymbol{x}_k)\right)^{M_k}+\sum_{m=1}^{M_k}\sum_{j=1}^{\binom{M_k}{m}}\left(U_{[\boldsymbol{x}_{k|k-1}^{(p)}]}(\boldsymbol{x}_k)\right)^{M_k-m}\right.\\
&\left.\prod_{i\in\mathcal{A}_j^m}\frac{\lambda_{T,k}}{\rho_k}\frac{1}{|r(\boldsymbol{x}_k)|}\frac{||[\boldsymbol{x}_k^{(p,i)}]||}{||[\boldsymbol{x}_{k|k-1}^{(p)}]||}U_{[\boldsymbol{x}_k^{(p,i)}]}(\boldsymbol{x}_k)\right). \qquad (3.38)
\end{aligned}$$

where $\mathcal{A}^m=\left\{\mathcal{A}_j^m, j\in\mathcal{J}\right\}$, with $\mathcal{J}=\left\{1,2,...,\binom{M_k}{m}\right\}$ and $\mathcal{A}_j^m\subseteq\mathcal{S}:|\mathcal{A}_j^m|=m$, where $\mathcal{S}=\{1,2,...,M_k\}$. For example, if $M_k=3$ and $m=2$ then $\mathcal{A}^m=\{\{1,2\},\{1,3\},\{2,3\}\}$.

The posterior state pdf is a weighted sum of uniform pdfs. The number of weighted uniform pdf's increases exponentially with the number of measurements, which can render the algorithm too computationally expensive for a large number of measurements. Typically, there is a large disparity between the weights of the summed uniform pdfs. This allows for the approximation of the posterior pdf by a single uniform pdf for each box particle. The dominating term in the uniform pdf weights is $\frac{\lambda_{T,k}}{\rho_k |r(\boldsymbol{x}_k)| |[\boldsymbol{x}^{(p)}_{k|k-1}]|}$. This term is maximised when all the measurements are assumed to originate from the crowd. In this case, the posterior state pdf is approximated by:

$$p(\boldsymbol{x}_k|\boldsymbol{Z}_{1:k}, \boldsymbol{\lambda}_k) \approx \frac{1}{\alpha_k} \sum_{p=1}^{N} w^{(p)}_{k-1} \left(\prod_{i \in \mathcal{S}} \frac{\lambda_{T,k}}{\rho_k} \frac{1}{|r(\boldsymbol{x}_k)|} \frac{|[\boldsymbol{x}^{(p,i)}_k]|}{|[\boldsymbol{x}^{(p)}_{k|k-1}]|} U_{[\boldsymbol{x}^{(p,i)}_k]}(\boldsymbol{x}_k) \right). \tag{3.39}$$

The multiplication of uniform pdfs can be further simplified to obtain a single uniform pdf with a corresponding weight. This includes the intersection of the intervals of all the uniform pdfs:

$$\begin{aligned} p(\boldsymbol{x}_k|\boldsymbol{Z}_{1:k}, \boldsymbol{\lambda}_k) &\propto \sum_{p=1}^{N} w^{(p)}_{k-1} \left(\prod_{i \in \mathcal{S}} \frac{\lambda_{T,k}}{\rho_k} \frac{1}{|r(\boldsymbol{x}_k)|} \frac{|[\boldsymbol{x}^{(p,i)}_k]|}{|[\boldsymbol{x}^{(p)}_{k|k-1}]|} \right) \frac{|\cap_{i \in \mathcal{S}} [\boldsymbol{x}^{(p,i)}_k]|}{\prod_{i \in \mathcal{S}} |[\boldsymbol{x}^{(p,i)}_k]|} U_{\cap_{i \in \mathcal{S}} [\boldsymbol{x}^{(p,i)}_k]}(\boldsymbol{x}_k) \\ &\propto \sum_{p=1}^{N} w^{(p)}_{k-1} \left(\prod_{i \in \mathcal{S}} \frac{\lambda_{T,k}}{\rho_k |r(\boldsymbol{x}_k)| |[\boldsymbol{x}^{(p)}_{k|k-1}]|} \right) |\cap_{i \in \mathcal{S}} [\boldsymbol{x}^{(p,i)}_k]| U_{\cap_{i \in \mathcal{S}} [\boldsymbol{x}^{(p,i)}_k]}(\boldsymbol{x}_k). \end{aligned} \tag{3.40}$$

However, this intersection result typically does not exist or leads to a poor contraction due to the implicit assumption that the measurements originate from the crowd. A more robust approximation for the posterior state pdf, which does not require explicit knowledge of the origin of a measurement, is given by:

$$\begin{aligned} p(\boldsymbol{x}_k|\boldsymbol{Z}_{1:k}, \boldsymbol{\lambda}_k) \approx \sum_{p=1}^{N} w^{(p)}_{k-1} \left(U_{[\boldsymbol{x}^{(p)}_{k|k-1}]}(\boldsymbol{x}_k) \right)^{M_k - (|\mathcal{S}^{(p)}_E| - q)} \times \\ \left(\prod_{i \in \mathcal{S}^{(p)}_E} \frac{\lambda_{T,k}}{\rho_k |r(\boldsymbol{x}_k)| |[\boldsymbol{x}^{(p)}_{k|k-1}]|} \right) | \overset{\{q\}}{\cap}_{i \in \mathcal{S}^{(p)}_E} [\boldsymbol{x}^{(p,i)}_k] | U_{\overset{\{q\}}{\cap}_{i \in \mathcal{S}^{(p)}_E} [\boldsymbol{x}^{(p,i)}_k]}(\boldsymbol{x}_k), \end{aligned} \tag{3.41}$$

where $\mathcal{S}_E^{(p)}$ is the set of indices for the contracted boxes, $[x_{k,m}^{(p)}]$, that exist[2], and q is the maximum number of clutter measurements indexed by $\mathcal{S}_E^{(p)}$. The symbol $\overset{\{q\}}{\cap}$ is the q-relaxed intersection first introduced in [66] to aid in the processing of clutter measurements in a purely interval framework.

The difference between the posterior pdf represented by equations (3.38) and (3.41) is highlighted graphically through an example in figure 3.1.

In summary, $p(\boldsymbol{x}_k|\boldsymbol{Z}_{1:k}, \boldsymbol{\lambda}_k)$ is approximated by $\{(\tilde{w}_k^{(p)}, [\boldsymbol{x}_k^{(p)}])\}_{p=1}^N$, where

$$[\boldsymbol{x}_k^{(p)}] = \overset{\{q\}}{\cap}_{i \in \mathcal{S}_E^{(p)}} [\boldsymbol{x}_k^{(p,i)}]. \tag{3.42}$$

and

$$\tilde{w}_k^{(p)} \propto w_{k-1}^{(p)} \left(U_{[\boldsymbol{x}_{k|k-1}^{(p)}]}(\boldsymbol{x}_k) \right)^{M_k - (|\mathcal{S}_E^{(p)}| - q)} \left(\prod_{i \in \mathcal{S}_E^{(p)}} \frac{\lambda_T}{\rho |r(\boldsymbol{x}_k)| |[\boldsymbol{x}_{k|k-1}^{(p)}]|} \right) |[\boldsymbol{x}_k^{(p)}]|, \tag{3.43}$$

The Box PF method for crowd tracking is summarised in Algorithm 6.

3.4.1 Box Particle Filter Implementation Considerations

Interval Contraction: In general, an important step in interval based techniques used for state estimation is interval contraction [66]. In the Box PF it is required to obtain the contracted box particles by solving the CSP described by equation (3.34). Without loss of generality, the Constraints Propagation (CP) technique [67] is the contractor utilised throughout this thesis. The main advantages of the CP method is its efficiency, especially in the presence of measurements and constraints which are highly redundant. The CP algorithm, which in this application is the calculation of the intersection of the box states for each particle with all the interval measurements, is illustrated in Algorithm 7.

Box PF Resampling: Generally, in particle filtering, there are a variety of different

[2] Measurements which result in a contraction of the state that does not exist are located a significant distance from the state and are considered to be clutter measurements.

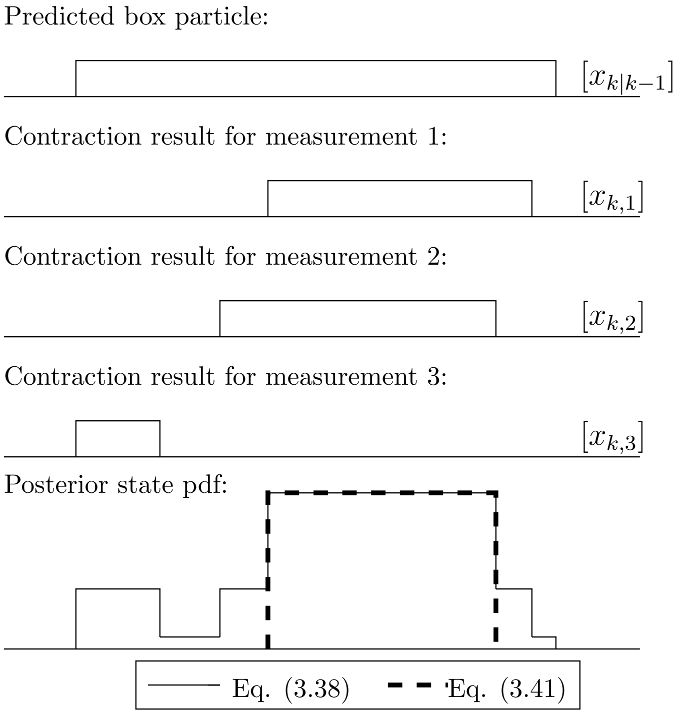

Predicted box particle:
$[x_{k|k-1}]$
Contraction result for measurement 1:
$[x_{k,1}]$
Contraction result for measurement 2:
$[x_{k,2}]$
Contraction result for measurement 3:
$[x_{k,3}]$
Posterior state pdf:
Eq. (3.38)
Eq. (3.41)

Figure 3.1: Illustration of the difference between the posterior state pdf represented by equations (3.38) and (3.41). This example consists of 3 measurements (measurement 3 represents a clutter measurement), a single state dimension, and a single box particle.

resampling schemes available [74]. Based on the weights, a particle is replicated a specific number of times. The Box PF differs by dividing a selected box particle into smaller box-particles as many times as it was to be replicated. Several subdivision strategies exist. In this thesis the box particles are subdivided based on the dimension with the largest box face.

Relaxed Intersection Considerations: The parameter q is introduced in equation (3.41). This specifies the maximum number of clutter measurements that result in a contraction of the states that exists. These are the clutter measurements which are located in the vicinity of the boundary of the crowd. The area in the measurement space where a measurement can result in a contraction of the state that exists is dependent on the size of the box particle. An estimate for q is given by:

$$q = \frac{\rho_k A_{CT}}{4}. \tag{3.45}$$

Algorithm 6 The Box Particle Filter for Crowd Tracking

```
***Initialisation***
Initialise the set of box particles, $\{[\boldsymbol{x}_0^{(p)}]\}_{p=1}^{N}$, sampling from the prior distribution.

for $k$ = 1,...,$T$ do
    ***Prediction***
    Generate the predicted box particles, $\{[\boldsymbol{x}_{k|k-1}^{(p)}]\}_{p=1}^{N}$, by propagating the box particles through the state evolution model and applying interval inclusion functions as described in [67, 65].
    ***Measurement Update***
    Upon the receipt of new measurements:
    Convert the measurements into a set of measurement boxes, $[\boldsymbol{Z}_k]$.
    Solve the CSP, as described in Section 3.4.1, to obtain the contracted box particles $[\boldsymbol{x}_k^{(p,m)}]$.
    Determine $[\boldsymbol{x}_k^{(p)}]$ according to (3.42).
    Update the weights $\{w_k^{(p)}\}_{p=1}^{N}$ according to (3.43).
    ***Output***
    Obtain an estimate for the state of the crowd through:
        $[\widehat{\boldsymbol{x}}_k] = \sum_{p=1}^{N} w_k^{(p)}[\boldsymbol{x}_k^{(p)}].$    (3.44)
    Further, a point estimate for the state can be obtained as the midpoint of the box estimate of the state.
    ***Resampling***
    Compute the effective sample size in (2.12)
    If $ESS \leq N_{thresh}$ (with e.g. $N_{thresh} = 2N/3$), then resample the box particles by division, and reset the weights: $\{w_k^{(p)}\} = 1/N$.
end for
```

The estimated clutter measurement rate is used:

$$\rho_k = \frac{\lambda_{C,k}}{A_{CR}}, \tag{3.46}$$

where the area of the clutter region is given by $A_{CR} = A_S - A_T$, A_S is the total area observed by the sensor, and A_T is the area of the crowd, approximated from the estimate of the crowd at the previous time instant, $k-1$. For the given crowd tracking problem, the area A_{CT} is given by:

$$\begin{aligned} A_{CT} = &\left(\left(\left(\overline{x_k^{(p)}} + \frac{\overline{a_k^{(p)}}}{2}\right) - \left(\underline{x_k^{(p)}} - \frac{\overline{a_k^{(p)}}}{2}\right)\right)\left(\left(\overline{y_k^{(p)}} + \frac{\overline{b_k^{(p)}}}{2}\right) - \left(\underline{y_k^{(p)}} - \frac{\overline{b_k^{(p)}}}{2}\right)\right)\right) \\ &- \left(\left(\left(\underline{x_k^{(p)}} + \frac{\underline{a_k^{(p)}}}{2}\right) - \left(\overline{x_k^{(p)}} - \frac{\underline{a_k^{(p)}}}{2}\right)\right)\left(\left(\underline{y_k^{(p)}} + \frac{\underline{b_k^{(p)}}}{2}\right) - \left(\overline{y_k^{(p)}} - \frac{\underline{b_k^{(p)}}}{2}\right)\right)\right). \end{aligned} \tag{3.47}$$

The factor of 4 in equation (3.45) was introduced to take into account that the area A_{CT} also includes the region inside of the crowd, where no clutter measurements are found. It is important to note that the algorithm is fairly robust to the value of q as this represents a maximum number of clutter points, and not the actual number of clutter points.

3.5 Review of the Convolution Particle Filter

The CPF approach relies on convolution kernel density estimation and regularisation of the distributions, respectively, of the states and measurements [99, 29, 119]. The CPF belongs to a class of PFs with valuable advantages: simultaneous estimation of state variables and unknown parameters and continuous approximation of the corresponding pdf. Being likelihood free filters makes them attractive for solving complex problems where the likelihood is not available for evaluation in an analytical form.

Considering the state space model of (2.1) and (2.2), an alternative form for the posterior state pdf is given by

$$p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k}) = \frac{p(\boldsymbol{x}_k, \boldsymbol{z}_{1:k})}{\int p(\boldsymbol{x}_k, \boldsymbol{z}_{1:k}) d\boldsymbol{x}_k}. \tag{3.49}$$

Assume that it is possible to sample from the prior state pdf, i.e. $\boldsymbol{x}_0^{(i)} \sim p(\boldsymbol{x}_0)$ where $i = 1, .., N$. Sampling recursively a k number of times from (2.3) results in the sample sets of $\{\boldsymbol{x}_k^{(i)}\}_{i=1}^{N}$ and $\{\boldsymbol{z}_{1:k}^{(i)}\}_{i=1}^{N}$.The samples represent an empirical estimate of the joint

Algorithm 7 CP algorithm for Rectangularly Shaped Crowds

```
Input: $[\boldsymbol{x}_k^{(p)}]$, $[\boldsymbol{x}_{k-1}^{(p)}]$, $[\boldsymbol{z}_{m,k}]$.
Let $[\boldsymbol{x}_k^{(p,m)}] = [\boldsymbol{x}_k^{(p)}]$, and DONE = FALSE.
while DONE == FALSE do
    Contract the intervals with each constraint:
    $$\begin{aligned}
    [x_k^{(p,m)}] &= [x_k^{(p,m)}] \cap \left( [z_{1,m,k}] \mp \frac{[a_k^{(p,m)}]}{2} \cdot [0,1] \right), \\
    [\dot{x}_k^{(p,m)}] &= [\dot{x}_k^{(p,m)}] \cap \left( \frac{[x_k^{(p,m)}] - [x_{k-1}^{(p)}]}{\frac{1}{\alpha_x}(1 - e^{-\alpha_x T_s})} \right), \\
    [y_k^{(p,m)}] &= [y_k^{(p,m)}] \cap \left( [z_{2,m,k}] \mp \frac{[b_k^{(p,m)}]}{2} \cdot [0,1] \right), \\
    [\dot{y}_{k,m}^{(p,m)}] &= [\dot{y}_k^{(p,m)}] \cap \left( \frac{[y_k^{(p,m)}] - [y_{k-1}^{(p)}]}{\frac{1}{\alpha_y}(1 - e^{-\alpha_y T_s})} \right), \\
    [a_k^{(p,m)}] &= [a_{k,m}^{(p,m)}] \cap \pm 2 \left( \frac{[z_{1,m,k}] - [x_k^{(p,m)}]}{[0,1]} \right), \\
    [b_k^{(p,m)}] &= [b_{k,m}^{(p,m)}] \cap \pm 2 \left( \frac{[z_{2,m,k}] - [y_k^{(p,m)}]}{[0,1]} \right), \\
    [z_{1,m,k}] &= [z_{1,m,k}] \cap \left( [x_k^{(p,m)}] \pm \frac{[a_k^{(p,m)}]}{2} \cdot [0,1] \right), \\
    [z_{2,m,k}] &= [z_{2,m,k}] \cap \left( [y_k^{(p,m)}] \pm \frac{[b_k^{(p,m)}]}{2} \cdot [0,1] \right).
    \end{aligned} \tag{3.48}$$
    If convergence criterion is met (e.g. difference in the size of pre-contracted and
    contracted boxes is below a threshold), DONE = TRUE.
end while
Output: $[\boldsymbol{x}_k^{(p,m)}]$.
```

pdf,

$$\begin{aligned} \widehat{p}(\boldsymbol{x}_k, \boldsymbol{z}_{1:k}) &\approx p(\boldsymbol{x}_k, \boldsymbol{z}_{1:k}) \\ &= \frac{1}{N} \sum_{i=1}^{N} \delta(\boldsymbol{x}_k - \boldsymbol{x}_k^{(i)}, \boldsymbol{z}_{1:k} - \boldsymbol{z}_{1:k}^{(i)}). \end{aligned} \tag{3.50}$$

A kernel estimate for the joint state pdf, $p_k^K(\boldsymbol{x}_k, \boldsymbol{z}_{1:k})$, is obtained through the convolution of the empirical estimate in (3.50) with a suitable kernel

$$\begin{aligned} p_k^K(\boldsymbol{x}_k, \boldsymbol{z}_{1:k}) &= \widehat{p}(\boldsymbol{x}_k, \boldsymbol{z}_{1:k}) * K_h \\ &= \frac{1}{N}\sum_{i=1}^{N} K_h^x(\boldsymbol{x}_k - \boldsymbol{x}_k^{(i)}) K_h^{\bar{z}}(\boldsymbol{z}_{1:k} - \boldsymbol{z}_{1:k}^{(i)}), \end{aligned} \tag{3.51}$$

where $*$ represents the convolution operator, K_h, K_h^x, and $K_h^{\bar{z}}$ are Parzen-Ronsenblatt kernels of appropriate dimensions, and $K_h^{\bar{z}}(\boldsymbol{z}_{1:k} - \boldsymbol{z}_{1:k}^{(i)}) = \prod_{j=1}^{k} K_h^z(\boldsymbol{z}_j - \boldsymbol{z}_j^{(i)})$. Finally this results in the following kernel approximation of the posterior state pdf in (3.49)

$$p_k^K(\boldsymbol{x}_k|\boldsymbol{z}_{1:k}) = \frac{\sum_{i=1}^{N} K_h^x(\boldsymbol{x}_k - \boldsymbol{x}_k^{(i)}) K_h^{\bar{z}}(\boldsymbol{z}_{1:k} - \boldsymbol{z}_{1:k}^{(i)})}{\sum_{i=1}^{N} K_h^{\bar{z}}(\boldsymbol{z}_{1:k} - \boldsymbol{z}_{1:k}^{(i)})} \tag{3.52}$$

Based on this principal, a recursive algorithm for the CPF, including a resampling step, is summarised in Algorithm 8.

3.6 The Convolution Particle Filter for Crowd Tracking

In this Section an adaptive CPF algorithm for crowds tracking is developed. The key novelty of the proposed adaptive CPF algorithm and the advantage of using it for crowd tracking is: *i*) its ability to efficiently deal with multiple measurements, including a high level of clutter, *ii*) ability to resolve data association problems, without the need to estimate clutter parameters, *iii*) estimation of dynamically changing parameters of crowds jointly with the dynamic kinematic states.

For the purposes of crowds tracking the marginal posterior state pdf has to be calculated and can be expressed to be independent of the clutter and measurement rates, reducing the expression from equation (3.18) to:

$$p(\boldsymbol{\zeta}_k|\boldsymbol{Z}_{1:k}) = p(\boldsymbol{x}_k|\boldsymbol{Z}_{1:k})p(\lambda_{T,k}|\boldsymbol{Z}_{1:k})p(\lambda_{C,k}|\boldsymbol{Z}_{1:k}). \tag{3.54}$$

The focus is then on obtaining a kernel estimate of $p(\boldsymbol{x}_k|\boldsymbol{Z}_{1:k})$ in order to extract

Algorithm 8 The Standard Convolution Particle Filter

```
***Initialisation***
Initialise the set of particles, $\{\boldsymbol{x}_0^{(p)}\}_{p=1}^N$, sampling from the prior distribution.
**for** $k = 1,...,T$ **do**
    **for** $i = 1,...,N$ **do**
        ***Prediction***
        State sampling: $\boldsymbol{x}_k^{(i)} \sim p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^{(i)})$
        Measurement sampling: $\boldsymbol{z}_k^{(i)} \sim p(\boldsymbol{z}_k|\boldsymbol{x}_k^{(i)})$
        ***Measurement Update***
        weight update: $w_k^{(i)} = w_{k-1}^{(i)} K_h^z(\boldsymbol{z}_k - \boldsymbol{z}_k^{(i)})$
    **end for**
    ***Output***
    Obtain an estimate for the state through:
        $$\widehat{\boldsymbol{x}}_k = \sum_{i=1}^{N} \bar{w}_k^{(i)} \boldsymbol{x}_k^{(i)}. \qquad (3.53)$$
    where $\bar{w}_k^{(p)}$ is the normalised weight of particle $i$.
    ***Resampling***
    Compute the effective sample size in (2.12)
    If $ESS \leq N_{thresh}$ (with e.g. $N_{thresh} = 2N/3$), then resample particles, and reset
    the weights: $\{w_k^{(i)}\}_{i=1}^N = 1/N$.
**end for**
```

estimates for the kinematics and extent of the crowd.

The formulation of the CPF for crowds tracking follows the same sampling and kernel principles as in (3.49) to (3.52). However, when dealing with point objects, a point in the state space corresponds to a single point in the measurement space, according to the model in (2.2). In contrast, in the application of crowd tracking, a point in the state space translates into a region in the measurement space, through (3.13). The role of the measurement kernel in the CPF can be interpreted as a mechanism of assigning a likelihood to a measurement. In the point object case this likelihood varies in the measurement space according to the sampled measurement point and the parameters of the kernel. However, in the crowd tracking case a likely region in the measurement space is already specified. The densities that describe the sensor characteristics and object model can be used to obtain an approximate region in the measurement space for each predicted particle, and are thus equivalent to the kernel. In this case the bandwidth of the kernel varies according to the state, resulting in a variable bandwidth which adds additional flexibility to the CPF while also removing the need to specify a bandwidth parameter. In this application the kernel is approximated as a variable uniform distribution.

An advantage of the proposed CPF framework is that it implicitly resolves the data association problem. Since there are multiple measurements assumed to be independent, the weights of individual measurements are multiplied to obtain a single weight for the particle. However, clutter measurements may occur outside of the support of the adaptive uniform kernel. This would result in particles having a weight of 0 when evaluated by the kernel. To overcome this, the adaptive uniform kernel based on the crowd is added with a uniform distribution which covers the entire observation area of the sensor. The advantage to such an approach is that it removes the need for the estimation of the clutter and measurement rates when only the kinematic states and extent parameters are of interest. The weights are updated sequentially according to

$$w_k^{(i)} = w_{k-1}^{(i)} \prod_{m=1}^{M_k} K_h^Z \left(\boldsymbol{z}_{m,k} \right). \tag{3.55}$$

For the crowd tracking problem presented, the kernel $K_h^Z(\boldsymbol{z}_{m,k})$ is a compositional kernel comprised of a sum of two uniform pdfs:

$$K_h^Z(\boldsymbol{z}_k) = U_{CS}(\boldsymbol{z}_k) + U_{SS}(\boldsymbol{z}_k), \tag{3.56}$$

where the support SS is the entire region observed by the sensor, and the support CS is related to the location of crowd measurements given the particle state. The proposed region is $r(\boldsymbol{x}_k)$, as described in Appendix A.

A detailed description of the CPF for crowd tracking is given in Algorithm 9.

Algorithm 9 The Convolution Particle Filter for Crowd Tracking

```
Initialisation
Initialise the set of particles, $\{\boldsymbol{x}_0^{(p)}\}_{p=1}^{N}$, sampling from the prior distribution.
for $k$ = 1,...,$T$ do
    for $i$ = 1,...,$N$ do
        Prediction
        State sampling: $\boldsymbol{x}_k^{(i)} \sim p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^{(i)})$
        Determine the kernel parameters: $r(\boldsymbol{x}_k^{(i)})$
        Measurement Update
        Update the particle weight, $w_k^i$, according to (3.55)
    end for
    Output
    Obtain an estimate for the state of the crowd through:

        $\widehat{\boldsymbol{x}}_k = \sum_{p=1}^{N} \bar{w}_k^{(p)} \boldsymbol{x}_k^{(p)}.$    (3.57)

    where $\bar{w}_k^{(p)}$ is the normalised weight of particle $p$.
    Resampling
    Compute the effective sample size in (2.12)
    If $ESS \leq N_{thresh}$ (with e.g. $N_{thresh} = 2N/3$), then resample particles, and reset
    the weights: $\{w_k^{(p)}\}_{p=1}^{N} = 1/N$.
end for
```

3.7 Performance Evaluation

In this Section the performance of the crowd tracking Box PF and CPF are compared with the SIR PF described in Section 2.1.1, utilising the generalised likelihood in (3.30). The performance evaluation is done using simulated measurements data.

3.7.1 Computing Platform

The algorithms are implemented in the interpreted MATLAB language environment. Simulations are performed on a mobile computer with Intel(R) Core(TM) i7-4702HQ CPU @ 2.20GHz (4 cores, 8 threads) with 16GB of DDR3 RAM. The Mersenne Twister pseudo-random noise generator, with a seed based on the current time, is used to generate noise when required.

3.7.2 Test Environment

Two different crowd simulations were used to demonstrate the performance.

Rectangular Group Object Simulator: A crowd with a rectangular extent located in a two dimensional plane. The centre of the crowd undergoes motion according to a correlated velocity model. The lengths of the sides of the crowd vary at each time step according to a random walk. Crowd measurements comprise of a number of points uniformly located within the confines of the crowd at each time step. In addition to the crowd measurements, clutter measurements are also present, uniformly located in a region about the crowd.

Realistic Crowd Simulator: Individuals within the crowd are represented as points moving in a two dimensional space. The dynamics of the group is determined by forces acting on those individuals: forces of attraction towards one or more static 'goal' points; constrained forces of repulsion between the elements of the group; constrained forces of repulsion from a set of linear contextual constraints. The net effect is that a crowd of individuals will move in a reasonably realistic manner between constraints. The simulator outputs a set of points corresponding to the positions of each individual in the crowd at each sampling step. The positions of the individuals represent the measurement sources. Additionally, clutter measurements are also present, uniformly located in a region about the crowd.

3.7.3 Rectangular Group Object Simulator Results

This section presents results based on the Rectangular group object simulator. The parameters are as follows:

- *Simulation*: The mean number of measurement sources: $\lambda_T = 100$, Simulation time duration: $T = 40$ s, Sampling time, $T_s = 0.125$ s, Initial rectangular object kinematic state: $\boldsymbol{X}_0 = (100\text{ m}, 0\text{ m/s}, 100\text{ m}, 0\text{ m/s})^\top$, Initial rectangular object extent parameters: $\boldsymbol{\Theta}_0 = (40\text{ m}, 40\text{ m})^\top$, Crowd centre dynamics parameters: Velocity correlation time constant, $T_{cv} = 15$ s, Velocity standard deviation parameters, $\sigma_{v,x} = \sigma_{v,y} = 10$ m/s, Group extent dynamics parameters $\sigma_a = \sigma_b = 1$ m per time step.

- *Sensor*: Measurement uncertainty: $\sigma_{z_1} = \sigma_{z_2} = 0.1$ m. Clutter parameters: Clutter density, $\rho = 1 \times 10^{-2}$. Clutter area = Circular region with radius of 100 m about the centre of the crowd subtracted by the area of the crowd.

- *Filter Parameters*: The CPF and SIR PF utilise a uniform distribution for each state to initialise the particles. In the case of the Box PF, the same uniform region where the CPF and SIR PF randomly generate particles from is subdivided so that the entire region is encompassed by all the box particles. This region for each state is: $x_0^{(p)} = [x_0 - 50; x_0 + 50]$ m, $\dot{x}_0^{(p)} = [\dot{x}_0 - 10; \dot{x}_0 + 10]$ m/s, $y_0^{(p)} = [y_0 - 50; y_0 + 50]$ m, $\dot{y}_0^{(p)} = [\dot{y} - 10; \dot{y} + 10]$ m/s, $a_0^{(p)} = [a_0 - 30; a_0 + 30]$ m, and $b_0^{(p)} = [b_0 - 30; b_0 + 30]$ m.

The root mean square error (RMSE) of the filter estimates are illustrated in this section. The RMSE values for each time step are calculated over a number of MC simulation runs according to

$$RMSE = \sqrt{\frac{1}{N_{MC}} \sum_{i=1}^{N_{MC}} ||\hat{\boldsymbol{x}}_i - \boldsymbol{x}_i||^2}, \tag{3.58}$$

where $\boldsymbol{x}_i$ is the ground truth, $\hat{\boldsymbol{x}}_i$ is the filter estimate, and N_{MC} represents the number of MC runs.

The first set of results illustrate how the filters perform when estimating the marginal posterior state pdf, $p(\boldsymbol{x}_k|\boldsymbol{Z}_{1:k}, \boldsymbol{\lambda}_k)$, with measurement and clutter rates assumed known. Only 4 box particles are required to track the crowd. For comparison, the CPF and SIR PF were also run with 4 particles, however, this resulted in consistent filter divergence due to particle degeneracy. Instead the number of particles were selected based on achieving a similar computational expense for all algorithms. The number of MC runs is 100. The resultant RMSE values are illustrated in Figure 3.2. The comparison of the computational complexity for these results are presented

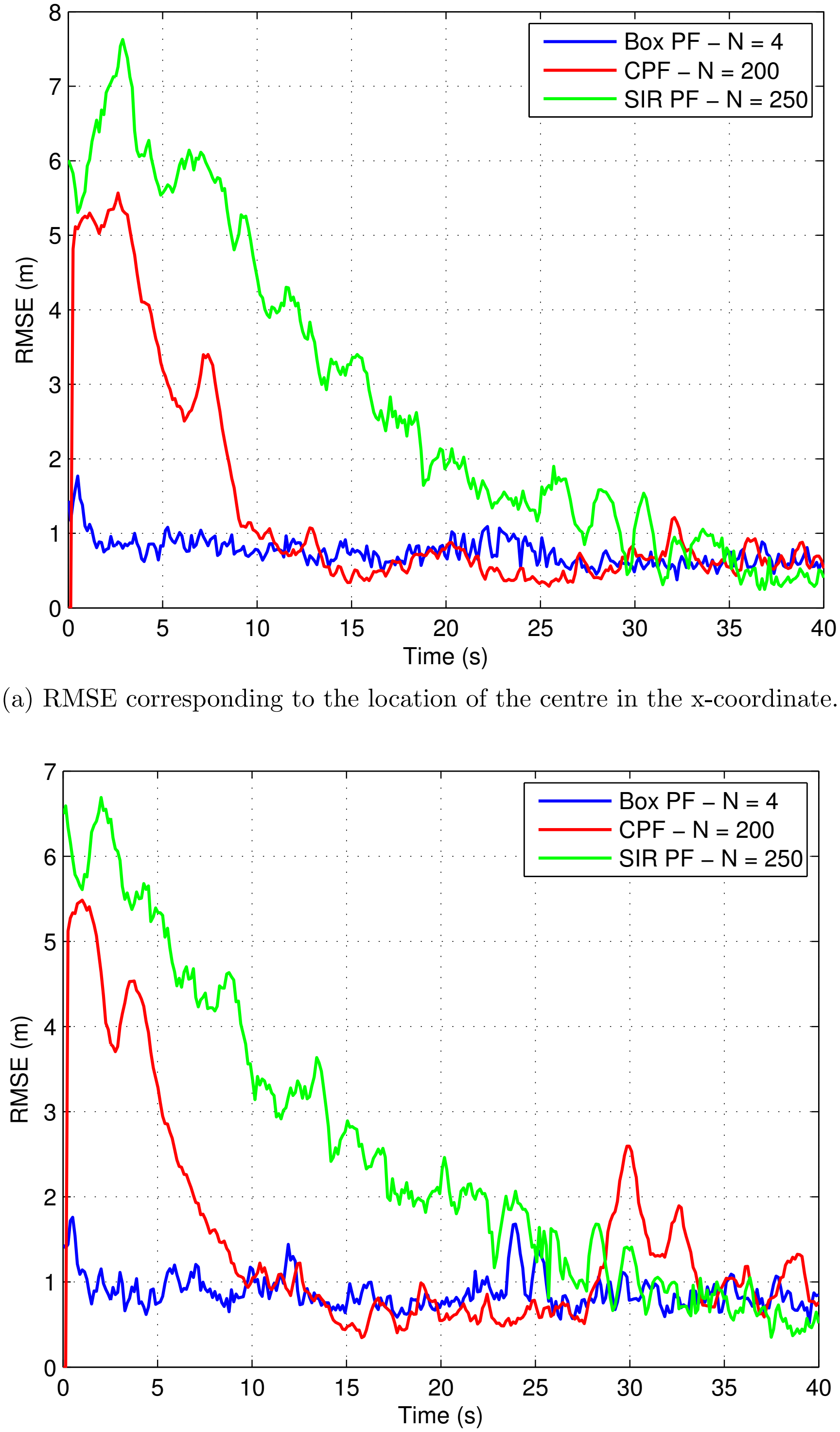
Table 3.1: MATLAB computational time corresponding to the results in Figure 3.2.

Algorithm	Computation Time (s)
Box PF	13.47
CPF	14.43
SIR PF	13.01

in Table 3.1. It is worth noting that the implementation of the Box PF utilises the INTLAB toolbox [101] for performing interval operations. INTLAB was initially designed and optimised for estimating rounding errors. Utilising alternative methods for the interval operations could significantly reduce the computational complexity of the Box PF. The Box PF and CPF are able to lock on to the crowd significantly faster than the SIR PF. It is noted that the RMSE is generally higher for the Box PF once all filters have locked onto the crowd. This can be attributed to the approximations made in the derivation of the marginal posterior state pdf. The SIR PF is also matched in terms of the model noise and likelihood expression.

A computational complexity analysis for Box PFs for point objects, can be found for Bernoulli filters in [53] and with a PHD filter [102]. In these works it is shown that the Box PF for point object tracking requires a significantly smaller number of box particles compared with the particles needed in the Bernoulli and PHD filters, including an even greater computational saving.

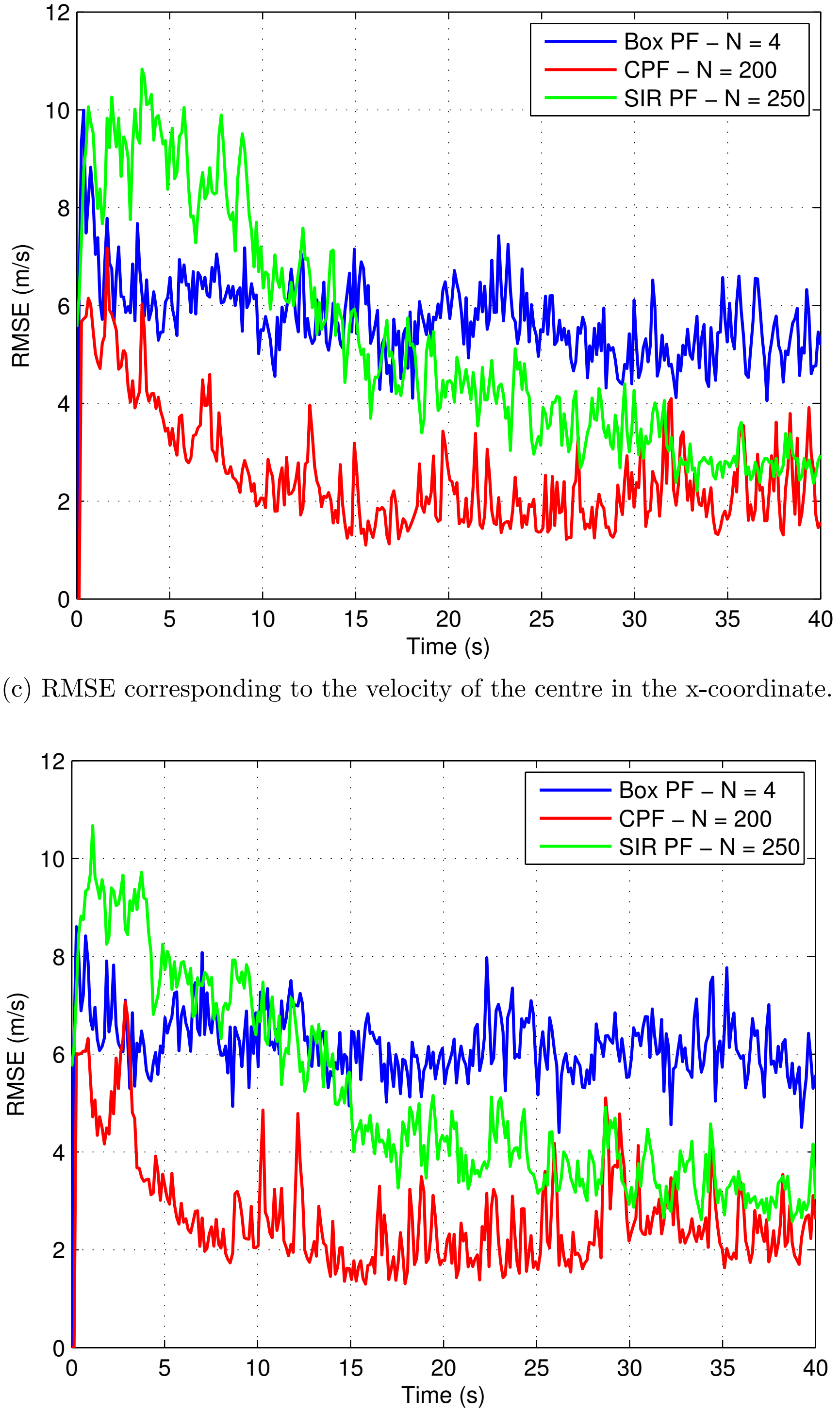
The second set of results re-iterate the experiment with a significant increase in the number of particles for the CPF and SIR PF in order to improve tracking performance with an increase in computational expense. The resultant RMSE values are illustrated in Figure 3.3, and the computational cost comparison for these results are presented in Table 3.2. Increasing the number of particles in the CPF and SIR PF decrease the amount of time required to lock on to the crowd, however, the faster lock comes at a significantly larger computational burden.



(a) RMSE corresponding to the location of the centre in the x-coordinate.

(b) RMSE corresponding to the location of the centre in the y-coordinate.

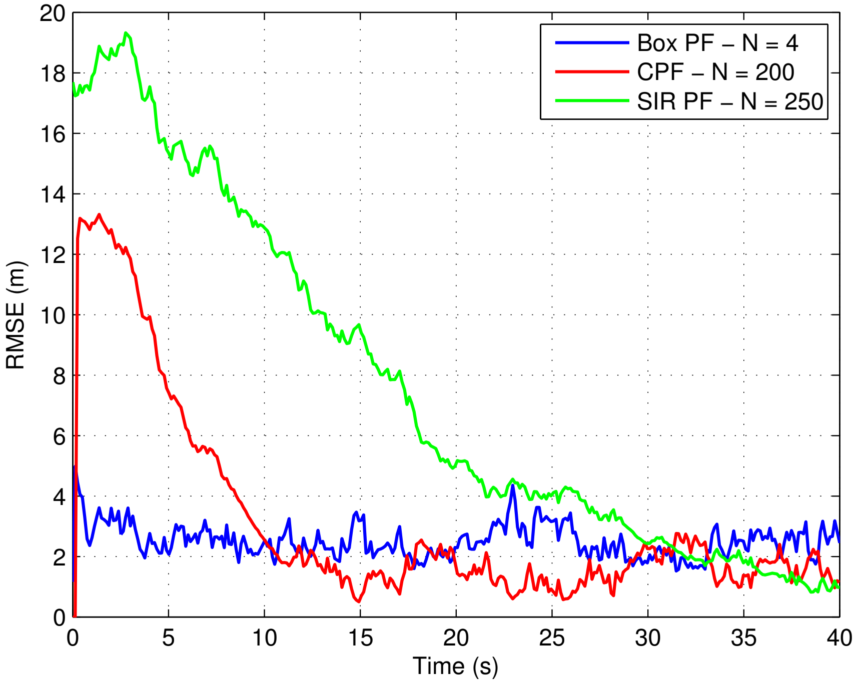
Figure 3.2: Comparison of the RMSE for the states of the Box PF, CPF and SIR PF with equal computational complexity.



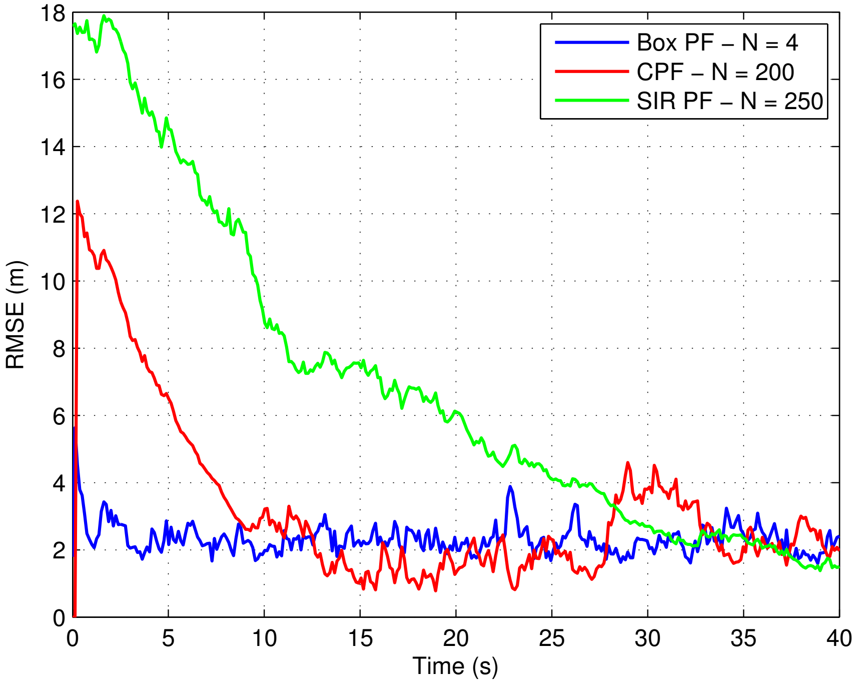
(c) RMSE corresponding to the velocity of the centre in the x-coordinate.

(d) RMSE corresponding to the velocity of the centre in the y-coordinate.

Figure 3.2: Comparison of the RMSE for the states of the Box PF, CPF and SIR PF with equal computational complexity.(cont.)

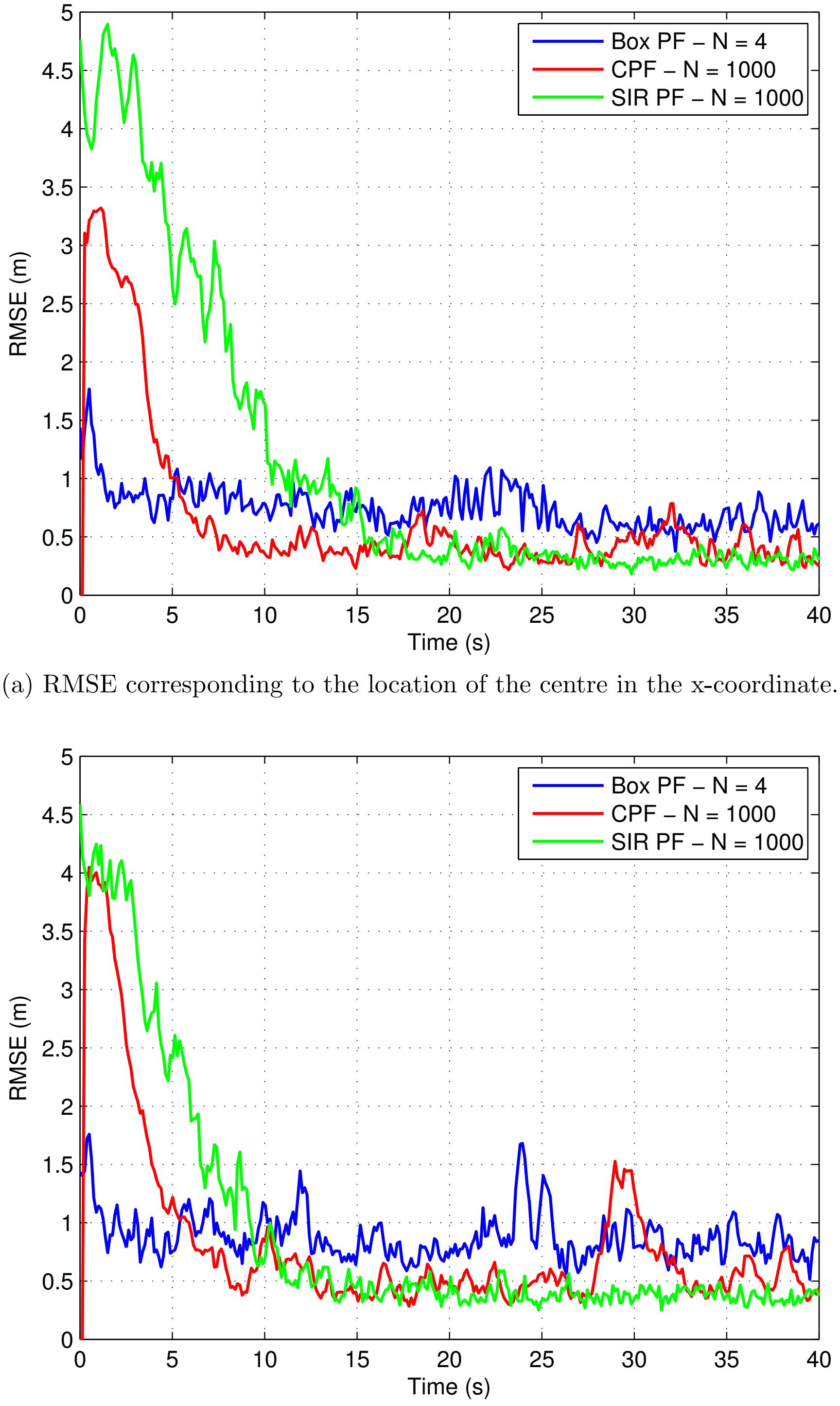


(e) RMSE corresponding to the length of side A.



(f) RMSE corresponding to the length of side B.

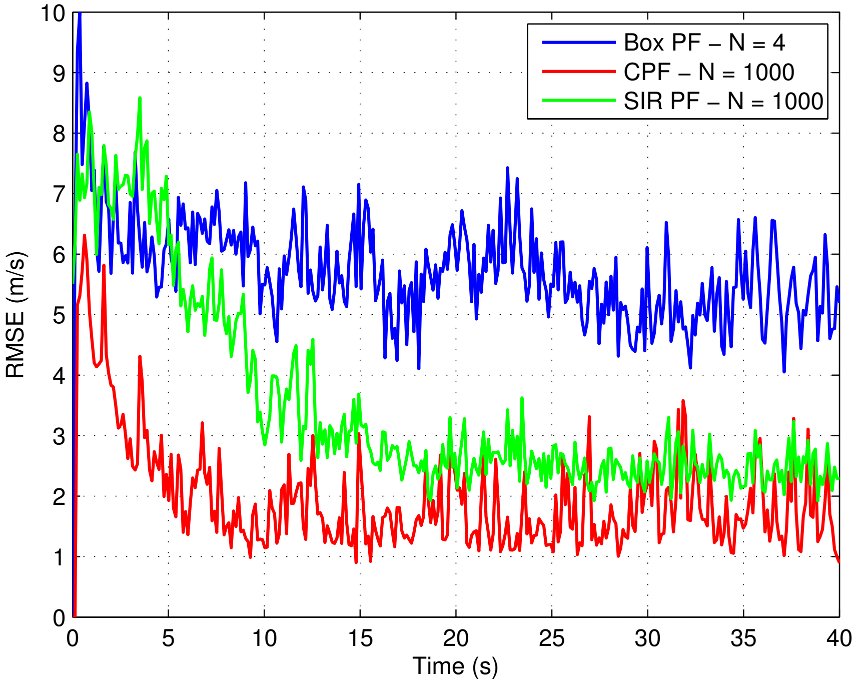
Figure 3.2: Comparison of the RMSE for the states of the Box PF, CPF and SIR PF with equal computational complexity. (cont.)



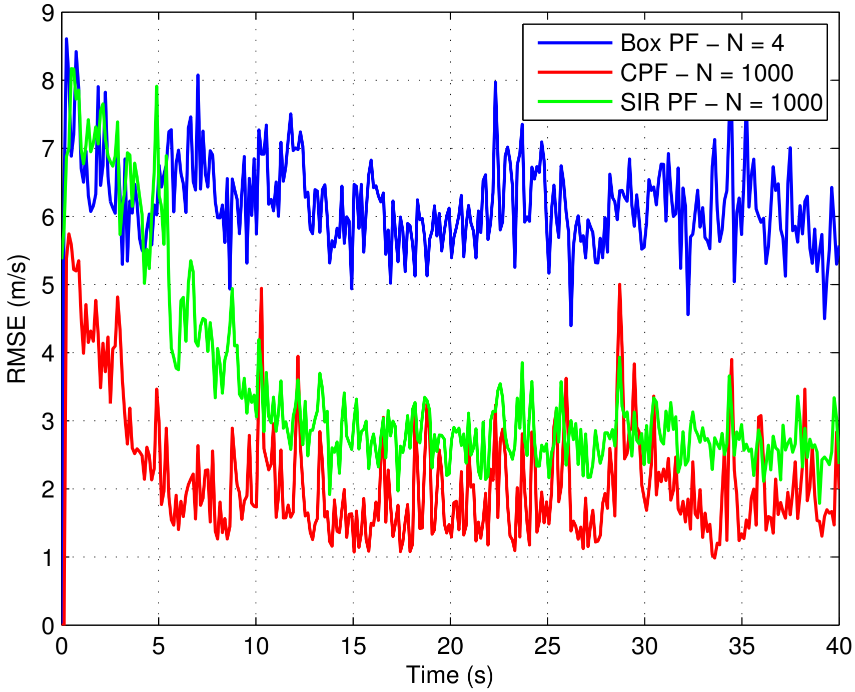
(a) RMSE corresponding to the location of the centre in the x-coordinate.

(b) RMSE corresponding to the location of the centre in the y-coordinate.

Figure 3.3: Comparison of the RMSE for the states of the Box PF, CPF and SIR PF for maximised performance.

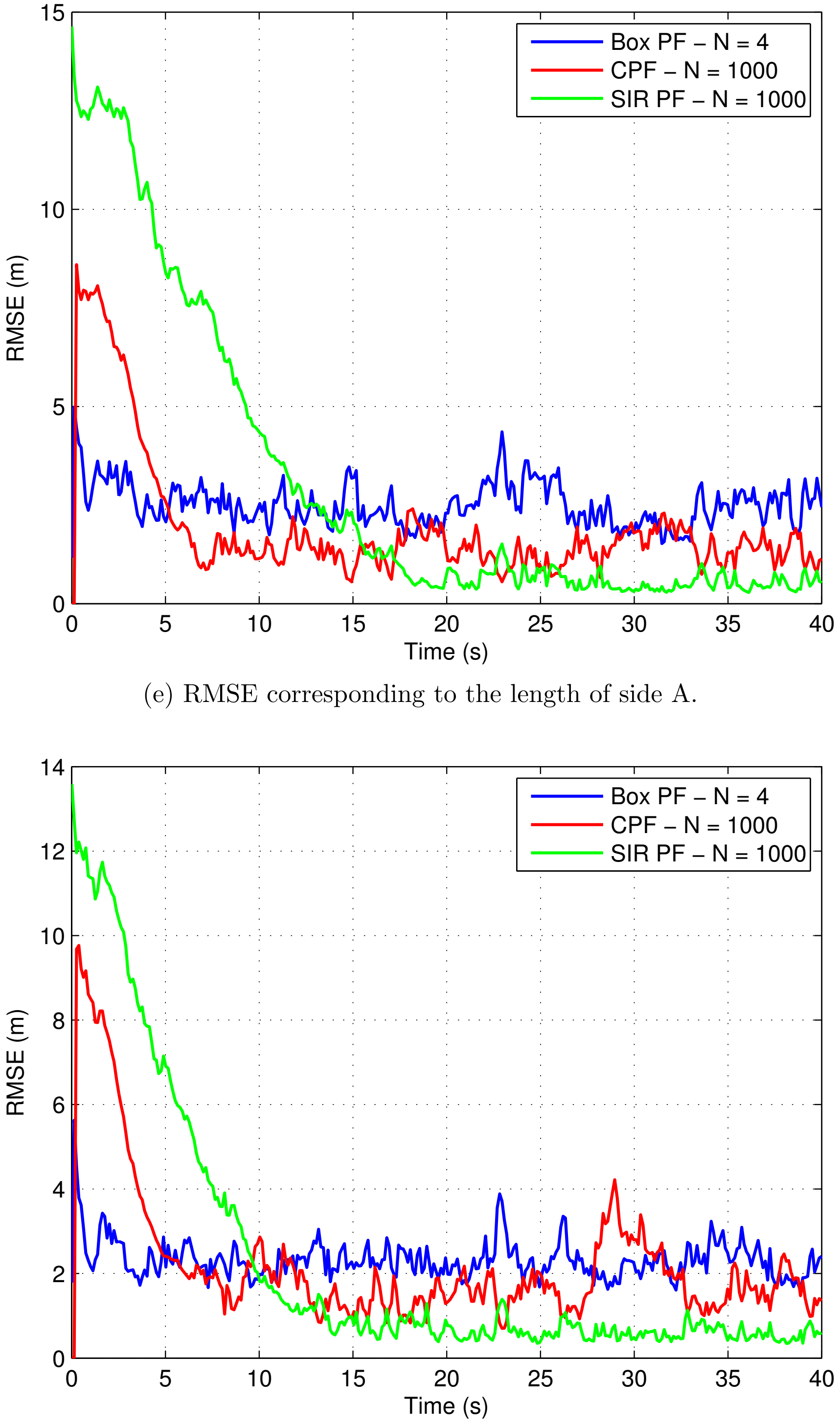


(c) RMSE corresponding to the velocity of the centre in the x-coordinate.



(d) RMSE corresponding to the velocity of the centre in the y-coordinate.

Figure 3.3: Comparison of the RMSE for the states of the Box PF, CPF and SIR PF for maximised performance. (cont.)



(e) RMSE corresponding to the length of side A.

(f) RMSE corresponding to the length of side B.

Figure 3.3: Comparison of the RMSE for the states of the Box PF, CPF and SIR PF for maximised performance. (cont.)

Table 3.2: MATLAB computational time corresponding to the results in Figure 3.3.

Algorithm	Computation Time (s)
Box PF	13.47
CPF	42.16
SIR PF	45.58

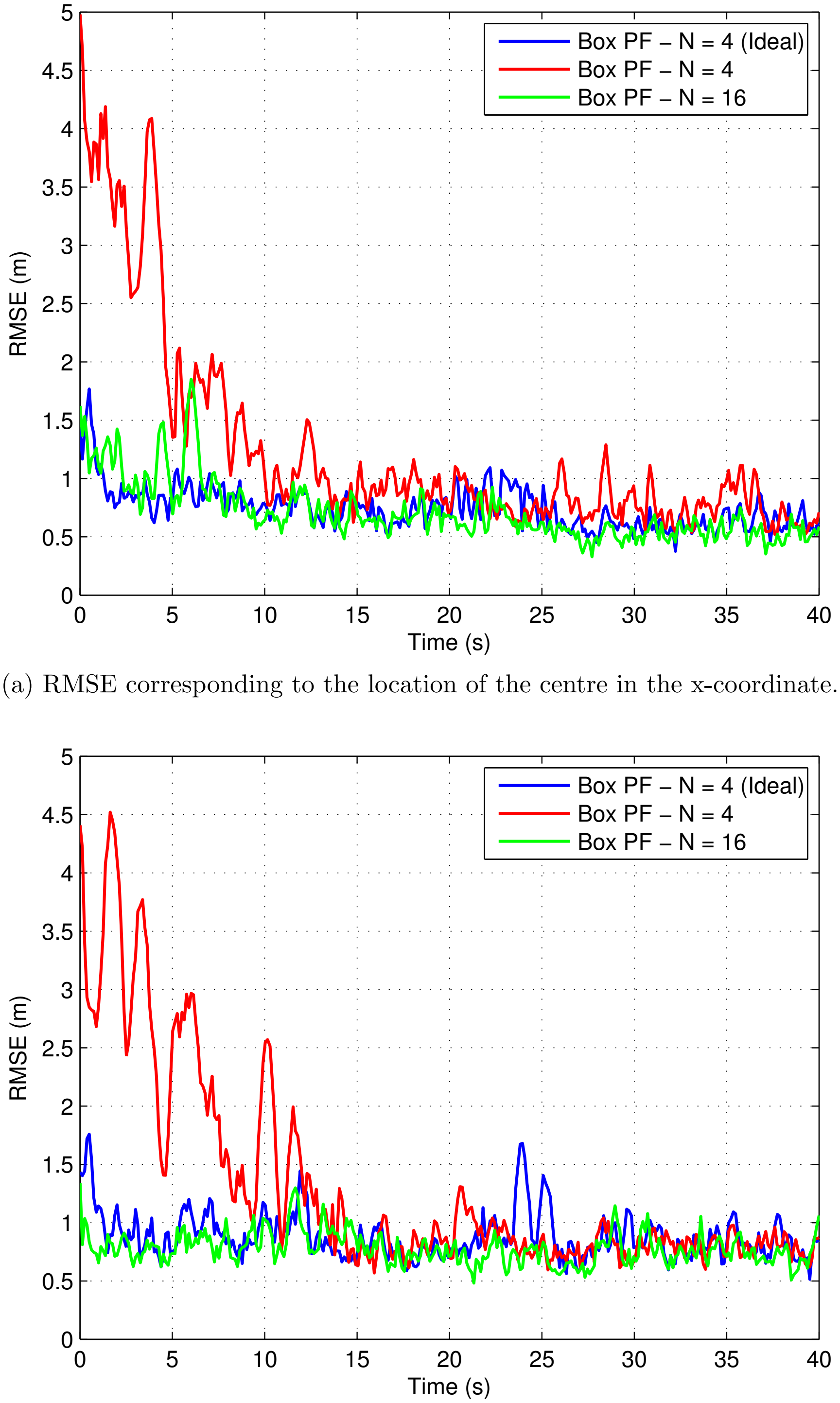
Table 3.3: MATLAB computational time corresponding to the results in Figure 3.4.

Number of Box Particles	Computation Time (s)
4	13.47
16	25.22

The third set of results focuses on the effect of jointly estimating the crowd and clutter measurement rates on the Box PF performance. This is compared with the performance of the Box PF for the ideal case where crowd and clutter measurement rates are known. The resultant RMSE values are illustrated in Figure 3.4. The computational cost comparison for these results are presented in Table 3.3. The joint estimation results in an increase in the time required to lock onto the crowd, however, this is overcome by increasing the number of box particles at the cost of an increased computational burden.

3.7.4 The Realistic Crowd Simulator Results

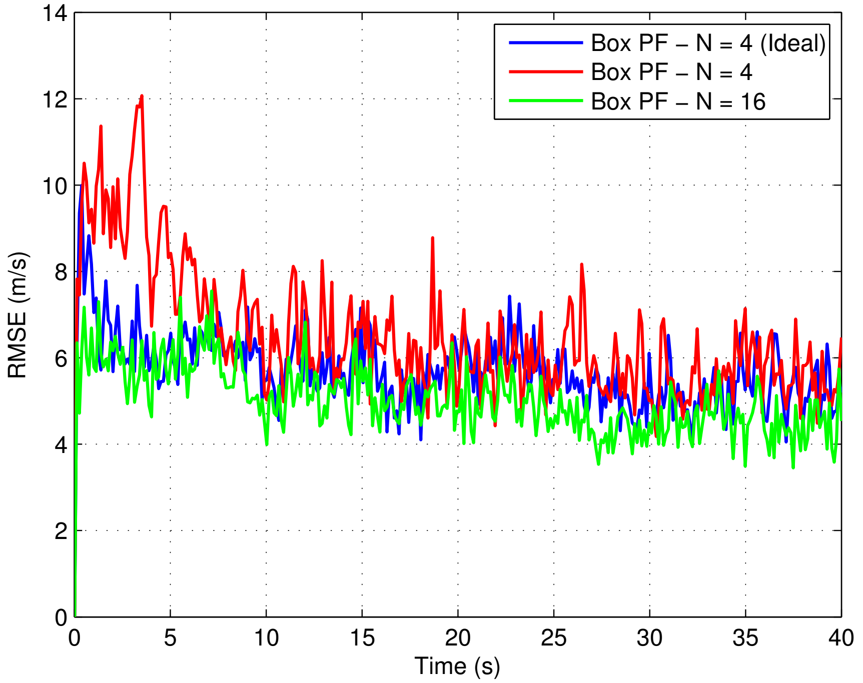
In the realistic crowd simulator the crowd moves through a corridor which consists of a bottleneck. The crowd is initialised at the entrance of the bottleneck. This is illustrated in Figure 3.5. In this section a comparison between the Box PF and CPF is presented to illustrate the filters operation on the realistic crowd simulator. The SIR PF is not included since it is incapable of operating without knowledge of the crowd and clutter measurements which are not available in a realistic situation. The parameters for the simulations are as follows:



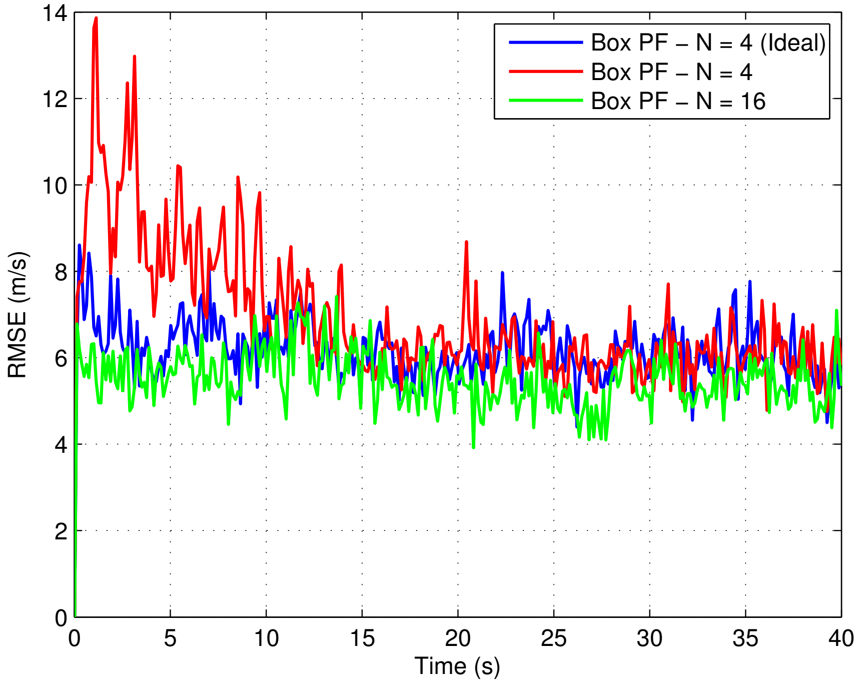
(a) RMSE corresponding to the location of the centre in the x-coordinate.

(b) RMSE corresponding to the location of the centre in the y-coordinate.

Figure 3.4: Comparison of the RMSE for the states of the Box PF with crowd and clutter rate estimation.

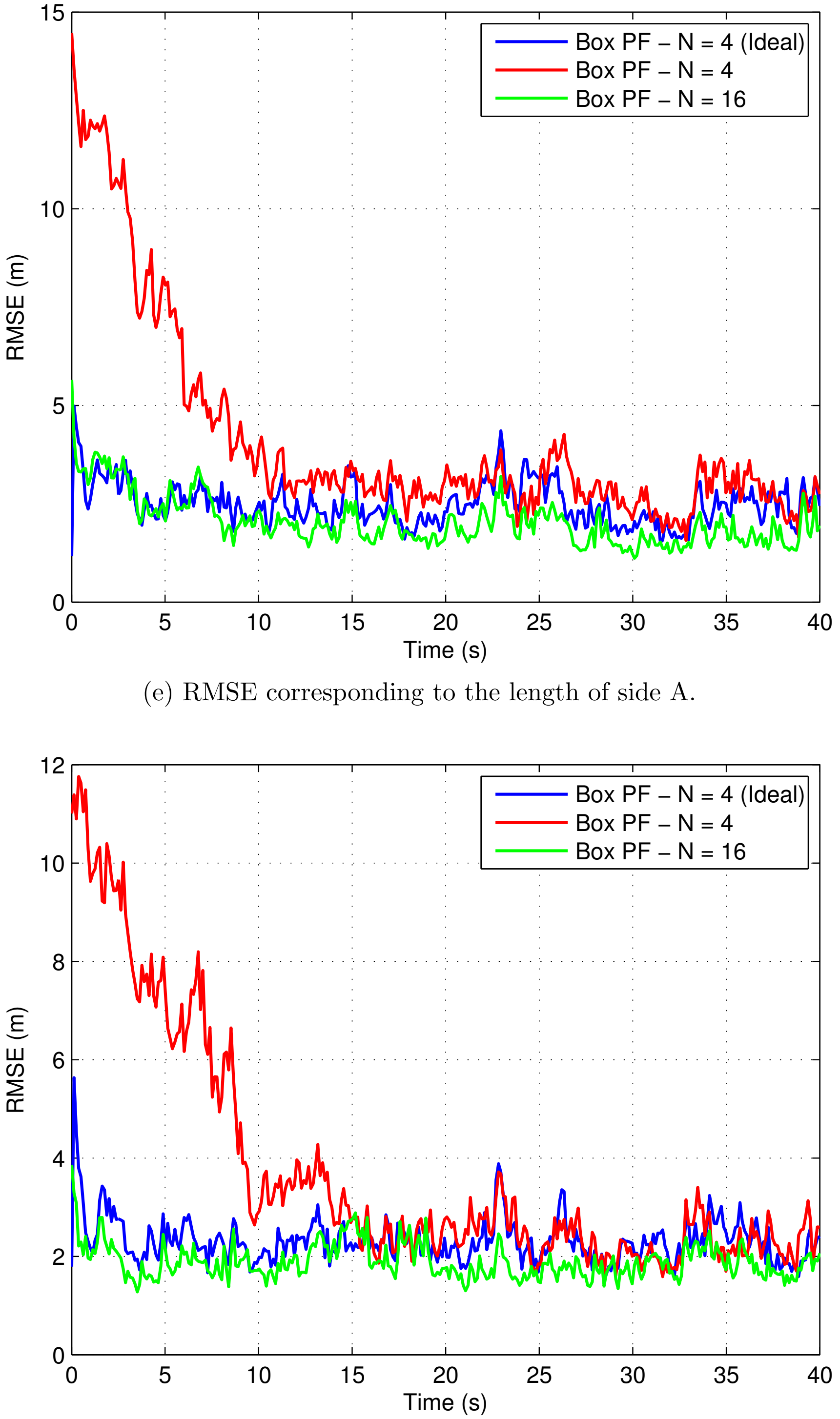


(c) RMSE corresponding to the velocity of the centre in the x-coordinate.



(d) RMSE corresponding to the velocity of the centre in the y-coordinate.

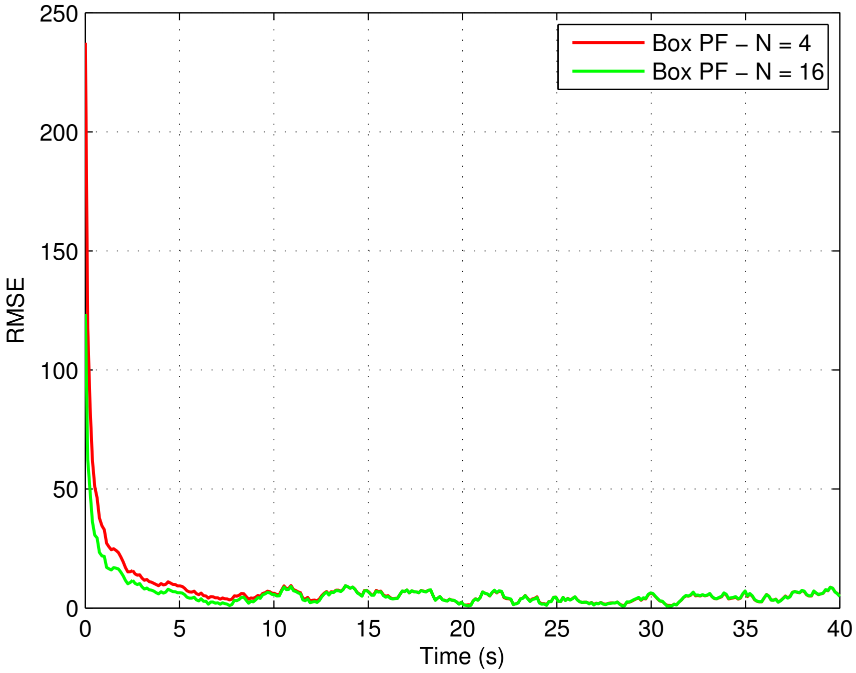
Figure 3.4: Comparison of the RMSE for the states of the Box PF with crowd and clutter rate estimation. (cont.)



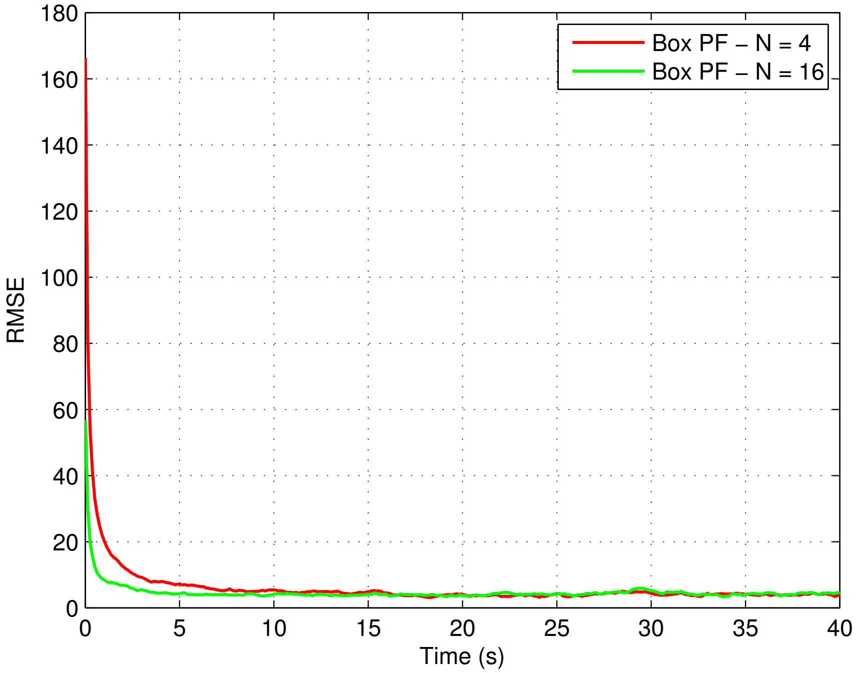
(e) RMSE corresponding to the length of side A.

(f) RMSE corresponding to the length of side B.

Figure 3.4: Comparison of the RMSE for the states of the Box PF with crowd and clutter rate estimation. (cont.)



(g) RMSE corresponding to the crowd measurement rate.



(h) RMSE corresponding to the clutter measurement rate.

Figure 3.4: Comparison of the RMSE for the states of the Box PF with crowd and clutter rate estimation. (cont.)

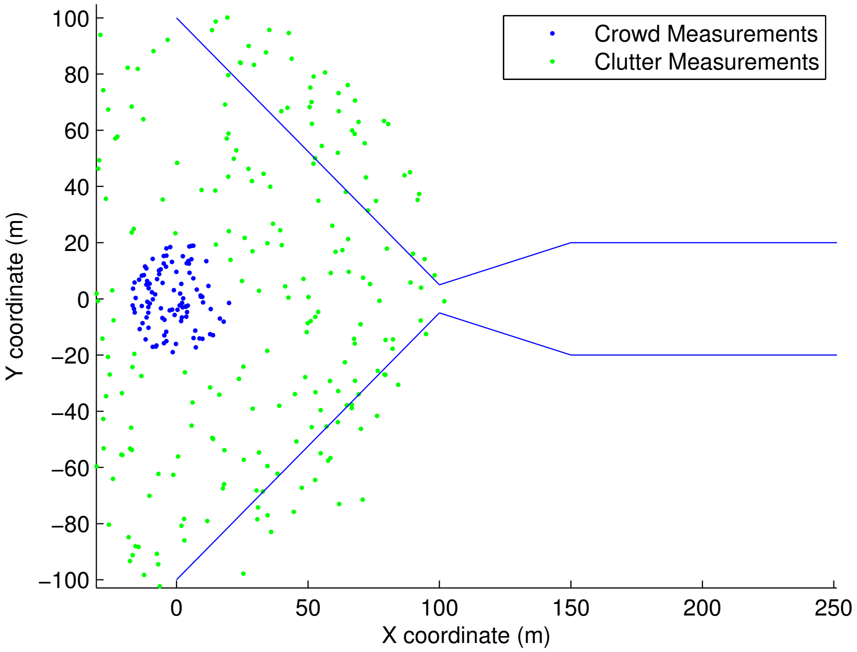


Figure 3.5: Initialisation of the realistic crowd simulator.

- *Simulation*: The number of entities in the crowd: $N_T = 100$, Simulation time duration: $T = 150$ s, Sampling time, $T_s = 0.125$ s,

- *Sensor*: Measurement uncertainty: $\sigma_{z_1} = \sigma_{z_2} = 0.1$ m, Clutter parameters: Clutter density, $\rho = 1 \times 10^{-3}$, Clutter area = Circular region with radius of 100 m about the centre of the crowd,

- *Filter Parameters*: Number of box particles: $N = 16$, Number of CPF particles: $N = 1000$, Crowd centre dynamics parameters: Velocity correlation time constant, $T_{cv} = 30$ s, Velocity standard deviation parameters, $\sigma_{v,x} = \sigma_{v,y} = 1$ m/s, Group extent dynamics parameters $\sigma_a = \sigma_b = 0.1$ m per time step. Measurement uncertainties: matched to the sensor parameters. Initialisation: Initialised in the same manner as for the rectangular group object simulator.

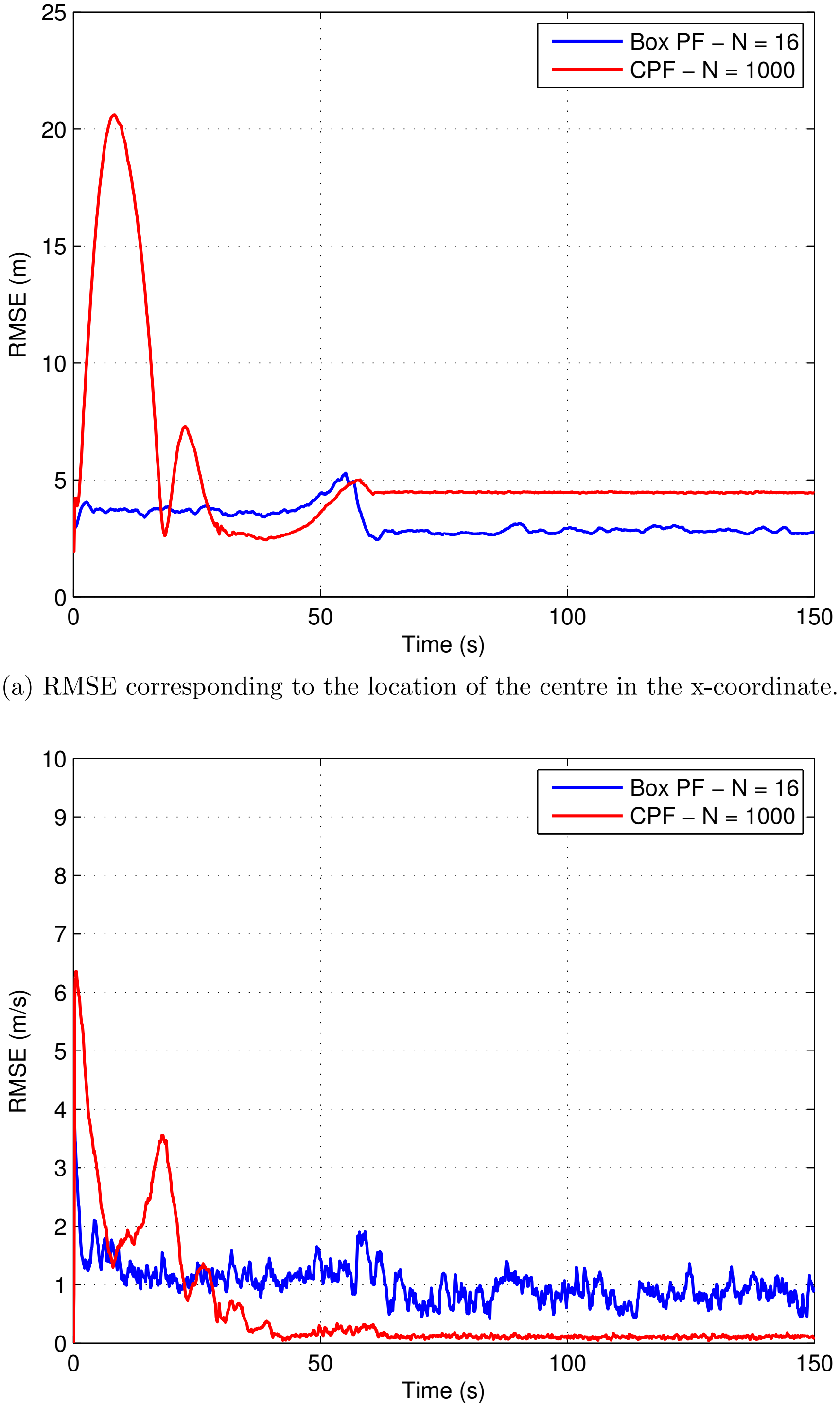
The RMSE for each state, based on the ground truth extracted from the crowd measurements, are illustrated in Figure 3.6 for both the Box PF and CPF. The number of MC runs is 50. The crowd moves through the bottleneck in the vicinity of 60 seconds. Initially, the CPF struggles to lock on to the crowd. Once locked, and after the crowd has passed through the bottleneck, the RMSE for the length corresponding to side a is increased. This is due to several crowd entities spreading out further away from the majority of the crowd and thus being mistaken as a clutter measurements.

3.8 Summary

In this chapter the challenging problem of tracking a large number of objects, referred to as a crowd, was considered. In a crowd, it is assumed that the objects maintain a certain pattern of motion that is sufficiently described by the kinematics of the centroid of the crowd. From a modelling perspective, this allows for the crowd to be treated as a single entity with an extent which varies with time. The state space model consists of the kinematics of the centroid of the crowd, the dynamic parameters which describe a shape approximating the extent of the crowd, and the measurement rates.

Two novel methods, based on the Bayesian framework, were presented. Both techniques rely on particle approximations since complexities in the state space model prevent an analytical solution from existing. The first method is based on a Box PF approach. The Box PF approach is a combination of SMC and interval analysis. The Box PF method relies on the concept of box particles which have a non-zero volume in the state space. This translates into a much lower number of box particles being required to represent the posterior state pdf when compared to standard particle approaches. Previously, a rigorous solution for the Box PF approach has only been available for the point target model without clutter. In this chapter a solution is derived for the crowd tracking problem.

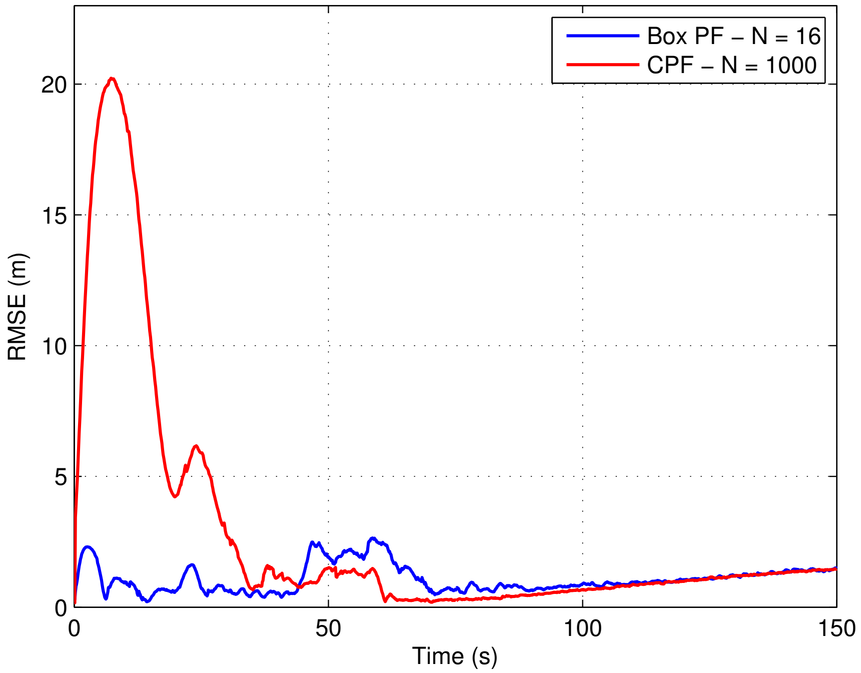
The second method presented is based on the CPF approach. The proposed CPF is able to deal with multiple measurements, including a high level of clutter. The CPF is able to resolve the data association problem without the need of estimating



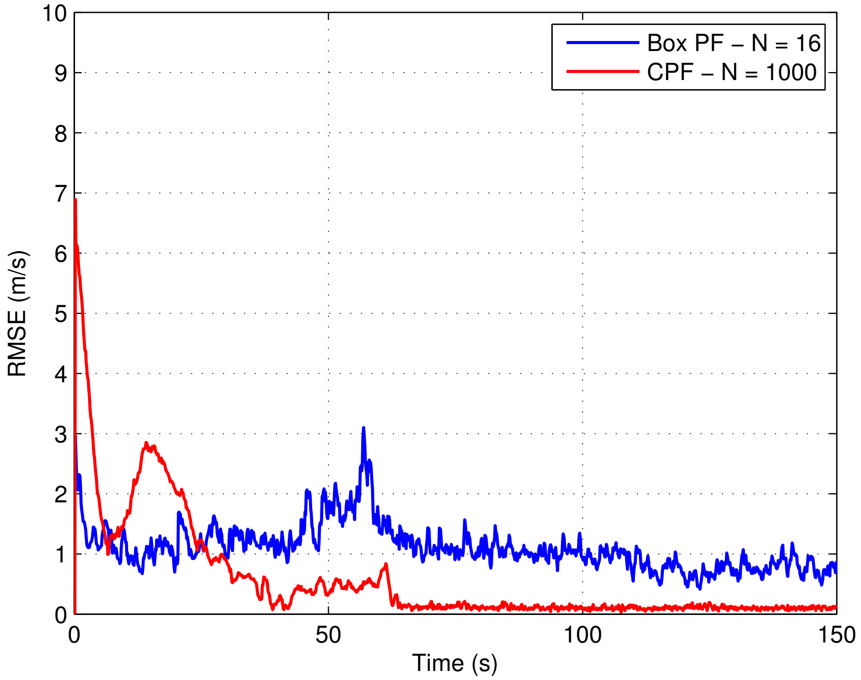
(a) RMSE corresponding to the location of the centre in the x-coordinate.

(b) RMSE corresponding to the velocity of the centre in the x-coordinate.

Figure 3.6: RMSE of the Box PF and CPF estimates for the realistic crowd simulator.

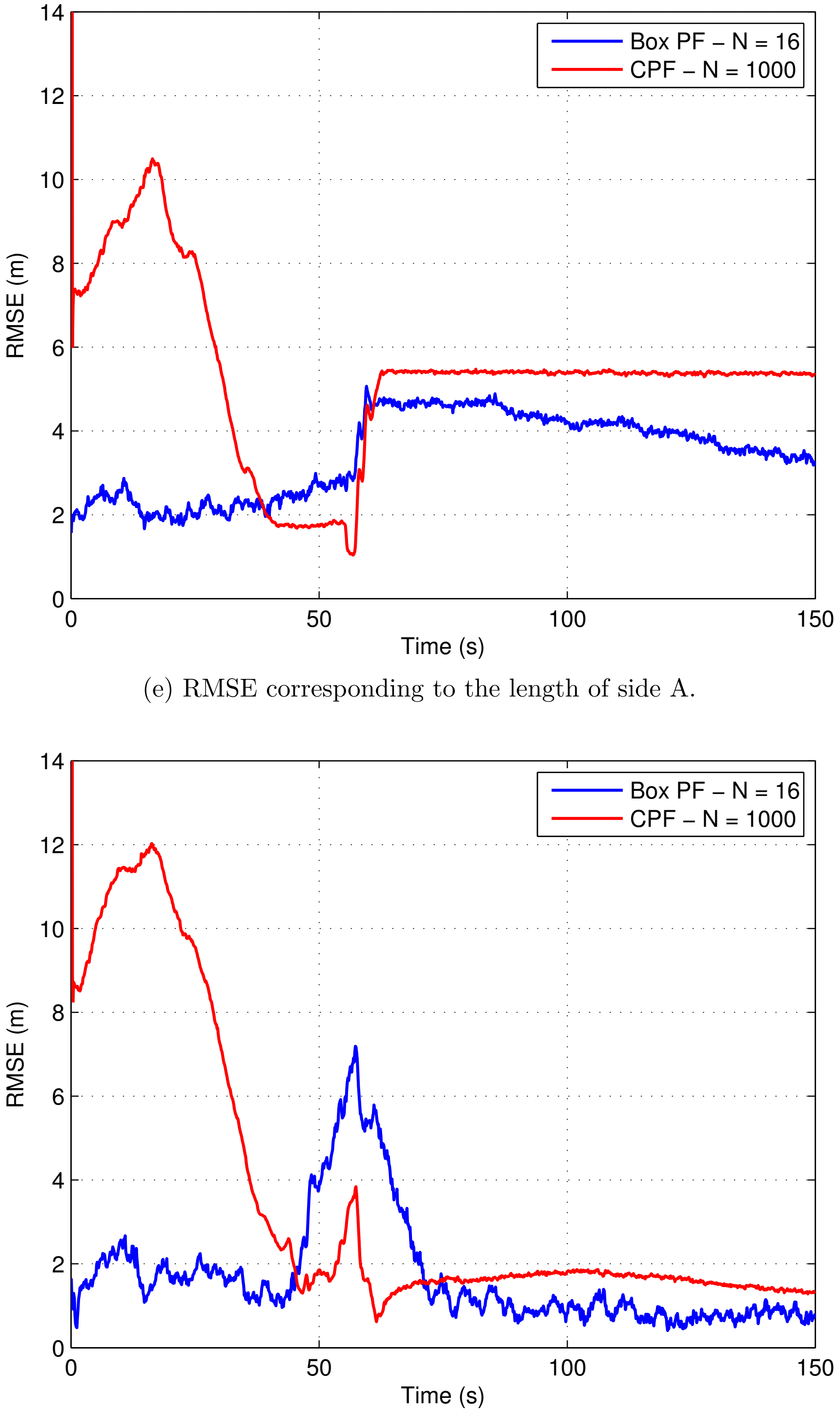


(c) RMSE corresponding to the location of the centre in the y-coordinate.



(d) RMSE corresponding to the velocity of the centre in the y-coordinate.

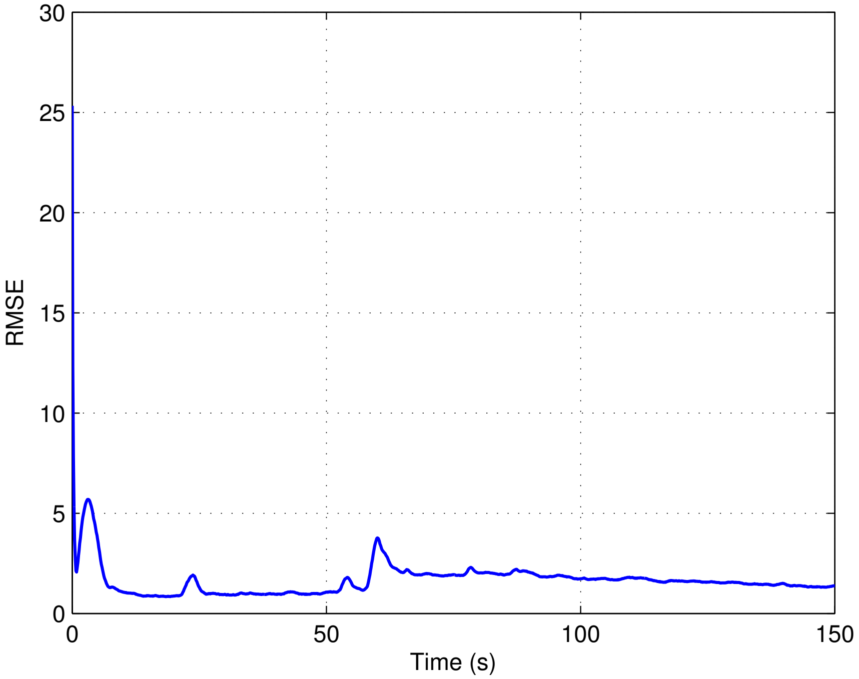
Figure 3.6: RMSE of the Box PF and CPF estimates for the realistic crowd simulator. (cont.)



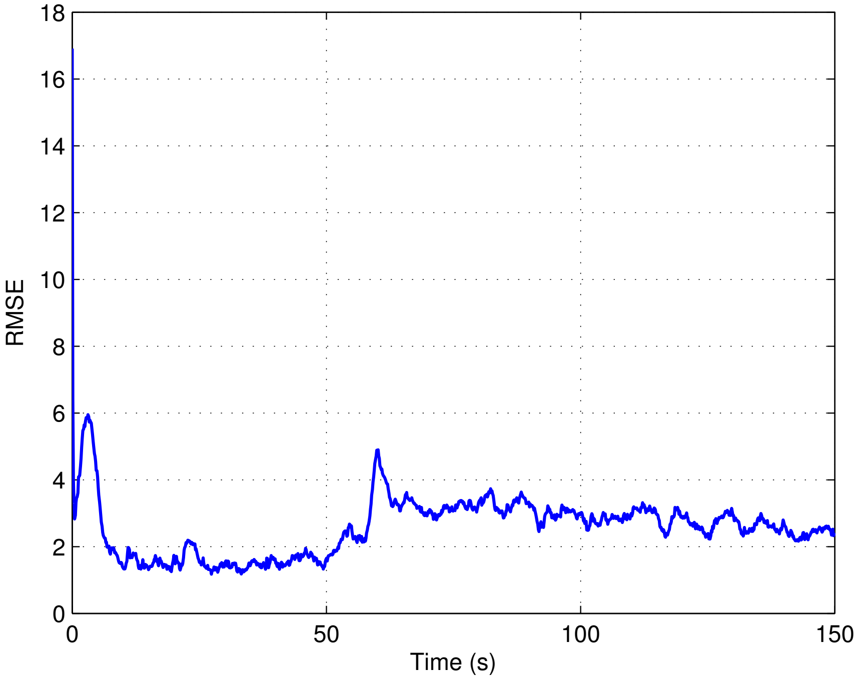
(e) RMSE corresponding to the length of side A.

(f) RMSE corresponding to the length of side B.

Figure 3.6: RMSE of the Box PF and CPF estimates for the realistic crowd simulator. (cont.)



(g) RMSE corresponding to the crowd measurement rate estimated by the Box PF.



(h) RMSE corresponding to the clutter measurement rate estimated by the Box PF.

Figure 3.6: RMSE of the Box PF and CPF estimates for the realistic crowd simulator. (cont.)

the measurement rates.

The performance of both methods was analysed in two experiments. The first experiment was based on a simulated crowd which directly matched the state space model. The second experiment was based on a realistic crowd simulator where a group of objects moved through a bottle neck. The experiment results showed that the Box PF and CPF require a significantly smaller number of (box) particles than the SIR PF, and are also more robust to initialisation errors.

Chapter 4

MULTIPLE EXTENDED OBJECT TRACKING

The focus of the previous chapter is on the crowd tracking problem where solutions are presented based on the proposed Box PF and CPF. In this chapter the related problem of extended object tracking is considered. This problem can be formulated in the same way, however, the focus in this chapter is on dealing with the unique challenges which are present when considering multiple extended objects. The superior performance of the Box PF is illustrated in Section 3.7 thus motivating the work in this chapter which focuses on a Box PF method for multiple extended object tracking. For the first time it is shown how interval based approaches can be used to deal with data association by reducing the computational complexity in the data association process. The method presented in this chapter is based on a novel SMC approach for multiple extended object tracking. This is in contrast to several RFS approaches which have been proposed. An overview of these approaches has been recently presented in [57]. In Section 4.1 details of the general problem formulation for multiple extended object tracking are given. Section 4.2 includes more specific details on circular extended object modelling. In Section 4.3 details of the BP PF are presented. In Section 4.4 the Box PF for multiple object tracking is presented. An evaluation of the effectiveness of the proposed method is described in Section 4.5.

4.1 Multiple Extended Object Tracking as State and Parameter Estimation

The multiple extended object tracking problem can be formulated as joint state and parameter estimation in the presence of multiple measurements coming simultaneously from the border or surface of multiple objects. It is also considered that some of the measurements may not originate from an object, in this case referred to as clutter. The latent states of all the objects are combined into a single state vector with fixed dimension, $\boldsymbol{x}_k = (\boldsymbol{x}_{1,k}^\top, \boldsymbol{x}_{2,k}^\top, \ldots, \boldsymbol{x}_{N_T,k}^\top)^\top$, N_T represents the maximum number of extended objects.

In extended object tracking, each extended object sub-state vector is defined as $\boldsymbol{x}_{i,k} = \left(\boldsymbol{X}_{i,k}^\top, \boldsymbol{\Theta}_{i,k}^\top\right)^\top$. The subset of states, $\boldsymbol{X}_{i,k}$, includes all the states related to the kinematics (e.g. position coordinates, velocities) of the centroid of motion of the object. This typically includes the position, velocity and any other higher order position derivatives defined by the motion model. The subset of states, $\boldsymbol{\Theta}_{i,k}$, includes all the parameters used to model the extent of the object. This allows for the extent of the object to be represented by a variety of parametric shapes.

In a similar fashion to Chapter 3, an unordered set of measurements is collected at each time step, k, $\boldsymbol{Z}_k = \{\boldsymbol{z}_{1,k}, \boldsymbol{z}_{2,k} \ldots \boldsymbol{z}_{M_k,k}\}$. Here the measurements also originate from either an object or clutter, however, since we have multiple objects the overall number of measurements is given by $M_k = \sum_i^{N_T} M_{T,k}^i + M_{C,k}$.

4.1.1 Birth and Disappearance of Extended Objects

In multiple object tracking, an object may enter or leave the area observed by the sensors at any time. This is referred to as the birth or death of an object, respectively. To cater for a varying number of extended objects, a binary variable representing the existence of each extended object is introduced, inspired by [118, 107], $\boldsymbol{e}_k = (e_{1,k}, e_{2,k} \ldots e_{N_T,k})^\top$ with $e_{i,k} \in \{0, 1\}$. The existence variable, $e_{i,k}$, evolves according to a Markov chain with the following property,

$$p(e_{i,k}|e_{i,k-1} = \ell) = \begin{cases} P_e & \text{when } e_{i,k} = \ell, \\ 1 - P_e & \text{otherwise,} \end{cases} \tag{4.1}$$

where P_e represents the probability of existence.

4.1.2 Problem Formulation within the Bayesian Framework

Similarly to the steps in (2.4) and (2.5), the posterior state pdf for the extended state vector, $\boldsymbol{s}_k = (\boldsymbol{x}_k^\top, \boldsymbol{e}_k^\top)^\top$, can be updated sequentially based on a prediction step,

$$p(\boldsymbol{s}_k|\boldsymbol{Z}_{1:k-1}) = \int p(\boldsymbol{s}_k|\boldsymbol{s}_{k-1})p(\boldsymbol{s}_{k-1}|\boldsymbol{Z}_{1:k-1})d\boldsymbol{s}_{k-1}, \tag{4.2}$$

followed by an update step

$$p(\boldsymbol{s}_k|\boldsymbol{Z}_{1:k}) = \frac{p(\boldsymbol{Z}_k|\boldsymbol{s}_k)p(\boldsymbol{s}_k|\boldsymbol{Z}_{1:k-1})}{p(\boldsymbol{Z}_k|\boldsymbol{Z}_{1:k-1})}, \tag{4.3}$$

where $p(\boldsymbol{s}_k|\boldsymbol{Z}_{1:k-1})$ is the predictive posterior state pdf, $p(\boldsymbol{s}_k|\boldsymbol{s}_{k-1})$ is the state transition pdf, $p(\boldsymbol{Z}_k|\boldsymbol{s}_k)$ is the likelihood function and $p(\boldsymbol{Z}_k|\boldsymbol{Z}_{1:k-1})$ is a normalisation factor.

4.2 Circular Extended Object Modelling

Consider the tracking of extended objects in a two dimensional plane with a circular extent, without loss of generality. Each objects system sub-states corresponding to the kinematics of the object is a vector, $\boldsymbol{X}_{i,k} = (x_{i,k}, \dot{x}_{i,k}, y_{i,k}, \dot{y}_{i,k})^\top$, which includes the position coordinates, $(x_{i,k}, y_{i,k})$, and respective velocity components, $(\dot{x}_{i,k}, \dot{y}_{i,k})$, of the extended object. In this scenario the subset of states that includes the parameters used to model the extent of the object reduces to a scalar representing the radius of the object, $\Theta_{i,k} = R_{i,k}$.

4.2.1 State Transition Representation

The state transition pdf can be further factorised as:

$$p(\boldsymbol{s}_k|\boldsymbol{s}_{k-1}) = \prod_{i=1}^{N_T} p(\boldsymbol{x}_{i,k}|\boldsymbol{x}_{i,k-1}, e_{i,k}, e_{i,k-1})p(e_{i,k}|e_{i,k-1}). \tag{4.4}$$

The sub-state transition pdf for the ith object is defined as:

$$p(\boldsymbol{x}_{i,k}|\boldsymbol{x}_{i,k-1}, e_{i,k}, e_{i,k-1}) = \begin{cases} p_b(\boldsymbol{x}_{i,k}) & \{e_{i,k}, e_{i,k-1}\} = \{1,0\} \\ p_d(\boldsymbol{x}_{i,k}) & \{e_{i,k}\} = \{0\} \\ p(\boldsymbol{x}_{i,k}|\boldsymbol{x}_{i,k-1}) & \{e_{i,k}, e_{i,k-1}\} = \{1,1\}, \end{cases} \tag{4.5}$$

where $p_b(\boldsymbol{x}_{i,k})$ and $p_d(\boldsymbol{x}_{i,k})$ is the probability of an object birth and death respectively, and $p(\boldsymbol{x}_{i,k}|\boldsymbol{x}_{i,k-1})$ represents the motion of existent extended objects. In this chapter the nearly constant velocity motion model [11] is considered as a representation of this motion. In two dimensions, the state of the object is then given by

$$\boldsymbol{X}_{i,k} = \boldsymbol{A}\boldsymbol{X}_{i,k-1} + \boldsymbol{\Gamma}\boldsymbol{\eta}_X, \tag{4.6}$$

where $\boldsymbol{A} = diag(\boldsymbol{A}_1, \boldsymbol{A}_1)$, $\boldsymbol{A}_1 = \begin{pmatrix} 1 & T_s \\ 0 & 1 \end{pmatrix}$, $\boldsymbol{\Gamma} = \begin{pmatrix} T_s^2/2 & T_s & 0 & 0 \\ 0 & 0 & T_s^2/2 & T_s \end{pmatrix}^\top$, T_s is the sampling interval and $\boldsymbol{\eta}_X \sim \mathcal{N}(0, \boldsymbol{Q}_X)$ is the system dynamics noise, with covariance matrix $\boldsymbol{Q}_X$. It is assumed that $\boldsymbol{Q}_X = diag(\boldsymbol{Q}_1\sigma_x^2, \boldsymbol{Q}_1\sigma_y^2)$, where $\boldsymbol{Q}_1 = \begin{pmatrix} T_s^4/4 & T_s^3/2 \\ T_s^3/2 & T_s^2 \end{pmatrix}$ and σ_x and σ_y are the standard deviations for the x and y coordinate, respectively. The evolution model for the extent parameter is assumed to be

$$\Theta_{i,k} = \Theta_{i,k-1} + \eta_\Theta, \tag{4.7}$$

where $\eta_\Theta \sim \mathcal{N}(0, \sigma_R^2)$.

4.2.2 Likelihood Representation

The likelihood in equation (4.3) can be calculated in various ways with different data association algorithms. One of the best approaches, which alleviates the combinatorial complexity in data association, is proposed in [48]. It adopts Poisson assumptions of the number of measurements originated from the objects and the number of clutter points. This is an extension of the generalised likelihood function used in Section 3.4, which is equivalent to only considering a single extended object. This generalised likelihood function is of the form

$$p(\boldsymbol{Z}_k|\boldsymbol{s}_k) = \frac{e^{-\sum_{i\in\mathcal{J}}\lambda_{T,i}}}{M_k!}\prod_{m=1}^{M_k}\left(\rho + \sum_{i\in\mathcal{J}}\lambda_{T,i}p(\boldsymbol{z}_{m,k}|\boldsymbol{x}_{i,k})\right), \tag{4.8}$$

where $\mathcal{J}$ denotes a set corresponding to the index of active objects at the current time step, $\rho = \frac{\lambda_C}{A_C}$ is the clutter density, and $p(\boldsymbol{z}_{m,k}|\boldsymbol{x}_{i,k})$ is the measurement likelihood for a single object.

Consider a scenario where multiple sensors observe the extended objects. The state of sensor s is given by $\widetilde{\boldsymbol{x}}_{s,k} = (\widetilde{x}_{s,k}, \widetilde{y}_{s,k}, \alpha_{1,k}, \alpha_{2,k})^\top$, where $(\widetilde{x}_{s,k}, \widetilde{y}_{s,k})$ are the sensor position coordinates, $\alpha_{1,k}$ and $\alpha_{2,k}$ represent two parameters defining the angle of view of the sensor. When an extended object is visible from sensor s, the sensor states and object system sub-states geometrically define the visible border of the extended object, $\mathcal{V}_k(\boldsymbol{x}_{i,k}, \widetilde{\boldsymbol{x}}_{s,k})$. The angles $\alpha_{1,k}$ and $\alpha_{2,k}$, geometrically define another two angles $\theta_{1,k}$, $\theta_{2,k}$ that specify the visible border of the extended object,

$$\mathcal{V}_k(\boldsymbol{x}_{i,k}, \widetilde{\boldsymbol{x}}_{s,k}) = (x_{i,k} + R_{i,k}\cos(\theta_k), y_{i,k} + R_{i,k}\sin(\theta_k)), \tag{4.9}$$

where $\theta_k \in [\theta_{1,k}, \theta_{2,k}]$. For a single time instance k, the jth measurement is related to a specific point on the visible surface of an extended object. This point is referred to as the jth point source and denoted by $\boldsymbol{V}_{i,k}^j$. Selecting a specific value from the angle set i.e. $\theta_k^j \in [\theta_{1,k}, \theta_{2,k}]$, results in the following description of the jth point source,

$$\boldsymbol{V}_{i,k}^j = (x_{i,k} + R_{i,k}\cos(\theta_k^j), y_{i,k} + R_{i,k}\sin(\theta_k^j)). \tag{4.10}$$

The measurement $\boldsymbol{z}_{j,k}$ collected from a sensor is in polar coordinates and consists of range d_k^j and bearing β_k^j. The observation equation with respect to the jth measurement can then be written in the form:

$$\boldsymbol{z}_{j,k} = (d_k^j, \beta_k^j)^\top = h(\boldsymbol{V}_{i,k}^j) + \boldsymbol{w}_k^j, \tag{4.11}$$

where $h(\cdot)$ is a non-linear function

$$h(\boldsymbol{V}_{i,k}^j) = \begin{pmatrix} \sqrt{(x_{i,k} + R_{i,k}\cos(\theta_k^j) - \widetilde{x}_{s,k})^2 + (y_{i,k} + R_{i,k}\sin(\theta_k^j) - \widetilde{y}_{s,k})^2} \\ \tan^{-1}\left(\frac{y_{i,k}+R_{i,k}\sin(\theta_k^j)-\widetilde{y}_{s,k}}{x_{i,k}+R_{i,k}\cos(\theta_k^j)-\widetilde{x}_{s,k}}\right) \end{pmatrix}. \tag{4.12}$$

The measurement noise $\boldsymbol{w}_k^j = (w_{d,k}^j, w_{\beta,k}^j)^\top$, is assumed to be Gaussian, with a known covariance matrix $\boldsymbol{\Sigma} = diag(\sigma_d^2, \sigma_\beta^2)$.

The measurement likelihood for a single object consists of the combination of two pdfs,

$$p(\boldsymbol{z}_{j,k}|\boldsymbol{x}_{i,k}) = \int p(\boldsymbol{z}_{j,k}|\boldsymbol{V}_{i,k}^j)p(\boldsymbol{V}_{i,k}^j|\boldsymbol{x}_{i,k})d\boldsymbol{V}_{i,k}^j, \tag{4.13}$$

where $p(\boldsymbol{z}_{j,k}|\boldsymbol{V}_{i,k}^j)$ denotes the likelihood of the measurement given a point source, and $p(\boldsymbol{V}_{i,k}^j|\boldsymbol{x}_{i,k})$ is the likelihood of the point source given the object sub-states.

One simplified assumption about the distribution of the point sources of measurements, given the object sub-states and the sensor states, is a uniform distribution along the region $\mathcal{V}_k(\boldsymbol{x}_{i,k}, \widetilde{\boldsymbol{x}}_{s,k})$, visible from the sensor position, i.e.

$$p(\boldsymbol{V}_{i,k}^j|\boldsymbol{x}_{i,k}) = \mathcal{U}_{\mathcal{V}_k(\boldsymbol{x}_{i,k},\widetilde{\boldsymbol{x}}_{s,k})}(\boldsymbol{V}_k) = \frac{1}{||\mathcal{V}_k(\boldsymbol{x}_{i,k}, \widetilde{\boldsymbol{x}}_{s,k})||}, \tag{4.14}$$

where $\mathcal{U}_{\mathcal{V}_k(\boldsymbol{x}_{i,k},\widetilde{\boldsymbol{x}}_{s,k})}(\cdot)$ is a uniform pdf with the support $\mathcal{V}_k(\boldsymbol{x}_{i,k}, \widetilde{\boldsymbol{x}}_{s,k})$ and $||\mathcal{V}_k(\boldsymbol{x}_{i,k}, \widetilde{\boldsymbol{x}}_{s,k})||$ denotes some measure of the region $\mathcal{V}_k(\boldsymbol{x}_{i,k}, \widetilde{\boldsymbol{x}}_{s,k})$, such as the Euclidean norm.

A typical assumption about the noise associated with a sensor is a Gaussian distribution, i.e. $p(\boldsymbol{z}_{j,k}|\boldsymbol{V}_{i,k}^j) = \mathcal{N}(\boldsymbol{z}_k^j; h(\boldsymbol{V}_{i,k}^j), \boldsymbol{\Sigma})$, where $h(\boldsymbol{V}_{i,k}^j)$ is the mean, and $\boldsymbol{\Sigma}$ is the covariance matrix.

4.3 The Border Parameterised Particle Filter

No analytical solution exists for the prediction and updating of the posterior state pdf in (4.2) and (4.3) due to the complexities in the state space model. Therefore, MC methods that approximate the posterior state pdf are considered. One such method is the SIR PF, as described in Section 2.1.1, which approximates the posterior state pdf with a weighted set of particles [8]

$$p\left(\boldsymbol{s}_k|\boldsymbol{Z}_{1:k}\right) = \sum_{p=1}^{N} w_k^{(p)} \delta\left(\boldsymbol{s}_k - \boldsymbol{s}_k^{(p)}\right), \tag{4.15}$$

where $\delta(\cdot)$ is the Dirac delta function, and the weights, $\{w_k^{(p)}\}_{p=1}^{N}$, are normalised so that $\sum_p w_k^{(p)} = 1$.

However, since the measurement likelihood for a single object, $p(\boldsymbol{z}_{j,k}|\boldsymbol{x}_{i,k})$, is analytically intractable, a MC method is used to approximate it. This modified SIR PF is referred to as the BP PF [92]. Measurement sources from the visible surface $\mathcal{V}_k(\boldsymbol{x}_{i,k}, \widetilde{\boldsymbol{x}}_{s,k})$ of each existent object are required to be sampled. For each particle existent object subspace, $\boldsymbol{x}_{i,k|k-1}^{(p)}$, the support of $p(\boldsymbol{V}_{i,k}^{j}|\boldsymbol{x}_{i,k|k-1}^{(p)})$ is defined by a uniform distribution over the angular range $[\theta_{1,k}, \theta_{2,k}]$ of the visible border $\mathcal{V}_k(\boldsymbol{x}_{i,k|k-1}^{(p)}, \widetilde{\boldsymbol{x}}_{s,k})$ with respect to the object center. Then a sampled point source can be obtained by first sampling from:

$$\left\{\ \theta_k^{(b,f)}\right\}_{b=1,f=1}^{N,F} \sim \left(\mathcal{U}_{[\theta_{1,k},\theta_{2,k}]}(\theta_k)\right), \tag{4.16}$$

followed by the substitution of $\left\{\theta_k^{(b,f)}\right\}_{b=1,f=1}^{N,F}$ into equation (4.10), resulting in a random set of samples denoted as $\mathcal{J}_k = \left\{\boldsymbol{V}_{i,k}^{j,(b,f)}\right\}_{b=1,f=1}^{N,\ F}$, where F is the number of samples from the object border. The Monte Carlo approximation for the measurement likelihood for a single object is then given by:

$$\begin{aligned} p(\boldsymbol{z}_{j,k}|\boldsymbol{x}_{i,k|k-1}^{(p)}) &= \int p(\boldsymbol{z}_{j,k}|\boldsymbol{V}_{i,k}^{j})p(\boldsymbol{V}_{i,k}^{j}|\boldsymbol{x}_{i,k|k-1}^{(p)})d\boldsymbol{V}_{i,k}^{j}, \\ &\approx \frac{1}{F}\sum_{\boldsymbol{V}_{i,k}^{j}\in\mathcal{J}_k} p(\boldsymbol{z}_{j,k}|\boldsymbol{V}_{i,k}^{j}). \end{aligned} \tag{4.17}$$

The BP PF algorithm for multiple extended object tracking is summarised in Algorithm 10.

Algorithm 10 The Border Parameterised Particle Filter for Multiple Extended Object Tracking

```
***Initialisation***
Initialise the set of particles, ${\boldsymbol{s}_0^{(p)}}_{p=1}^{N}$, sampling from the prior distribution.
**for** $k = 1,...,T$ **do**
    ***Prediction***
    Propagate the particles, $\{\boldsymbol{s}_{k-1}^{(p)}\}_{p=1}^{N}$ with the state transition pdf (equation (4.4)) to obtain the predicted particles, $\{\boldsymbol{s}_{k|k-1}^{(p)}\}_{p=1}^{N}$.
    ***Measurement Update***
    Upon the receipt of new measurements:
    Evaluate the measurement likelihood for a single object, $p(\boldsymbol{z}_{j,k}|\boldsymbol{x}_{i,k|k-1}^{(p)})$, according to (4.17) for all the measurements within the observability region of the sensor and all objects.
    Calculate the weights $\{w_k^{(p)}\}_{p=1}^{N}$ using terms from the previous step and equation (4.8).
    ***Output***
    Calculate the estimated state vector $\widehat{\boldsymbol{x}}_k$ based on the maximum weight:
        $$\widehat{\boldsymbol{x}}_k = \arg\max_{\boldsymbol{x}_k} w_k^{(p)} \tag{4.18}$$
    ***Resampling*** If $ESS \leq N_{thresh}$ (with e.g. $N_{thresh} = 2N/3$) resample the particles. Finally, reset the weights: $w_k^{(p)} = 1/N$.
**end for**
```

4.4 The Box Particle Filter for Multiple Extended Object Tracking

In contrast to the Box PF for crowd tracking derived in Section 3.4, the Box PF in this Section is developed to track multiple extended objects with measurements from the surface of the objects. It is important to note that the states of $\boldsymbol{s}_k$ corresponding to the existence variables, $\boldsymbol{e}_k$, are still considered to have zero area in the state space. The prediction step for the multiple extended object Box PF follows the same spirit as described by equations (3.23) to (3.25), with an inclusion function based on (4.6).

4.4.1 Box Particle Filter Likelihood for Multiple Extended Objects

Brute Force Approach: As noted in Section 4.2.2, the likelihood of a measurement given a point source, $p(\boldsymbol{z}_{m,k}|\boldsymbol{V}_{i,k}^m)$, is typically assumed to be Gaussian distributed. However, since the measurement noise is supposed to be bounded in the Box PF, a likelihood box is defined as a set containing the measurement and the noise boundaries. The interval measurements vector is $[\boldsymbol{z}_{j,k}] = ([d_k^j], [\beta_k^j])^\top$, where $[d_k^j]$ is the interval range and $[\beta_k^j]$ is the interval bearing of the measurement point j. One way to describe these components is:

$$\begin{aligned} [d_k^j] &= d_k^j + [-3\sigma_d, +3\sigma_d], \\ [\beta_k^j] &= \beta_k^j + [-3\sigma_\beta, +3\sigma_\beta]. \end{aligned} \tag{4.19}$$

The likelihood can then be described by a uniform distribution,

$$p(\boldsymbol{z}_{j,k}|\boldsymbol{V}_{i,k}^m) = U_{[\boldsymbol{z}_{m,k}]}\left(h\left(\boldsymbol{V}_{i,k}^m\right)\right). \tag{4.20}$$

As in the general PF, the update step for the Box PF assigns a weighting to each of the predicted box particles. However, it is also required to apply a contractor to each of the predicted box particles, as described in Section 3.3. Contraction is used to eliminate regions of the predicted box particles which are not consistent with the object emitted measurements. This is a challenging task when dealing with extended objects and clutter. To define the weight updates and contraction, it is required to derive an expression for the posterior state pdf.

Proposition 1: An alternative form of the generalised likelihood function of equation (4.8) is given by,

$$p(\boldsymbol{Z}_k|\boldsymbol{s}_k) = \frac{e^{-\sum_{i\in\mathcal{J}}\lambda_{T,i}}}{M_k!}\left(\rho^{M_k} + \sum_{m=1}^{M_k}\sum_{j=1}^{\binom{M_k}{m}}\sum_{n=1}^{|\mathcal{J}|^m}\rho^{M_k-m}\prod_{\ell=1}^{m}\lambda_{T,(\boldsymbol{b}_{m,n})_\ell}\,p(\boldsymbol{z}_{(\boldsymbol{a}_{m,j})_\ell,k}|\boldsymbol{x}_{(\boldsymbol{b}_{m,n})_\ell,k})\right), \tag{4.21}$$

where the notation $|\cdot|$ denotes the cardinality of a set, $((\boldsymbol{a}_{m,j})_{m=1}^{M_k})_{j=1}^{\binom{M_k}{m}}$ is a sequence of sequences corresponding to the index for all combinations of measurements, and $((\boldsymbol{b}_{m,n})_{m=1}^{M_k})_{n=1}^{|\mathcal{J}|^m}$ is a sequence of sequences corresponding to the index for all existent object to measurement associations. *Proof: See Appendix B.*

Example: consider a state vector for the case of when there are a maximum of three extended objects, where currently only the first and third objects exist, i.e. $\mathcal{J} = \{1, 3\}$, with two measurements. The sequences are thus defined as: $(\boldsymbol{a}_{1,1}) = (1)$; $(\boldsymbol{a}_{1,2}) = (2)$; $(\boldsymbol{a}_{2,1}) = (1, 2)$; $(\boldsymbol{b}_{1,1}) = (1)$; $(\boldsymbol{b}_{1,2}) = (3)$; $(\boldsymbol{b}_{2,1}) = (1, 1)$; $(\boldsymbol{b}_{2,2}) = (1, 3)$; $(\boldsymbol{b}_{2,3}) = (3, 1)$; $(\boldsymbol{b}_{2,4}) = (3, 3)$, resulting in the following generalised likelihood expression:

$$\begin{aligned} p(\boldsymbol{Z}_k|\boldsymbol{s}_k) = \frac{e^{-(\lambda_{T,1}+\lambda_{T,3})}}{2!}\Big(&\rho^2 + \rho\lambda_{T,1}p(\boldsymbol{z}_{1,k}|\boldsymbol{x}_{1,k}) + \rho\lambda_{T,3}p(\boldsymbol{z}_{1,k}|\boldsymbol{x}_{3,k}) + \rho\lambda_{T,1}p(\boldsymbol{z}_{2,k}|\boldsymbol{x}_{1,k}) \\ &+ \rho\lambda_{T,3}p(\boldsymbol{z}_{2,k}|\boldsymbol{x}_{3,k}) + \lambda_{T,1}^2 p(\boldsymbol{z}_{1,k}|\boldsymbol{x}_{1,k})p(\boldsymbol{z}_{2,k}|\boldsymbol{x}_{1,k}) + \lambda_{T,1}\lambda_{T,3}p(\boldsymbol{z}_{1,k}|\boldsymbol{x}_{1,k})p(\boldsymbol{z}_{2,k}|\boldsymbol{x}_{3,k}) + \\ &\lambda_{T,3}\lambda_{T,1}p(\boldsymbol{z}_{1,k}|\boldsymbol{x}_{3,k})p(\boldsymbol{z}_{2,k}|\boldsymbol{x}_{1,k}) + \lambda_{T,3}^2 p(\boldsymbol{z}_{1,k}|\boldsymbol{x}_{3,k})p(\boldsymbol{z}_{2,k}|\boldsymbol{x}_{3,k})\Big). \end{aligned} \tag{4.22}$$

This example highlights the fact that the evaluation of the generalised likelihood for a single state results in a summation of terms. Each term corresponds to a unique measurement association and for each object assigned measurement, an association with a specific object.

The posterior state pdf can be obtained through the combination of the predictive posterior state pdf and generalised likelihood:

$$\begin{aligned} p(\boldsymbol{s}_k|\boldsymbol{Z}_{1:k}) &= \frac{1}{\alpha_k} p(\boldsymbol{Z}_k|\boldsymbol{s}_k) p(\boldsymbol{s}_k|\boldsymbol{Z}_{1:k-1}), \\ &= \sum_{p=1}^{N} \frac{w_{k-1}^{(p)} e^{-\sum_{i\in\mathcal{J}(p)} \lambda_{T,i}}}{\alpha_k M_k!} \left(\rho^{M_k} U_{[\boldsymbol{s}_{k|k-1}^{(p)}]}(\boldsymbol{s}_k) + \sum_{m=1}^{M_k} \sum_{j=1}^{\binom{M_k}{m}} \sum_{n=1}^{|\mathcal{J}(p)|^m} \rho^{M_k-m} \prod_{\ell=1}^{m} \lambda_{T,(\boldsymbol{b}_{m,n})_\ell} \times \right. \\ & \left. p(\boldsymbol{z}_{(\boldsymbol{a}_{m,j})_\ell,k}|[\boldsymbol{x}_{(\boldsymbol{b}_{m,n})_\ell,k|k-1}^{(p)}]) U_{[\boldsymbol{s}_{k|k-1}^{(p)}]}(\boldsymbol{s}_k) \right). \end{aligned} \tag{4.23}$$

The expressions in the product of each of the latter terms can be further reduced based on the decomposition of the measurement likelihood for a single object, i.e. equation (4.13). For notational convenience, $(\boldsymbol{a}_{m,j})_\ell$ and $(\boldsymbol{b}_{m,n})_\ell$ are represented by a_ℓ and b_ℓ respectively,

$$p(\boldsymbol{z}_{a_\ell,k}|[\boldsymbol{x}_{b_\ell,k}^{(p)}]) U_{[\boldsymbol{s}_{k|k-1}^{(p)}]}(\boldsymbol{s}_k) = \int U_{[\boldsymbol{s}_{k|k-1}^{(p)}]}(\boldsymbol{s}_k) U_{[\boldsymbol{z}_{a_\ell,k}]}\Big(h\left(\boldsymbol{V}_{b_\ell,k}^{a_\ell}\right)\Big) \mathcal{U}_{\mathcal{V}_k(\boldsymbol{x}_{b_\ell,k},\boldsymbol{x}_{s,k})}(\boldsymbol{V}_{b_\ell,k}^{a_\ell}) d\boldsymbol{V}_{b_\ell,k}^{a_\ell}. \tag{4.24}$$

The terms within the integration form a constant function with a support being the following region

$$S_p^{a_\ell,b_\ell} = \Big\{ \boldsymbol{s}_k \in [\boldsymbol{s}_{k|k-1}^{(p)}] | \boldsymbol{V}_{b_\ell,k}^{a_\ell} \in \mathcal{V}_k(\boldsymbol{x}_{b_\ell,k}, \boldsymbol{x}_{s,k}), h\Big(\boldsymbol{V}_{b_\ell,k}^{a_\ell}\Big) \in [\boldsymbol{z}_{a_\ell,k}] \Big\}. \tag{4.25}$$

This represents a constraint and from its expression it can be deduced that the predicted supports $[\boldsymbol{s}_{k|k-1}^{(p)}]$, from the time update pdf approximation, have to be contracted with respect to the interval measurements $[\boldsymbol{Z}_k]$. This forms the basis for a CSP. The application of the CP for a circular extended object is illustrated in Algorithm 11 and Figure 4.1. The contracted box particle is represented by $[\boldsymbol{s}_k^{a_\ell,(p)}]$. It is important to note that contraction only occurs on the sub-states corresponding to the object indexed by b_ℓ, i.e. $[\boldsymbol{s}_k^{a_\ell,(p)}] = ([\boldsymbol{x}_{1,k|k-1}^{(p)}], ..., [\boldsymbol{x}_{b_\ell,k}^{a_\ell,(p)}], ..., [\boldsymbol{x}_{N_T,k|k-1}^{(p)}], [\boldsymbol{e}_{k|k-1}^{(p)}])^\top$ where $[\boldsymbol{x}_{i,k}^{a_\ell,(p)}]$ represents the sub-states of object i contracted by the measurement indexed by a_ℓ. Following the definition of the set $S_p^{a_\ell,b_\ell}$ in equation (4.25), equation

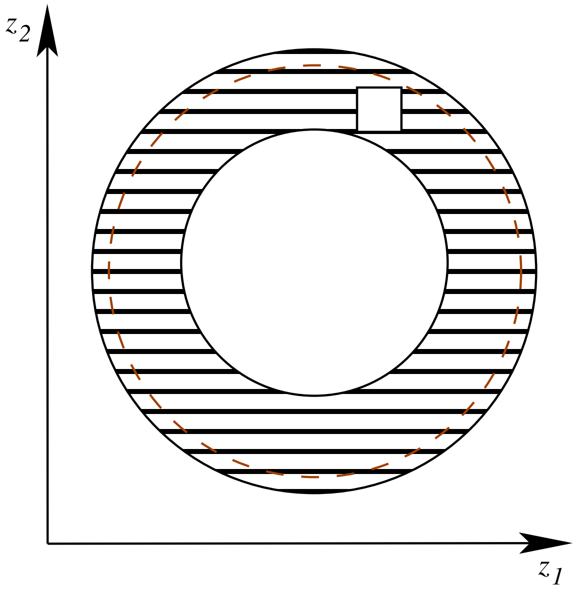


Figure 4.1: Illustration of the contraction of a box particle by a single measurement. The square box represents a measurement. The filled circular region represents the projection of a box particle sub-states for a single object to the measurement space. The dotted line illustrates the reduction in the interval shape due to contraction by the measurement.

(4.24) can be rewritten as follows

$$p(\boldsymbol{z}_{a_\ell,k}|[\boldsymbol{x}^{(p)}_{b_\ell,k}])U_{[\boldsymbol{s}^{(p)}_{k|k-1}]}(\boldsymbol{s}_k) = \frac{|[\boldsymbol{s}_k^{a_\ell,(p)}]|}{|[\boldsymbol{s}^{(p)}_{k|k-1}]|}U_{[\boldsymbol{s}_k^{a_\ell,(p)}]}(\boldsymbol{s}_k)p(\boldsymbol{z}_{a_\ell,k}|[\boldsymbol{x}^{(p)}_{b_\ell,k}]). \tag{4.26}$$

Note, the notation $|\cdot|$ for a box denotes the interval length (respectively the box volume in the multidimensional case), in contrast to the cardinality of a set. If the entire product is considered, $\prod_{\ell=1}^m p(\boldsymbol{z}_{a_\ell,k}|[\boldsymbol{x}^{(p)}_{b_\ell,k}])U_{[\boldsymbol{s}^{(p)}_{k|k-1}]}(\boldsymbol{s}_k)$, the contracted box, $U_{[\boldsymbol{s}_k^{a_1,(p)}]}(\boldsymbol{s}_k)$, is further contracted in the same manner as described by equations (4.25) and (4.26) by each measurement likelihood for a single object. In terms of interval arithmetic, the product of contracted boxes is equivalent to the intersection of the box particles contracted by the individual measurements,

$$\prod_{\ell=1}^{m} p(\boldsymbol{z}_{a_\ell,k}|[\boldsymbol{x}^{(p)}_{b_\ell,k}])U_{[\boldsymbol{s}^{(p)}_{k|k-1}]}(\boldsymbol{s}_k) = \frac{|[\boldsymbol{s}_k^{a,(p)}]|}{|[\boldsymbol{s}^{(p)}_{k|k-1}]|}U_{[\boldsymbol{s}_k^{a,(p)}]}(\boldsymbol{s}_k)\prod_{\ell=1}^{m} p(\boldsymbol{z}_{a_\ell,k}|[\boldsymbol{x}^{(p)}_{b_\ell,k}]), \tag{4.27}$$

where $[\boldsymbol{s}_k^{a,(p)}] = \bigcap_{\ell=1}^{m} [\boldsymbol{s}_k^{a_\ell,(p)}]$. This results in the following reduced form of the posterior state pdf in (4.23),

$$p(\boldsymbol{s}_k|\boldsymbol{Z}_{1:k}) = \sum_{p=1}^{N} \frac{w_{k-1}^{(p)} e^{-\sum_{i\in\mathcal{J}(p)} \lambda_{T,i}}}{\alpha_k M_k!} \left(\rho^{M_k} U_{[\boldsymbol{s}_{k|k-1}^{(p)}]}(\boldsymbol{s}_k) + \sum_{m=1}^{M_k} \sum_{j=1}^{\binom{M_k}{m}} \sum_{n=1}^{|\mathcal{J}(p)|^m} \frac{|[\boldsymbol{s}_k^{\boldsymbol{a}_{m,j},(p,n)}]|\rho^{M_k-m}}{|[\boldsymbol{s}_{k|k-1}^{(p)}]|} \times \right.$$
$$\left. U_{[\boldsymbol{s}_k^{\boldsymbol{a}_{m,j},(p,n)}]}(\boldsymbol{s}_k) \prod_{\ell=1}^{m} \lambda_{T,(\boldsymbol{b}_{m,n})_\ell} p(\boldsymbol{z}_{(\boldsymbol{a}_{m,j})_\ell,k} | [\boldsymbol{x}_{(\boldsymbol{b}_{m,n})_\ell,k}^{(p)}]) \right). \tag{4.28}$$

This is referred to as a brute force approach since every possible measurement association is considered. It is clear from the indices of the summations that a single predicted box particle can result in a summation of a large number of terms. For example, in the case of 3 objects and 15 measurements, each predicted box particle would result in over 1 billion weighted boxes after the update. Thus, the brute force implementation is not computationally tractable.

Standard Approach: There are two causes for why such a considerable number of boxes exists. The first cause is the uncertainty in which measurements are from which objects. This uncertainty can be reduced through the introduction of clustering. The clustering algorithm assigns the index of each measurement to a single cluster set $\mathcal{C}_i$, where $i \in 1, .., N_c$, with N_c the total number of clusters, assumed unknown. Measurements which are close to each other, according to a specific metric, are assigned to the same cluster. The validity of utilising clustering is based on the assumption that measurements from a single object are typically located within the vicinity of each other in the measurement space. However, care is taken to ensure that the algorithm is robust to sub-optimal clustering. Considering clustering, results in the following approximation of the posterior state pdf:

$$p(\boldsymbol{s}_k|\boldsymbol{Z}_{1:k}) \approx \sum_{p=1}^{N} \frac{w_{k-1}^{(p)} e^{-\sum_{i\in\mathcal{J}(p)} \lambda_{T,i}}}{\alpha_k M_k!} \left(\rho^{M_k} U_{[\boldsymbol{s}_{k|k-1}^{(p)}]}(\boldsymbol{s}_k) + \sum_{m=1}^{M_k} \sum_{j=1}^{\binom{M_k}{m}} \sum_{n=1}^{|\mathcal{J}(p)|^{d_j}} \frac{|[\boldsymbol{s}_k^{\boldsymbol{a}_{m,j},(p,n)}]|\rho^{M_k-m}}{|[\boldsymbol{s}_{k|k-1}^{(p)}]|} \times \right.$$
$$\left. U_{[\boldsymbol{s}_k^{\boldsymbol{a}_{m,j},(p,n)}]}(\boldsymbol{s}_k) \prod_{\ell=1}^{m} \lambda_{T,(\boldsymbol{b}_{m,n})_\ell} p(\boldsymbol{z}_{(\boldsymbol{a}_{m,j})_\ell,k} | [\boldsymbol{x}_{(\boldsymbol{b}_{m,n})_\ell,k}^{(p)}]) \right), \tag{4.29}$$

where d_j is the number of clusters that the jth unique combination of object assigned measurements originates from, and the sequences $(\boldsymbol{b}_{m,n})$ are reduced to only consider the measurements to object associations where measurements from the same cluster are assigned to the same object. Considering the same example of 3 objects and 15 measurements, if the clustering algorithm results in 3 clusters, each indexing 5 of the measurements, the number of weighted box particles after the update per predicted box particle is reduced from over 1 billion to 830 584. Although this reduces the number of weighted boxes by orders of magnitude for each box particle, this still results in a large computational burden.

Interval Analysis Approach: The second cause for the large number of boxes is the uncertainty in which measurements are emitted by an object or are clutter. Using an interval based approach, it is possible to reduce the number of boxes due to this uncertainty.

The weight of each term in the posterior state pdf describes how likely the associations are, given the measurements. As observed in equation (4.27), each term is non-zero on the predicted state interval contracted by all the assigned object measurements. This interval is equivalent to the intersection of the contraction results for each of the measurements assigned as object originated. Each term can have clutter measurements assigned as object measurements. However, the contraction due to a clutter measurement can be an interval which does not exist, or is disjoint with the contracted intervals from the object originated measurements, as illustrated in Figure 4.2. Since the overall result is dependent on the intersection, even a single clutter measurement assigned as an object measurement may result in the corresponding term having a zero weight. The computation of these terms can be avoided by approximating the intersection with the relaxed intersection. The relaxed intersection, first introduced in [66], corresponds to the classical intersection between intervals with the exception that it is allowed to relax a certain number of intervals in order to avoid an empty intersection. Utilising the relaxed intersection, the following approximation

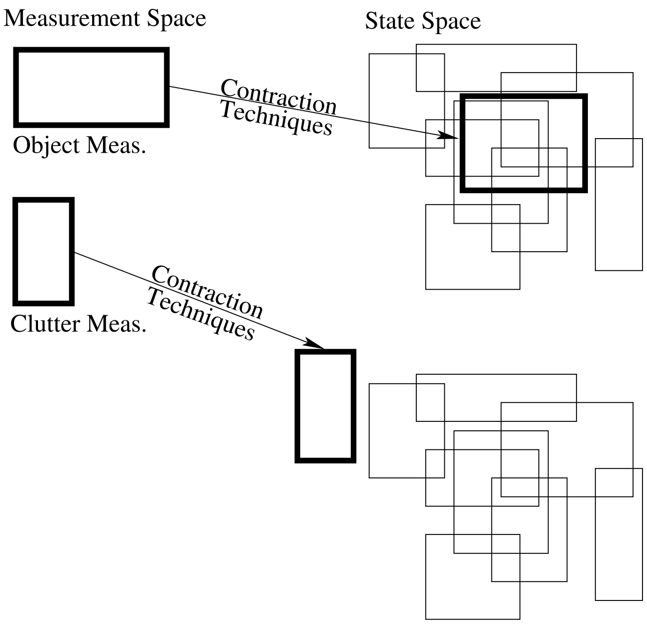


Figure 4.2: Illustration of the consistency between a set of box particles and object or clutter measurents.

for the posterior state pdf is obtained,

$$p(\boldsymbol{s}_k|\boldsymbol{Z}_{1:k})\approx\sum_{p=1}^{N}\frac{w_{k-1}^{(p)}e^{-\sum_{i\in\mathcal{J}(p)}\lambda_{T,i}}}{\alpha_k M_k!}\left(\sum_{n=1}^{|\mathcal{J}(p)|^d}\frac{|[\boldsymbol{s}_k^{\boldsymbol{a},(p,n)}]|\rho^{M_k-u}}{|[\boldsymbol{s}_{k|k-1}^{(p)}]|}\times\right.$$
$$\left.U_{[\boldsymbol{s}_k^{\boldsymbol{a},(p,n)}]}(\boldsymbol{s}_k)\prod_{\ell=1}^{u}\lambda_{T,(\boldsymbol{b}_n)_\ell}p(\boldsymbol{z}_{(\boldsymbol{a})_\ell,k}|[\boldsymbol{x}_{(\boldsymbol{b}_n)_\ell,k}^{(p)}])\right), \tag{4.30}$$

where u is the number of consistent intervals which results in a non-empty relaxed intersection. In order to determine the contracted state, $[\boldsymbol{s}_k^{\boldsymbol{a},(p,n)}]$, the sub-states of each object are considered individually, with u_i the corresponding number of consistent intervals for object sub-state i. The index for all the measurements assigned to object i, according to clustering, is defined by the set $\mathcal{B}$. Since only these measurements contract the sub-state of object i, the resulting contraction result for all measurements is given by $[\boldsymbol{x}_{i,k}^{\boldsymbol{a},(p,n)}] = \overset{\{|\mathcal{B}|-u_i\}}{\bigcap}_{\ell\in\mathcal{B}}[\boldsymbol{x}_{i,k}^{\ell,(p,n)}]$ with $u = \sum_{i\in\mathcal{J}(p)} u_i$.

Considering the same example of 3 objects and 15 measurements, if the clustering algorithm results in 3 clusters, each indexing 5 of the measurements, the number of weighted box particles after the update per predicted box particle is reduced from 830 584 with the approximate posterior in (4.29) to 27 with the approximate posterior in (4.30).

Two issues remain with the calculation of the approximate pdf in (4.30). Firstly, the relaxed intersection does not explicitly indicate the indices of the u measurements which result in the non-zero intersection, which means it is not possible to evaluate the corresponding measurement likelihood for a single object. Secondly, it is required to ensure that the box particle weight is represented by a single scalar value. If the measurement likelihood for a single object could be evaluated, this result may not be the case. There are several approaches which could be used to overcome this, such as selecting the midpoint of the box particle for evaluation of the measurement likelihood for a single object. However, the approach adopted here overcomes both the remaining issues by approximating the measurement likelihood for a single object with a uniform distribution, as done previously in Section 3.4,

$$p(\boldsymbol{z}_{j,k}|\boldsymbol{x}_{i,k}) \approx U_{r(\boldsymbol{x}_{i,k})}(\boldsymbol{z}_{j,k}). \tag{4.31}$$

This approximation is based on the fact that the magnitude of the uncertainty in the sensor is in general significantly smaller than the extent of the object. In summary, the posterior at the previous time step, $p(\boldsymbol{s}_{k-1}|\boldsymbol{Z}_{1:k-1})$, is approximated by $\{w_{k-1}^{(p)}, [\boldsymbol{s}_{k-1}^{(p)}]\}_{p=1}^{N}$, and the posterior at the current time step, $p(\boldsymbol{s}_k|\boldsymbol{Z}_{1:k})$ is approximated by $\{\{w_k^{(p,n)}, [\boldsymbol{s}_k^{(p,n)}]\}_{p=1}^{N}\}_{n=1}^{|\mathcal{J}(p)|^d}$, where $[\boldsymbol{s}_k^{(p,n)}] = [\boldsymbol{s}_k^{\boldsymbol{a},(p,n)}]$, and

$$w_k^{(p,n)} = \frac{w_{k-1}^{(p)} e^{-\sum_{i\in\mathcal{J}(p)} \lambda_{T,i}} |[\boldsymbol{s}_k^{\boldsymbol{a},(p,n)}]| \rho^{M_k-u}}{|[\boldsymbol{s}_{k|k-1}^{(p)}]|} \prod_{\ell\in\mathcal{J}(p)} \left(\frac{\lambda_{T,\ell}}{|[r(\boldsymbol{x}_{\ell,k})]|}\right)^{u_\ell}. \tag{4.32}$$

4.4.2 Box Particle Filter Resampling

The number of box particles representing the posterior state pdf grows randomly with each time step. To curb the increase in the number of box particles, a resampling step is introduced, where the number of resampled particles is equal to the original number of box particles. In addition, the resampling step also relieves particle degeneracy. The resampling step in the Box PF differs from the resampling step of the general PF. The resampling step in the Box PF can be performed by a division of box particles [53] (the box particle which has been selected n times during resampling can be partitioned into n disjoint smaller boxes) or by other techniques.

The algorithm of the Box PF for multiple extended object tracking is given in Algorithm 12.

4.5 Performance Evaluation

4.5.1 Testing Environment

The Box PF for multiple extended object tracking performance is evaluated using data obtained from the HAMLeT (Hazardous Material Localisation and Person Tracking) system environment [125] (Fraunhofer FKIE, Germany). The data is from a prototype security system developed by an EU funded project, representing an airport corridor. This data consists of range and bearing components obtained by three laser rangefinder devices. These devices have a scan interval of 0.5°, scan area of 360°, and provide data at a scan rate of 10 Hz. They are positioned at three key locations in a curved corridor (see Figure 4.3). The scenario presented in this section consists of three persons who enter and traverse the corridor while being observed by the sensors. Throughout their motion, each person moves in and out of the area visible by the sensors at different times. The sensors are positioned on the wall at the level of height of the hip. The high resolution of the measurement devices allowed for the manual visual extraction of ground truth for the centroid of the objects. This was required since no other sensors, e.g. global positioning system (GPS), were available for ground truth collection.

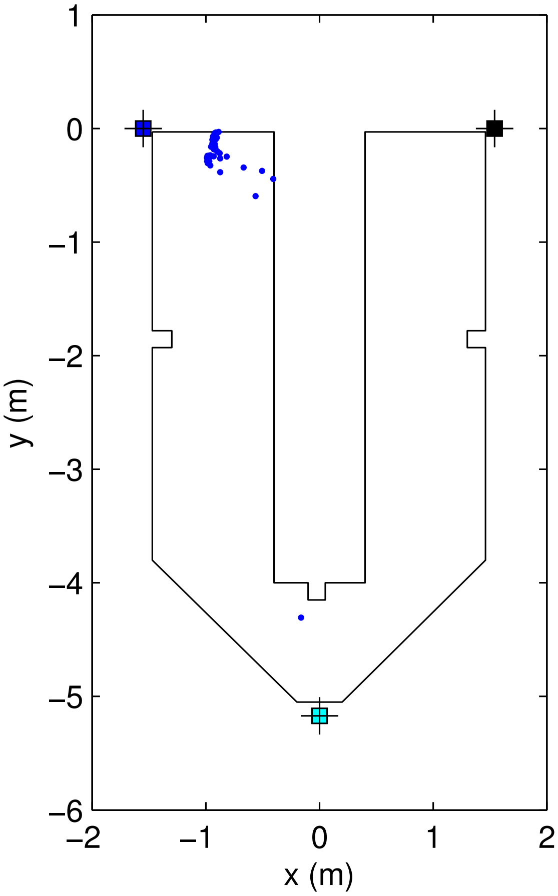


Figure 4.3: The layout of the corridor for the experiments. The three laser scanner devices are indicated with crossed boxes at the lower part of the graph. In this snapshot several measurements from the sensor located at the top left of the figure are displayed.

4.5.2 Performance Comparison

A comparison is made between the performance of the BP PF and Box PF over 100 MC runs. The measurements are perturbed by the measurement noise for each run. See Section 3.7.1 for details about the computing platform. The performance is evaluated based on the Optimal Sub-Pattern Assignment (OSPA) [104] for the position of the objects, cardinality for the existent variables, the statistics of the existent object extents, and the average simulation time. It is worth highlighting that in contrast to the extracted filter estimates in Chapter 3, here the particle with a maximum weight is selected as the filter estimate since particles can have a different number of existent objects being tracked.

4.5.3 Filter Parameters and initialisation

The Box PF utilises the DBSCAN algorithm [42] for clustering. This is a density based clustering algorithm which groups the measurements that are closely packed together into a single cluster. This clustering algorithm is well suited to the problem as it does not require knowledge of the number of clusters, and the density of the measurements from each object is consistent. DBSCAN requires two parameters, $\epsilon = 0.43$, related to the density of the clusters, and the minimum number of points required to form a dense region, which is selected as 1.

The other parameters used in simulation for the performance evaluation are as follow: $\sigma_x = 0.05m/s^2$, $\sigma_y = 0.05m/s^2$, $\sigma_R = 0.05m$, $\sigma_d = 0.025$, $\sigma_\beta = 0.1\pi/180$, $T_s = 1\,\mathrm{s}$, $\lambda_T = 50$, $\rho = 1 \times 10^{-4}, P_e = 0.9, F = 30$.

The filters utilise a uniform distribution to initialise each object sub-state when an object birth occurs. In the case of the Box PF, the same uniform region where the BP PF randomly generates particles from is subdivided so that the entire region is encompassed by all the box particles. This region, for each object sub-state, is located at the entrance/exit of the corridor: $x_c = [-1.5; -0.5] \cup [0.5; 1.5]$ m, $\dot{x}_c = [-0.1; 0.1]$ m/s, $y_c = [-1; 0]$ m, $\dot{y}_c = [-0.1; 0.1]$ m/s, $R = [0; 0.3]$ m, see Figure 4.3 as reference.

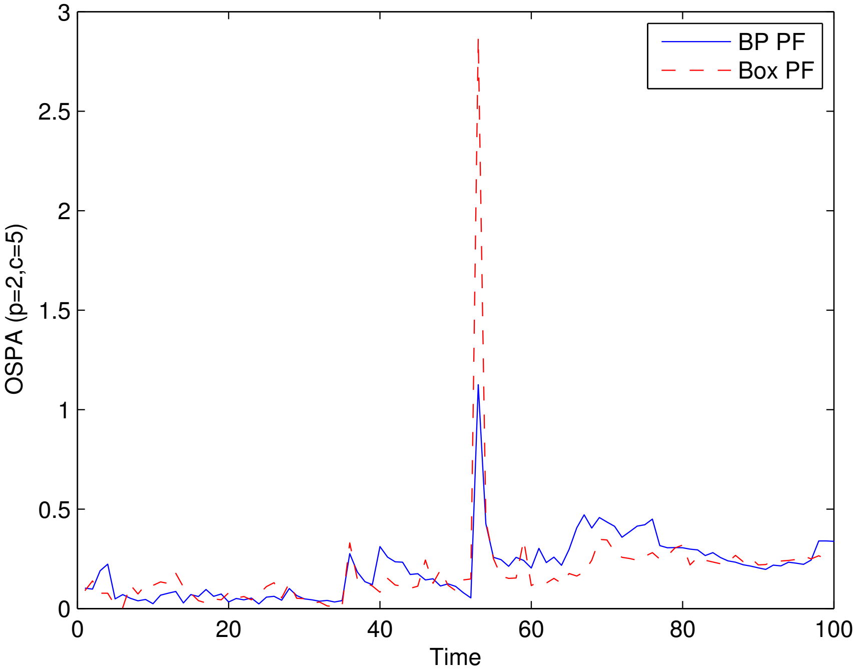


Figure 4.4: Comparison of the average OSPA for the BP PF with 5000 particles and the Box PF with 32 particles.

4.5.4 Results

The performance of the filters is examined for 3 cases: a small, medium, and large number of particles. The average OSPA results for each case are illustrated in Figures 4.4 to 4.6. The spikes in the results correspond to a mismatch in cardinality. This is caused by the fact that only a small number of measurements are observed from the objects when they first enter the observable region of a sensor. As expected, decreasing the number of particles increases the amount of error, however, it is worth noting that a decrease in the number of particles for the BP PF also causes the filter to become unstable when three objects are within the scene.

The average cardinality results for each case are illustrated in Figures 4.7 to 4.9. The cardinality of the Box PF is significantly more robust to different numbers of box particles.

The performance of the filters for the estimation of the extent parameter is il-

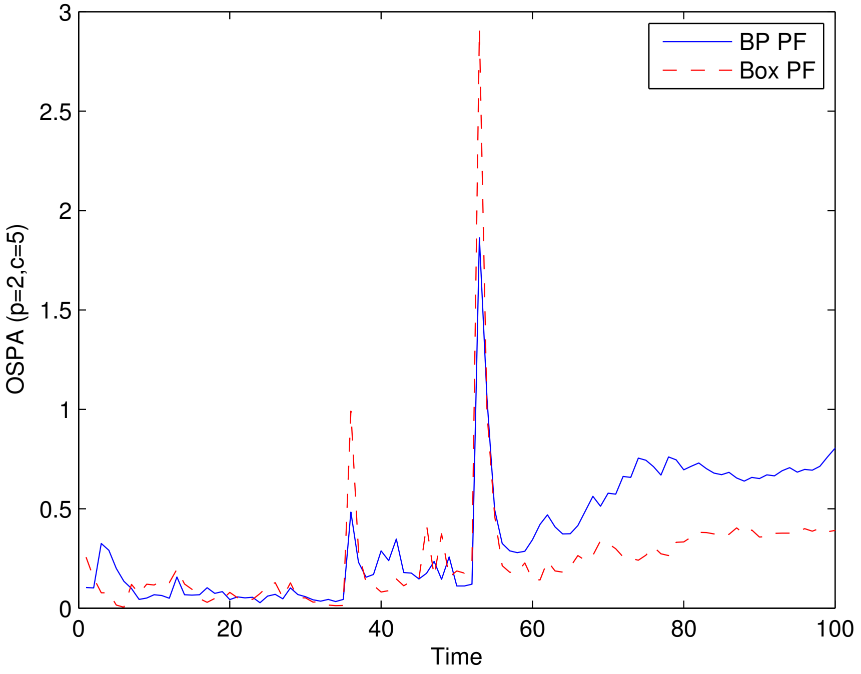


Figure 4.5: Comparison of the average OSPA for the BP PF with 2500 particles and the Box PF with 16 particles.

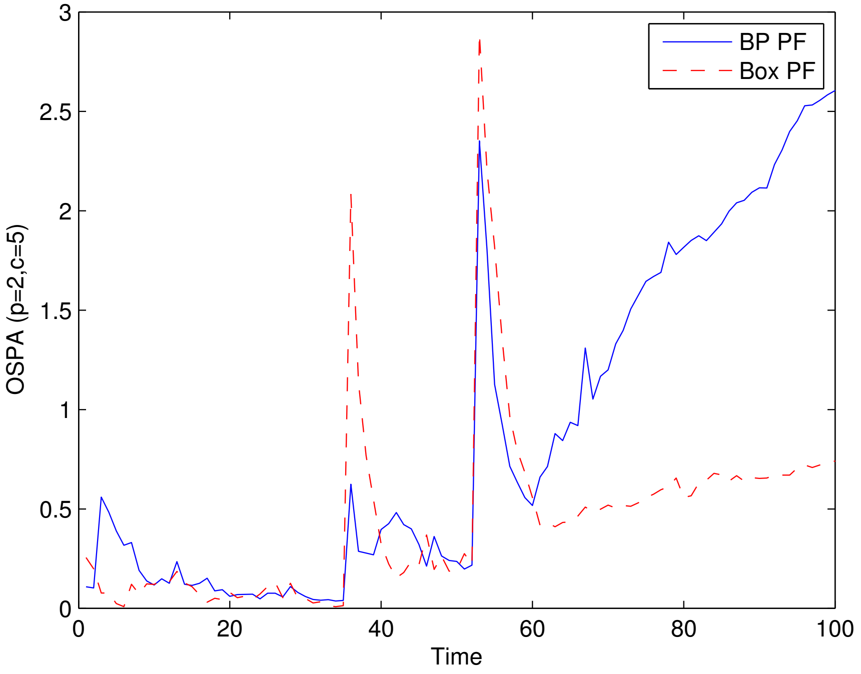


Figure 4.6: Comparison of the average OSPA for the BP PF with 1000 particles and the Box PF with 4 particles.

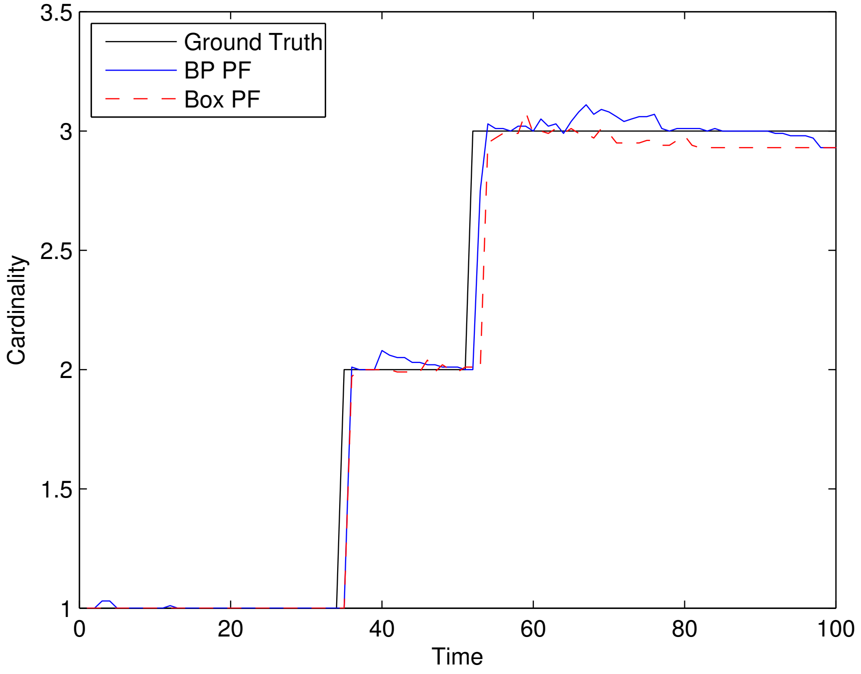


Figure 4.7: Comparison of the average cardinality for the BP PF with 5000 particles and the Box PF with 32 particles.

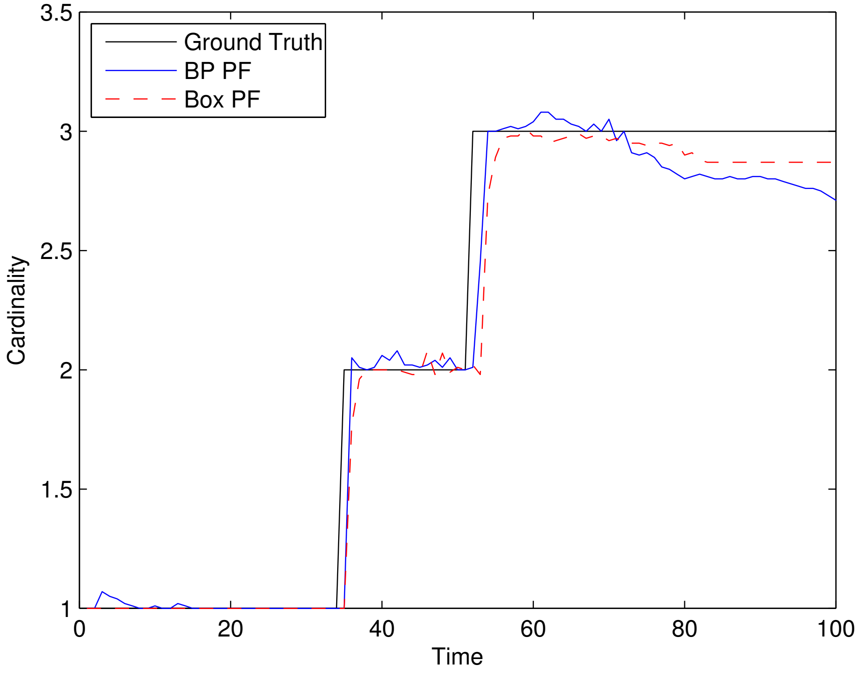


Figure 4.8: Comparison of the average cardinality for the BP PF with 2500 particles and the Box PF with 16 particles.

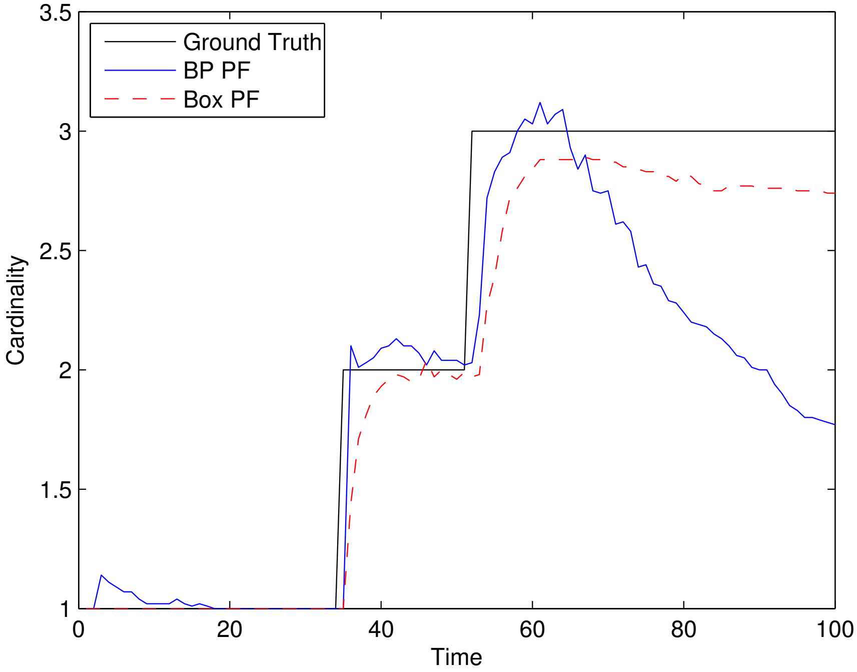


Figure 4.9: Comparison of the average cardinality for the BP PF with 1000 particles and the Box PF with 4 particles.

lustrated by the mean and standard deviation of the extent parameter for all active objects over all time steps, this is illustrated in Table 4.1. The contraction operation leads to the Box PF sustaining significantly lower variations in the extent, even with higher numbers of particles.

The computational time for each of the considered cases and both filters is given in Table 5.2. It is noted here that employing the INTLAB [101] toolbox with MATLAB for performing the Box PF simulation is just one way of implementing the Box PF code. This toolbox was initially designed and optimised for estimating rounding errors. Faster realisations of the Box PF in C/C++ are also possible. For instance, in [103] the Box Probability Hypothesis Density Filter is shown to be 10.9 times faster than the Probability Hypothesis Density Filter working with point particles (both implemented in C++). Further optimisation is considered possible for the Box PF realisation, thus the results in Table 5.2 would represent a minimum efficiency improvement. The Box PF is also a very attractive solution from the perspective of

Table 4.1: Existent object extent statistics.

Algorithm	N	Mean (m)	Standard Deviation (m)
Box PF	4	0.23	0.03
	16	0.21	0.04
	32	0.2	0.05
BP PF	1000	0.24	0.13
	2500	0.23	0.13
	5000	0.21	0.11

Table 4.2: Average MATLAB computational time comparison.

Algorithm	N	Computation Time (s)
Box PF	4	43.38
	16	118.56
	32	282.24
BP PF	1000	67.68
	2500	168.53
	5000	417.66

distributed estimation, as shown in [61].

An attractive benefit of the Box PF, not clearly illustrated in the results presented thus far, is the ability of the filter to handle large regions of initial uncertainty. For example, the prior distribution on the sub-states related to the velocity components of each object is a uniform distribution with the following region of support: $[-0.1, 0.1]$. This region caters for objects moving in any direction and was sufficient for the objects in the examined scenario, but when the magnitude is increased, the BP PF is unable to lock on to new born objects. This is due to the fact that the velocity of the object is not directly observed, causing the filter to diverge. However, due to contraction and the division of boxes in the resampling step, the Box PF is capable of handling larger regions of uncertainty. As an example, increasing the region to $[-1, 1]$, caused the BP PF to diverge in all three cases, where the Box PF performance was unaffected. This issue can be resolved by the BP PF by utilising a larger number of particles, but this comes at the cost of a greater computational complexity.

4.6 Summary

In this chapter, a Box PF method for multiple extended object tracking was presented. The extended objects are represented by a joint state vector which consists of the kinematics of the centroid of the objects, and parameters that describe a shape used to approximate the extent of the objects. In addition, an existence variable was introduced to cater for the appearance and disappearance of objects.

A theoretical derivation of the generalised likelihood function of the Box PF was presented. The proved equation is further modified to minimise the computational complexity.

The performance of the Box PF for multiple extended object tracking was evaluated with real data from laser rangefinder sensors. Three sensors were used to monitor people walking through a corridor. The extent of each person was modelled with a circle. The results have shown that the Box PF can work efficiently with four to thirty two box particles, whereas the PF working with point particles needs several thousands of particles to achieve the same accuracy. The Box PF has been shown to have several advantages when compared to the BP PF. This includes a significant computational gain, more than 32%, which could potentially be further exploited through an implementation on a platform that is efficient in interval arithmetic. The Box PF exhibits robustness for a significantly smaller number of box particles which completely encompass the initialisation region.

Algorithm 11 CP algorithm for Circular Extended Objects

```
Input: $[\boldsymbol{x}_{i,k}^{(p)}]$,$[\boldsymbol{x}_{i,k-1}^{(p)}]$, $[\boldsymbol{z}_{m,k}]$.
Let $[\boldsymbol{x}_{i,k}^{m,(p)}] = [\boldsymbol{x}_{i,k}^{(p)}]$, and DONE = FALSE.
while DONE == FALSE do
    Transform the range and bearing measurements into the $x$-$y$ plane using an inclusion function:
        $[z_1] = [C_{z_1}]\left([d_k^m],[\beta_k^m]\right),$
        $[z_2] = [C_{z_2}]\left([d_k^m],[\beta_k^m]\right),$   (4.33)
    Contract the intervals with each constraint:
        $\left[x_{i,k}^{m,(p)}\right] = \left[x_{i,k}^{m,(p)}\right] \cap \left([z_1] \pm \sqrt{\left[R_{i,k}^{(p)}\right]^2 - \left([z_2] - \left[y_{i,k}^{m,(p)}\right]\right)^2}\right),$   (4.34)
        $\left[\dot{x}_{i,k}^{m,(p)}\right] = \left[\dot{x}_{i,k}^{m,(p)}\right] \cap \left(\frac{\left[x_{i,k}^{m,(p)}\right] - \left[x_{i,k-1}^{(p)}\right]}{T_s}\right),$
        $\left[y_{i,k}^{m,(p)}\right] = \left[y_{i,k}^{m,(p)}\right] \cap \left([z_2] \pm \sqrt{\left[R_{i,k}^{(p)}\right]^2 - \left([z_1] - \left[x_{i,k}^{m,(p)}\right]\right)^2}\right),$
        $\left[\dot{y}_{i,k}^{m,(p)}\right] = \left[\dot{y}_{i,k}^{m,(p)}\right] \cap \left(\frac{\left[y_{i,k}^{m,(p)}\right] - \left[y_{i,k-1}^{(p)}\right]}{T_s}\right),$
        $\left[R_{i,k}^{m,(p)}\right] = \left[R_{i,k}^{m,(p)}\right] \cap \left(\sqrt{\left([z_1] - \left[x_{i,k}^{m,(p)}\right]\right)^2 + \left([z_2] - \left[y_{i,k}^{m,(p)}\right]\right)^2}\right),$
        $[z_1] = [z_1] \cap \left(\left[x_{i,k}^{m,(p)}\right] \pm \sqrt{\left[R_{i,k}^{m,(p)}\right]^2 - \left([z_2] - \left[y_{i,k}^{m,(p)}\right]\right)^2}\right),$
        $[z_2] = [z_2] \cap \left(\left[x_{i,k}^{m,(p)}\right] \pm \sqrt{\left[R_{i,k}^{m,(p)}\right]^2 - \left([z_1] - \left[x_{i,k}^{m,(p)}\right]\right)^2}\right).$
    Contract the original measurements with the contracted converted measurements:
        $[d_k^m] = [d_k^m] \cap \left[C_{z_1}^{-1}\right]\left([z_1],[z_2]\right),$
        $[\beta_k^m] = [\beta_k^m] \cap \left[C_{z_2}^{-1}\right]\left([z_1],[z_2]\right),$   (4.35)
    If convergence criterion is met (e.g. difference in the size of pre-contracted and contracted boxes is below a threshold), DONE = TRUE.
end while
Output: $[\boldsymbol{x}_{i,k}^{m,(p)}]$.
```

Algorithm 12 A Box Particle Filter for Multiple Extended Object Tracking

```
***Initialisation***
Initialise the set of box particles, $\{[\boldsymbol{x}_0^{(p)}]\}_{p=1}^{N}$, sampling from the prior distribution.

for $k$ = 1,...,$T$ do
    ***Prediction***
    Generate the predicted box particles, , $\{[\boldsymbol{x}_{k|k-1}^{(p)}]\}_{p=1}^{N}$, by propagating the box particles through the state evolution model and applying interval inclusion functions as described in [67, 65].
    ***Measurement Update***
    Upon the receipt of new measurements:
    Convert the measurements into measurement boxes, $[\boldsymbol{Z}_k]$.
    Cluster the measurements to obtain the set $\mathcal{C}_i$, where $i \in 1, .., N_c$.
    Solve the CSP in (4.25) using the CP algorithm (i.e. Algorithm 11), to obtain the contracted box particles for each measurement $[\boldsymbol{s}_k^{a_\ell,(p,n)}]$.
    Determine the combined contracted box particle, $[\boldsymbol{s}_k^{\boldsymbol{a},(p,n)}]$, and the number of consistent intervals, $u$, through the calculation of the relaxed intersection.
    Generate the set of weighted box particles according to (4.32).
    ***Output***
    Obtain a box estimate for the state of the extended objects based on the maximum weight:
        $$[\widehat{\boldsymbol{x}}_k] = \arg\max_{\boldsymbol{x}_k} w_k^{(p)} \qquad (4.36)$$
    and a point estimate $\widehat{\boldsymbol{x}}_k$ for the extended shape using the mid-points of the box estimates of the state vector $[\widehat{\boldsymbol{x}}_k]$.
    ***Resampling*** Resample $N$ particles with high weights by division. Finally, reset the weights: $w_k^{(p)} = 1/N$.
end for
```

Chapter 5

OBJECT TRACKING WITH TALL DATA

In Chapters 3 and 4, object tracking scenarios with modelling complexities introduced by receiving multiple measurements at each time step were considered. An additional complexity associated with receiving multiple measurements at each time step is an increase in the computational load. This computational expense increases as the number of measurements grows. In this chapter, the focus is shifted to the problem of efficiently processing excessively large amounts of data for object tracking applications, which would other wise hinder the algorithms from being viable solutions. Typically in these scenarios measurement reduction techniques are utilised, e.g. gating [10], as a mechanism that attempts to remove uninformative measurements. However, these techniques may be limited in applications where sensors are capable of returning a large number of informative measurements, e.g. a 3D LiDAR sensor can return 2.2 million measurements per second [117]. Section 5.1 begins with an overview of recent advances made in SMCMC for object tracking. In Section 5.2 the concept of adaptive subsampling is presented and incorporated into the SMCMC framework. In contrast to subsampling the large amounts of data, Section 5.3 and 5.4 present a divide and conquer approach where batches of measurements are processed in parallel. This is based on the incorporation of the EP algorithm within an SMCMC and PF framework. Finally, Section 5.5 illustrates the performance of the proposed methods through a number of simulated examples.

5.1 *Advances in Sequential Markov chain Monte Carlo for Object Tracking*

The SMCMC filter described by Algorithm 5 was shown to work well in state space models containing a high number of dimensions when compared to techniques relying on importance sampling, however, this direct approach results in a high computational expense. It was proposed in [108] to consider targeting the joint filtering posterior state pdf of $\boldsymbol{x}_k$ and $\boldsymbol{x}_{k-1}$,

$$p(\boldsymbol{x}_k, \boldsymbol{x}_{k-1}|\boldsymbol{z}_{1:k}) \propto p(\boldsymbol{z}_k|\boldsymbol{x}_k)p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})p(\boldsymbol{x}_{k-1}|\boldsymbol{z}_{1:k-1}), \tag{5.1}$$

as the equilibrium distribution in order to help alleviate the high computational demand. In a similar fashion, an approximation for the joint filtering posterior state pdf can be obtained through MCMC methods by representing $p(\boldsymbol{x}_{k-1}|\boldsymbol{z}_{1:k-1})$ with a set of unweighted particles. This approach has the advantage of avoiding the direct MC computation of the predictive posterior density. Furthermore, the approximation can be trivially marginalised to obtain the filtering posterior state pdf of interest.

More specifically, at each time step, samples from the joint filtering posterior state pdf are obtained by the MH algorithm, referred to as a joint draw, as both $\boldsymbol{x}_k$ and $\boldsymbol{x}_{k-1}$ are sampled simultaneously. However, sampling from a higher dimensional space decreases the likelihood of acceptance in the MH algorithm. Therefore, this is followed by Gibbs sampling of the marginals of the joint filtering posterior state pdf. By decomposing the state vector with N_d dimensions into P disjoint sub-states, $\{\boldsymbol{x}_{\Omega_p,k}\}_{p=1}^{P}$, such that $\bigcup_p \Omega_p = \{1, ..., N_d\}$ and $\Omega_p \cap \Omega_q = \emptyset, \forall p \neq q$, then the corresponding marginals include $\{p(\boldsymbol{x}_{\Omega_q,k}|\boldsymbol{z}_{1:k})\}_{p=1}^{P}$ and $p(\boldsymbol{x}_{k-1}|\boldsymbol{z}_{1:k})$. Since the expressions of these marginals are unknown and cannot be sampled from, the implementation consists of a MH sampler embedded within the Gibbs sampler, also known as Metropolis-within-Gibbs. This is referred to as the refinement step and aids in the mixing of the chain. An appropriate burn in period, N_b, is introduced to minimise the effect of the initial values of the Markov chain. This approach is highlighted by Algorithm 13 and is referred to as standard SMCMC.

Algorithm 13 Standard Sequential Markov Chain Monte Carlo

```
Initialise particle set: $\{\boldsymbol{x}_0^{(j)}\}_{j=1}^N$
for $k = 1,...,T$ do
    for $m = 1,...,N + N_b$ do
        Joint Draw
        Propose $\{\boldsymbol{x}_k^*, \boldsymbol{x}_{k-1}^*\} \sim q_1\left(\boldsymbol{x}_k, \boldsymbol{x}_{k-1} | \boldsymbol{x}_k^{m-1}, \boldsymbol{x}_{k-1}^{m-1}\right)$
        Compute the MH acceptance probability $\rho_1 =$
        $\min\left(1, \frac{p(\boldsymbol{x}_k^*, \boldsymbol{x}_{k-1}^* | \boldsymbol{z}_{1:k})}{q_1\left(\boldsymbol{x}_k^*, \boldsymbol{x}_{k-1}^* | \boldsymbol{x}_k^{m-1}, \boldsymbol{x}_{k-1}^{m-1}\right)} \frac{q_1\left(\boldsymbol{x}_k^{m-1}, \boldsymbol{x}_{k-1}^{m-1} | \boldsymbol{x}_k^*, \boldsymbol{x}_{k-1}^*\right)}{p(\boldsymbol{x}_k^{m-1}, \boldsymbol{x}_{k-1}^{m-1} | \boldsymbol{z}_{1:k})}\right)$
        Accept $\{\boldsymbol{x}_k^m, \boldsymbol{x}_{k-1}^m\} = \{\boldsymbol{x}_k^*, \boldsymbol{x}_{k-1}^*\}$ with probability $\rho_1$
        Refinement
        Propose $\{\boldsymbol{x}_{k-1}^*\} \sim q_2\left(\boldsymbol{x}_{k-1} | \boldsymbol{x}_k^m, \boldsymbol{x}_{k-1}^m\right)$
        Compute the MH acceptance probability $\rho_2 =$
        $\min\left(1, \frac{p(\boldsymbol{x}_{k-1}^* | \boldsymbol{x}_k^m, \boldsymbol{z}_{1:k})}{q_2\left(\boldsymbol{x}_{k-1}^* | \boldsymbol{x}_k^m, \boldsymbol{x}_{k-1}^m\right)} \frac{q_2\left(\boldsymbol{x}_{k-1}^m | \boldsymbol{x}_k^m, \boldsymbol{x}_{k-1}^*\right)}{p(\boldsymbol{x}_k^m | \boldsymbol{x}_{k-1}^m, \boldsymbol{z}_{1:k})}\right)$
        Accept $\boldsymbol{x}_{k-1}^m = \boldsymbol{x}_{k-1}^*$ with probability $\rho_2$
        Divide $\boldsymbol{x}_k$ into $P$ disjoint blocks $\{\Omega_p\}_{p=1}^P$ such that
        $\bigcup_p \Omega_p = \{1, ..., N_d\}$ and $\Omega_p \cap \Omega_q = \emptyset, \forall p \neq q$
        for $p = 1,...,P$ do
            Propose $\{\boldsymbol{x}_{\Omega_p,k}^*\} \sim q_{3,p}\left(\boldsymbol{x}_{\Omega_p,k} | \boldsymbol{x}_k^m, \boldsymbol{x}_{k-1}^m\right)$
            Compute the MH acceptance probability $\rho_{3,p} =$
            $\min\left(1, \frac{p(\boldsymbol{x}_{\Omega_p,k}^* | \boldsymbol{x}_{k-1}^m, \boldsymbol{z}_{1:k})}{q_{3,p}\left(\boldsymbol{\Omega_p}, \boldsymbol{x}_k^* | \boldsymbol{x}_k^m, \boldsymbol{x}_{k-1}^m\right)} \frac{q_{3,p}\left(\boldsymbol{x}_{\Omega_p,k}^m | \boldsymbol{x}_k^*, \boldsymbol{x}_{k-1}^m\right)}{p(\boldsymbol{x}_{\Omega_p,k}^m | \boldsymbol{x}_{k-1}^m, \boldsymbol{z}_{1:k})}\right)$
            Accept $\boldsymbol{x}_{\Omega_p,k}^m = \boldsymbol{x}_{\Omega_p,k}^*$ with probability $\rho_{3,p}$
        end for
    end for
    Approximation of the marginal posterior distribution with the following empirical
    measure:
    $\widehat{p}(\boldsymbol{x}_k | \boldsymbol{z}_{1:k}) = \frac{1}{N} \sum_{j=N_b+1}^{N+N_b} \delta(\boldsymbol{x}_k - \boldsymbol{x}_k^{(j)})$
end for
```

It is worth noting that there is a considerable amount of flexibility in the structure of SMCMC based methods. This has resulted in an abundance of related works, such as the utilisation of multiple MCMC chains, that interact with genetic algorithm inspired operators [106, 24]. However, the focus of these methods have been on increasing the MCMC convergence rate, and still requires the evaluation of all the measurements.

5.2 *Adaptive Subsampling Sequential Markov chain Monte Carlo*

In the standard SMCMC algorithm, calculating the acceptance probabilities, ρ_1 and $\rho_{3,p}$, requires the evaluation of all the measurements. In this section the concept of adaptive subsampling and SMCMC are merged to reduce this computational burden.

Looking back at the standard MH sampler in Algorithm 3, the expression in (2.15) can be further developed by applying Bayes' rule and assuming that there are M conditionally independent measurements, $\boldsymbol{z}_i$:

$$u < \frac{p(\boldsymbol{x}^*)q(\boldsymbol{x}^{m-1}|\boldsymbol{x}^*)}{p(\boldsymbol{x}^{m-1})q(\boldsymbol{x}^*|\boldsymbol{x}^{m-1})} \prod_{i=1}^{M} \frac{p(\boldsymbol{z}_i|\boldsymbol{x}^*)}{p(\boldsymbol{z}_i|\boldsymbol{x}^{m-1})}. \tag{5.2}$$

The previous state of the chain is stored as the current state, $\boldsymbol{x}^m = \boldsymbol{x}^{m-1}$, when the proposed sample does not meet this criterion. Further manipulating this expression into a form with the likelihoods isolated results in:

$$\begin{gathered} \frac{1}{M}\log\left[u\frac{p(\boldsymbol{x}^{m-1})q(\boldsymbol{x}^*|\boldsymbol{x}^{m-1})}{p(\boldsymbol{x}^*)q(\boldsymbol{x}^{m-1}|\boldsymbol{x}^*)}\right] < \frac{1}{M}\sum_{i=1}^{M}\log\left[\frac{p(\boldsymbol{z}_i|\boldsymbol{x}^*)}{p(\boldsymbol{z}_i|\boldsymbol{x}^{m-1})}\right], \\ \psi(\boldsymbol{x}^{m-1},\boldsymbol{x}^*) < \Lambda^M(\boldsymbol{x}^{m-1},\boldsymbol{x}^*). \end{gathered} \tag{5.3}$$

When the number of measurements is very large, the log likelihood ratio becomes the most computationally expensive part of the standard SMCMC algorithm. To reduce the computational complexity, a MC approximation for the log likelihood ratio has been proposed [14]:

$$\Lambda^{S_m}(\boldsymbol{x}^{m-1},\boldsymbol{x}^*) = \frac{1}{S_m}\sum_{i=1}^{S_m}\log\left[\frac{p(\boldsymbol{z}_{i,*}|\boldsymbol{x}^*)}{p(\boldsymbol{z}_{i,*}|\boldsymbol{x}^{m-1})}\right] \tag{5.4}$$

where the set $\boldsymbol{z}_* = \{\boldsymbol{z}_{1,*}, ..., \boldsymbol{z}_{S_m,*}\}$ is drawn uniformly without replacement from the original set of M measurements.

The difficulty which arises is in selecting a minimum value for S_m that results in a set of subsampled measurements that contain enough information to make the correct decision in the MH sampler. To overcome this difficulty in standard MCMC for static inference, the authors in [12] proposed to use concentration inequalities which provide a probabilistic bound on how functions of independent random variables deviate from their expectation. In this case, the independent random variables are the log likelihood ratio terms. Thus, it is possible to obtain a bound on the deviation of the MC approximation in (5.4) from the complete log likelihood ratio:

$$P(|\Lambda^{S_m}(\boldsymbol{x}^{m-1},\boldsymbol{x}^*) - \Lambda^M(\boldsymbol{x}^{m-1},\boldsymbol{x}^*)| \leq c_{S_m}) \geq 1-\delta_{S_m} \tag{5.5}$$

where $\delta_{S_m} > 0$, and c_{S_m} is dependent on which inequality is used. There are several inequalities which could be used, including the empirical Bernstein inequality [9, 14], which results in:

$$c_{S_m} = \sqrt{\frac{2V_{S_m}\log(3/\delta_{S_m})}{S_m}} + \frac{3R\log(3/\delta_{S_m})}{S_m} \tag{5.6}$$

where V_{S_m} represents the sample variance of the log likelihood ratio, and R is the range given by

$$R = \max_{1\leq i\leq M}\left\{\log\left[\frac{p(\boldsymbol{z}_i|\boldsymbol{x}^*)}{p(\boldsymbol{z}_i|\boldsymbol{x}^{m-1})}\right]\right\} - \min_{1\leq i\leq M}\left\{\log\left[\frac{p(\boldsymbol{z}_i|\boldsymbol{x}^*)}{p(\boldsymbol{z}_i|\boldsymbol{x}^{m-1})}\right]\right\} \tag{5.7}$$

Looking back at the standard SMCMC approach, the joint draw is accepted based on the condition $\Lambda^M(\boldsymbol{x}^*,\boldsymbol{x}^{m-1}) > \psi(\boldsymbol{x}^*,\boldsymbol{x}^{m-1})$. It is required to relate this expression in terms of the MC approximation of (5.4). Since the MC approximation is bounded, it is not possible to make a decision when the value of $\psi(\boldsymbol{x}^*,\boldsymbol{x}^{m-1})$ falls within the region specified by the bound. Thus it is required that $|\Lambda^{S_m}(\boldsymbol{x}^{m-1},\boldsymbol{x}^*)-\psi(\boldsymbol{x}^*,\boldsymbol{x}^{m-1})| > c_{S_m}$, where $|\cdot|$ represents the absolute value, in order to be able to make a decision, with probability at least $1-\delta_{S_m}$.

This forms the underlying principle for the creation of a stopping rule [12, 88]. Let $\delta_s \in (0,1)$ be a user specified input parameter. The idea is to sequentially increase the size of S_m while at the same time checking if the stopping criterion, $|\Lambda^{S_m}(\boldsymbol{x}^{m-1},\boldsymbol{x}^*) - \psi(\boldsymbol{x}^*,\boldsymbol{x}^{m-1})| > c_{S_m}$, is met. If the stopping criterion is never met, then this will result in $S_m = M$, i.e. requiring the evaluation of all the measurements.

Selecting $\delta_{S_m} = \frac{p_s - 1}{p_s {S_m}^{p_s}} \delta_s$ results in $\sum_{S_m \geq 1} \delta_{S_m} \leq \delta_s$. The event

$$\mathcal{E} = \bigcap_{S_m \geq 1} \left\{ \left| \Lambda^{S_m}(\boldsymbol{x}^{m-1}, \boldsymbol{x}^*) - \Lambda^{M}(\boldsymbol{x}^{m-1}, \boldsymbol{x}^*) \right| \leq c_{S_m} \right\} \tag{5.8}$$

thus holds with probability at least $1 - \delta_s$ by a union bound argument.

This iterative procedure allows for an adaptive size of the number of measurements required to be evaluated. However, there is cause for concern with the definition of the stopping rule. That is the fact that the range, R, used in the calculation of (5.6), is dependent on the log likelihood for all M measurements. Calculating this range would thus inherently require at least the same number of calculations as in the standard SMCMC approach. In certain applications it may be possible to obtain an expression for the range which is independent of the measurements, however, this is not the general case. In order to overcome the computational complexity of the calculation of the range, and to reduce the sample variance V_{S_m} in the bound, a control variate has been introduced in [13], referred to as a proxy:

$$\wp_i(\boldsymbol{x}^{m-1}, \boldsymbol{x}^*) \approx \log \left[\frac{p(\boldsymbol{z}_i | \boldsymbol{x}^*)}{p(\boldsymbol{z}_i | \boldsymbol{x}^{m-1})} \right]. \tag{5.9}$$

Thus the MC approximation in (5.4) is augmented into

$$\Lambda_1^{S_m}(\boldsymbol{x}^{m-1}, \boldsymbol{x}^*) = \frac{1}{S_m} \sum_{i=1}^{S_m} \log \left[\frac{p(\boldsymbol{z}_{i,*} | \boldsymbol{x}^*)}{p(\boldsymbol{z}_{i,*} | \boldsymbol{x}^{m-1})} \right] - \wp_i(\boldsymbol{x}^{m-1}, \boldsymbol{x}^*).$$

It is required to amend the MH acceptance accordingly to take the inclusion of the proxy into account.

In [14], it was proposed to utilise a Taylor series as an approximation for the log likelihood, $\ell_i(\boldsymbol{x}) = \log p(\boldsymbol{z}_i | \boldsymbol{x})$. Since object tracking scenarios are typically time sensitive, a first order Taylor series is proposed to minimise computational cost,

$$\hat{\ell}_i(\boldsymbol{x}) = \ell_i(\boldsymbol{x}^+) + (\nabla \ell_i)^{\top}_{\boldsymbol{x}^+} \cdot (\boldsymbol{x} - \boldsymbol{x}^+), \tag{5.10}$$

where $(\nabla \ell_i)_{\boldsymbol{x}^+}$ represents the gradient of $\ell_i(\boldsymbol{x})$ evaluated at $\boldsymbol{x}^+$. This results in the following form of the proxy

$$\begin{aligned} \wp_i(\boldsymbol{x}^{m-1}, \boldsymbol{x}^*) &= \hat{\ell}_i(\boldsymbol{x}^*) - \hat{\ell}_i(\boldsymbol{x}^{m-1}), \\ &= (\nabla \ell_i)_{\boldsymbol{x}^+}^\top \cdot (\boldsymbol{x}^* - \boldsymbol{x}^{m-1}). \end{aligned} \tag{5.11}$$

With the inclusion of the proxy, the range, R, is now computed as,

$$\begin{aligned} R = &\max_{1 \le i \le M} \left\{ \log \left[\frac{p(\boldsymbol{z}_i | \boldsymbol{x}^*)}{p(\boldsymbol{z}_i | \boldsymbol{x}^{m-1})} \right] - \wp_i(\boldsymbol{x}^{m-1}, \boldsymbol{x}^*) \right\} \\ &- \min_{1 \le i \le M} \left\{ \log \left[\frac{p(\boldsymbol{z}_i | \boldsymbol{x}^*)}{p(\boldsymbol{z}_i | \boldsymbol{x}^{m-1})} \right] - \wp_i(\boldsymbol{x}^{m-1}, \boldsymbol{x}^*) \right\}. \end{aligned} \tag{5.12}$$

An upper bound for the range, R^B, can be derived, i.e. where $R^B \ \geq \ R$, which can be computed efficiently

$$\begin{aligned} R^B &= 2 \max_{1 \le i \le M} \left\{ \left| \log \left[\frac{p(\boldsymbol{z}_i | \boldsymbol{x}^*)}{p(\boldsymbol{z}_i | \boldsymbol{x}^{m-1})} \right] - \wp_i(\boldsymbol{x}^{m-1}, \boldsymbol{x}^*) \right| \right\} \\ &= 2 \max_{1 \le i \le M} \left\{ \left| \ell_i(\boldsymbol{x}^*) - \ell_i(\boldsymbol{x}^{m-1}) - \hat{\ell}_i(\boldsymbol{x}^*) + \hat{\ell}_i(\boldsymbol{x}^{m-1}) \right| \right\} \\ &= 2 \max_{1 \le i \le M} \left\{ \left| B(\boldsymbol{x}^*) - B(\boldsymbol{x}^{m-1}) \right| \right\} \end{aligned} \tag{5.13}$$

where $B(\boldsymbol{x}) = \ell_i(\boldsymbol{x}) - \hat{\ell}_i(\boldsymbol{x})$ is the remainder. The Taylor-Lagrange inequality states that if $|\nabla^2(\ell_i(\boldsymbol{x}))| \le \boldsymbol{Y}$, where $\nabla^2(\ell_i(\boldsymbol{x}))$ represents the Hessian of the log likelihood, on some interval $I = [a, b]$, then the remainder term, $B(\boldsymbol{x})$, can be upper bounded according to $|B(\boldsymbol{x})| \le \frac{\boldsymbol{Y}|\boldsymbol{x} - \boldsymbol{x}^+|^2}{2}$ on the same interval I. Finally, based on the triangle inequality, an upper bound on the range term is given by

$$\begin{aligned} R^B &= 2 \left| |B(\boldsymbol{x}^*)| + \left| B(\boldsymbol{x}^{m-1}) \right| \right|, \\ &= \left| \boldsymbol{Y} \left(\left| \boldsymbol{x}^* - \boldsymbol{x}^+ \right|^2 + \left| \boldsymbol{x}^{m-1} - \boldsymbol{x}^+ \right|^2 \right) \right|. \end{aligned} \tag{5.14}$$

The complete adaptive subsampling SMCMC approach is illustrated by Algorithms 14 and 15.

5.3 Expectation Propagation Sequential Markov Chain Monte Carlo

In the previous approach, reduction in computational complexity was based on Bayesian filtering with only a subset of all of the data. In contrast, the algorithm presented in this section utilises all of the data in a distributed way. The only way to achieve computational efficiency is to consider a divide and conquer based approach which processes subsets of the data in parallel. Firstly, the set of M_k measurements is divided into D subsets of measurements such that $\boldsymbol{z}_k = \bigcup_{d=1}^{D} \boldsymbol{z}_{d,k}$ and $\boldsymbol{z}_{i,k} \bigcap \boldsymbol{z}_{j,k} = \emptyset : i \neq j$. The joint filtering posterior state pdf in equation (5.1) is further factored,

$$p(\boldsymbol{x}_k, \boldsymbol{x}_{k-1}|\boldsymbol{z}_{1:k}) \propto p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) p(\boldsymbol{x}_{k-1}|\boldsymbol{z}_{1:k-1}) \prod_{d=1}^{D} p(\boldsymbol{z}_{d,k}|\boldsymbol{x}_k). \tag{5.15}$$

The D subsets of measurements are processed in parallel on D computing nodes. The challenge in divide and conquer based approaches is in combining the results from the computing nodes to obtain samples from the joint filtering posterior state pdf. A natural method of doing this is through the utilisation of concepts from EP[1]. EP is a variational message passing scheme [86], the EP framework allows for the incorporation of inference from all other $D-1$ computing nodes as a prior in the inference step for any given computing node. This is achieved by approximating the likelihood of the $D-1$ sets of measurements from the other computing nodes with a distribution from the exponential density family,

$$\pi(\boldsymbol{x}_k|\boldsymbol{\eta}) = h(\boldsymbol{x}) g(\boldsymbol{\eta}) \exp\left\{\boldsymbol{\eta}^\top \boldsymbol{u}(\boldsymbol{x})\right\}, \tag{5.16}$$

where $\boldsymbol{\eta}$ represents the natural parameters (NPs) and $\boldsymbol{u}(\boldsymbol{x})$ is a function which varies depending on the member of the exponential family. The local joint filtering posterior state pdf for computing node d is then given by:

$$p_d(\boldsymbol{x}_k, \boldsymbol{x}_{k-1}|\boldsymbol{z}_{1:k}) \propto p(\boldsymbol{z}_{d,k}|\boldsymbol{x}_k) p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) p(\boldsymbol{x}_{k-1}|\boldsymbol{z}_{1:k-1}) \prod_{i \neq d} \pi(\boldsymbol{x}_k|\boldsymbol{\eta}_i). \tag{5.17}$$

[1] Refer to Appendix D for a review of the EP algorithm.

Each local joint filtering posterior state pdf is an approximation of the joint filtering posterior state pdf in (5.15).

The algorithm proceeds iteratively, beginning with the application of MCMC to draw a batch of samples from (5.17) on each computing node. The NPs of each computing node, $\boldsymbol{\eta}_d$, are then determined. This is done by firstly considering the marginalised local filtering posterior state pdf,

$$p_d(\boldsymbol{x}_k|\boldsymbol{z}_{1:k}) \propto p(\boldsymbol{z}_{d,k}|\boldsymbol{x}_k)p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1}) \prod_{i \neq d} \pi(\boldsymbol{x}_k|\boldsymbol{\eta}_i). \tag{5.18}$$

A discrete approximation for the marginalised local filtering posterior state pdf can be cheaply obtained from the MCMC samples drawn from the local joint filtering posterior state pdf. Further, by replacing the likelihood expression with the approximate likelihood term:

$$\widehat{p}_d(\boldsymbol{x}_k|\boldsymbol{z}_{1:k}) \propto \pi(\boldsymbol{x}_k|\boldsymbol{\eta}_d)p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1}) \prod_{i \neq d} \pi(\boldsymbol{x}_k|\boldsymbol{\eta}_i). \tag{5.19}$$

The idea is to select the NPs, $\boldsymbol{\eta}_d$, in a manner which results in the minimisation of $\mathrm{KL}(p_d(\boldsymbol{x}_k|\boldsymbol{z}_{1:k})||\widehat{p}_d(\boldsymbol{x}_k|\boldsymbol{z}_{1:k}))$, where $\mathrm{KL}(\,\cdot\,)$ refers to the Kullback-Leibler divergence. It has been shown [21] that the minimisation occurs when:

$$\mathbb{E}_{p_d(\boldsymbol{x}_k|\boldsymbol{z}_{1:k})}\left[\boldsymbol{u}(\boldsymbol{x})\right] = \mathbb{E}_{\widehat{p}_d(\boldsymbol{x}_k|\boldsymbol{z}_{1:k})}\left[\boldsymbol{u}(\boldsymbol{x})\right], \tag{5.20}$$

where $\mathbb{E}\left[\,\cdot\,\right]$ represents the expectation, which corresponds to matching the expected sufficient statistics. Approximating the discrete distributions with the same exponential density family as the likelihood term approximation, i.e.. $\pi(\boldsymbol{x}_k|\boldsymbol{\eta}_{p,d}) \approx p_d(\boldsymbol{x}_k|\boldsymbol{z}_{1:k})$ and $\pi(\boldsymbol{x}_k|\boldsymbol{\eta}_{f,d}) \approx p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1})$, results in the NPs being determined by:

$$\boldsymbol{\eta}_d = \boldsymbol{\eta}_{p,d} - \left(\boldsymbol{\eta}_{f,d} + \sum_{i \neq d} \boldsymbol{\eta}_i\right). \tag{5.21}$$

Finally, the NPs are distributed to all $D \setminus d$ computing nodes, followed by the next iteration. The number of iterations is dependent on the rate of convergence of $\boldsymbol{\eta}_d$ and is treated as a fixed parameter. The EP SMCMC algorithm is described by Algorithm 16.

5.3.1 Proposal Distributions

The standard SMCMC framework consists of two sampling stages, the joint draw and refinement step. However, the framework is flexible in the sense that both sampling stages sample from the target distribution and are thus not both necessarily required for operation. The joint draw has the advantage of only requiring a single evaluation of the measurements. The refinement step introduces additional computational complexity but has also shown to significantly increase the efficiency of the sampling in higher dimensional state space models. Once an appropriate architecture for the SMCMC is selected, there is additional flexibility which arises in the form of selection of the proposal distributions. A common choice for the joint draw is to utilise the following proposal distribution:

$$q_1\left(\boldsymbol{x}_k, \boldsymbol{x}_{k-1} | \boldsymbol{x}_k^{m-1}, \boldsymbol{x}_{k-1}^{m-1}\right) = p(\boldsymbol{x}_k | \boldsymbol{x}_{k-1}) \frac{1}{N} \sum_{j=N_b+1}^{N+N_b} \delta(\boldsymbol{x}_{k-1} - \boldsymbol{x}_{k-1}^{(j)}). \tag{5.22}$$

In this case, the MH acceptance probability simplifies to a ratio of two likelihoods. This is typically followed by the following proposal distributions for the refinement step:

$$\begin{aligned} q_2\left(\boldsymbol{x}_{k-1} | \boldsymbol{x}_k^m, \boldsymbol{x}_{k-1}^m\right) &= p(\boldsymbol{x}_{k-1} | \boldsymbol{x}_k, \boldsymbol{z}_{1:k}) \\ &= \sum_{j=N_b+1}^{N_b+N} \frac{p(\boldsymbol{x}_k = \boldsymbol{x}_k^m | \boldsymbol{x}_{k-1}^j)}{\sum_i^N p(\boldsymbol{x}_k = \boldsymbol{x}_k^m | \boldsymbol{x}_{k-1}^i)} \delta(\boldsymbol{x}_{k-1} - \boldsymbol{x}_{k-1}^{(j)}), \end{aligned} \tag{5.23}$$

and

$$q_{3,p}\left(\boldsymbol{x}_{\Omega_p,k} | \boldsymbol{x}_k^m, \boldsymbol{x}_{k-1}^m\right) = p(\boldsymbol{x}_{\Omega_p,k} | \boldsymbol{z}_{1:k}, \boldsymbol{x}_{k-1}, \boldsymbol{x}_{\{1,\ldots,N_d\} \setminus \Omega_p,k}), \tag{5.24}$$

thus the acceptance ratios ρ_2 and $\{\rho_{3,p}\}_{p=1}^P$ will be equal to 1, leading to a refinement stage equivalent to a series of "perfect" Gibbs samplers [98].

However, sampling from (5.23) is possible at the expense of a large computational cost. Nevertheless the advantage is that this quantity does not depend on the data which is the main challenge in a setting consisting of massive amounts of data. It is also possible to avoid this complexity by using a uniform draw from an index, the acceptance ratio will then reduce to the ratio of two prior distributions.

Typically, sampling from (5.24) is not possible. Alternatively, the proposal distribution in (5.24) can be replaced with a conditional prior or random-walk [108]. An additional advantage of the EP-SMCMC framework is that each computing node d can utilise the information from the measurements at the other $D \setminus d$ computing nodes in the proposal distribution. It has recently been shown in [109] how information about the measurements can be utilised in the standard SMCMC framework, however, this typically requires additional computations and the evaluation of gradients of the likelihood.

5.4 Expectation Propagation and the Particle Filter

In this section a variation of the EP-SMCMC is presented. This entails replacing the MCMC sampling mechanism with importance sampling, i.e. the PF, thus referred to as the EP-PF. The formulation of the filter is similar, with some subtle differences. In contrast to the EP-SMCMC, the EP-PF target distribution is the marginal filtering posterior state pdf, $p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k})$, factorised accordingly:

$$p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k}) \propto p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1}) \prod_{d=1}^{D} p(\boldsymbol{z}_{d,k}|\boldsymbol{x}_k). \tag{5.25}$$

For this scenario, the local filtering posterior state pdf at each processing node d is given by:

$$p_d(\boldsymbol{x}_k|\boldsymbol{z}_{1:k}) \propto p(\boldsymbol{z}_{d,k}|\boldsymbol{x}_k) p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1}) \prod_{i \neq d} \pi(\boldsymbol{x}_k|\boldsymbol{\eta}_i), \tag{5.26}$$

To compute the NPs, the likelihood term for node d in (5.26) is replaced by the approximated likelihood term:

$$\widehat{p}_d(\boldsymbol{x}_k|\boldsymbol{z}_{1:k}) \propto \pi(\boldsymbol{x}_k|\boldsymbol{\eta}_d)p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1})\prod_{i\neq d}\pi(\boldsymbol{x}_k|\boldsymbol{\eta}_i). \tag{5.27}$$

The natural parameters can then be found through the minimisation of the KL divergence, $\mathrm{KL}(p_d(\boldsymbol{x}_k|\boldsymbol{z}_{1:k})||\widehat{p}_d(\boldsymbol{x}_k|\boldsymbol{z}_{1:k}))$ in the same manner as in (5.20) and (5.21). A detailed description of the EP-PF is given in Algorithm 17.

5.4.1 Particle Filter Proposal Distributions

Selecting the proposal distribution is an important step during the design of a PF. Utilising a good proposal distribution results in the particles being moved to regions in the state space with higher likelihood values, which helps avoid weight degeneracy. It has been shown [39] that the optimal proposal distribution is the distribution which minimises the variance of the importance weights,

$$q(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^{(j)}, \boldsymbol{z}_k) = p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^{(j)}, \boldsymbol{z}_k). \tag{5.28}$$

However, sampling from this proposal distribution is generally not tractable. There are a variety of techniques which have been proposed to approximate the optimal proposal distribution [30]. A common approach is to simply utilise the transition density,

$$q(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^{(j)}, \boldsymbol{z}_k) = p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^{(j)}), \tag{5.29}$$

due to its direct availability. This approach also simplifies the weight update to be proportional to the evaluation of the likelihood. However, the transition density does not include any information from the measurements and thus moves the particles blindly.

The EP-PF framework allows for an intuitive inclusion of information from the measurements at the neighbouring nodes in the proposal distribution. This can be done when the prior distribution is the same, or approximated as a member of the exponential family used to approximate the likelihood terms. The resulting proposal distribution is given by

$$q(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^{(j)}, \boldsymbol{z}_k) = \pi(\boldsymbol{x}_k|\boldsymbol{\eta}_p), \tag{5.30}$$

where $\boldsymbol{\eta}_p = \boldsymbol{\eta}_c + \sum_{i \neq d} \boldsymbol{\eta}_i$, and $\boldsymbol{\eta}_c$ represents the natural parameters of the transition density.

5.5 Performance Evaluation

In this section the performance of the proposed adaptive subsampling SMCMC algorithm, and EP SMCMC and PF algorithms, referred to as AS-SMCMC, EP-SMCMC and EP-PF, respectively, is studied. See Section 3.7.1 for details about the computing platform. All results are averaged over 50 MC runs. Refer to Appendix C for further information related to the calculation of the Hessian of the log likelihood required for the AS-SMCMC in this section.

5.5.1 EP-SMCMC and EP-PF considerations

For the examples presented in this section, the member of the exponential family selected to approximate the likelihood terms is the multivariate Gaussian distribution. For this case the NPs are given by:

$$\boldsymbol{\eta} = \left(\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}, \boldsymbol{\Sigma}^{-1}\right)^{\top}, \tag{5.31}$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ represent the mean and covariance of the multivariate Gaussian distribution. In this case, the NPs update in (5.21) simplifies to:

$$\begin{aligned}
\boldsymbol{\Sigma}_d^{-1}\boldsymbol{\mu}_d &= \boldsymbol{\Sigma}_{p,d}^{-1}\boldsymbol{\mu}_{p,d} - \left(\boldsymbol{\Sigma}_{f,d}^{-1}\boldsymbol{\mu}_{f,d} + \sum_{i \neq d} \boldsymbol{\Sigma}_i^{-1}\boldsymbol{\mu}_i\right) \\
\boldsymbol{\Sigma}_d^{-1} &= \boldsymbol{\Sigma}_{p,d}^{-1} - \left(\boldsymbol{\Sigma}_{f,d}^{-1} + \sum_{i \neq d} \boldsymbol{\Sigma}_i^{-1}\right),
\end{aligned} \tag{5.32}$$

where standard techniques are used to obtain unbiased mean and covariance estimates for the discrete distributions. It is important to note that the difference between two positive definite matrices is not necessarily itself positive definite. Techniques, such as $SoftAbs$ [20], can be used to ensure that the result remains positive definite.

5.5.2 Example 1: Dynamic Gaussian Process with Gaussian likelihood

The first example is based on a Gaussian state space model with corresponding transition density and likelihood,

$$\begin{aligned} p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) &= \mathcal{N}\left(\boldsymbol{x}_k; \boldsymbol{A}\boldsymbol{x}_{k-1}, \boldsymbol{Q}\right) \\ p(\boldsymbol{z}_{c,k}|\boldsymbol{x}_k) &= \mathcal{N}\left(\boldsymbol{z}_{c,k}; \boldsymbol{H}\boldsymbol{x}_k, \boldsymbol{R}\right). \end{aligned} \tag{5.33}$$

The measurements are assumed independent, hence resulting in the joint likelihood expression for all measurements:

$$p(\boldsymbol{z}_k|\boldsymbol{x}_k) = \prod_{c=1}^{M_k} p(\boldsymbol{z}_{c,k}|\boldsymbol{x}_k). \tag{5.34}$$

The advantage of studying the Gaussian model is that the KF [68] can be used as a benchmark for performance. In this example the performance of the MCMC based techniques is considered. Unless otherwise specified, the following parameters were utilised for all experiments. The filter parameters include: the number of particles, SMCMC & AS-SMCMC, $N_p = 4000$, EP-SMCMC, $N_p = 500$ for each computing node (number of computing nodes, $D = 4$); the number of EP iterations, $L = 2$; the subsampling parameters, $\gamma_s = 1.2$, $\delta_s = 0.1$, $p_s = 2$. The simulation parameters include: the number of measurements at each time step, $M = 500$; the total simulation time, $T_{tot} = 20$ s; the transition density parameters, $Q = 0.08$, $A = 0.9$; the likelihood parameters, $H = 1$, $R = 2$; the state space dimension size, $N_d = 1$.

For this example a SMCMC framework consisting of only a refinement step for all three algorithms was utilised. In addition, the proposal distribution in (5.23) was used for the first step in refinement. The conditional posterior for the second refinement step for the SMCMC and AS-SMCMC algorithms is:

$$p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^m, \boldsymbol{z}_{1:k}) = p(\boldsymbol{z}_k|\boldsymbol{x}_k)p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^m). \tag{5.35}$$

The following proposal distribution was selected:

$$q_3\left(\boldsymbol{x}_k|\boldsymbol{x}_k^m, \boldsymbol{x}_{k-1}^m\right) = p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^m). \tag{5.36}$$

In the case of EP-SMCMC, the conditional posterior for local computing node d is given by:

$$p_d(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^m, \boldsymbol{z}_{1:k}) = p(\boldsymbol{z}_{k,d}|\boldsymbol{x}_k)p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^m)\prod_{i\neq d}\pi(\boldsymbol{x}_k|\boldsymbol{\eta}_i). \tag{5.37}$$

The following proposal distribution was selected:

$$\begin{aligned} q_3\left(\boldsymbol{x}_k|\boldsymbol{x}_k^m, \boldsymbol{x}_{k-1}^m\right) &\propto p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^m)\prod_{i\neq d}\pi(\boldsymbol{x}_k|\boldsymbol{\eta}_i), \\ &= \mathcal{N}\left(\boldsymbol{x}_k; \boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q\right). \end{aligned} \tag{5.38}$$

where $\boldsymbol{\mu}_q$ and $\boldsymbol{\Sigma}_q$ are derived from the NPs $\boldsymbol{\eta}_q = \boldsymbol{\eta}_{g,d} + \sum_{i\neq d}\boldsymbol{\eta}_i$, and $\boldsymbol{\eta}_{g,d}$ represents the NPs of the transition density, $p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^m)$. Table 5.1 illustrates the computational complexity of the algorithms for 500 and 5000 measurements. It is interesting to note that an increase in measurements leads to an increase in computational saving in AS-SMCMC.

Tables 5.2 and 5.3 compare the acceptance rates of the algorithms for the first and second refinement steps, respectively. In Table 5.2, the acceptance probabilities for the different algorithms do not differ significantly. This is expected since all three algorithms utilise the same proposal distribution and acceptance ratio for the first refinement step, and additionally, this refinement step is not dependent on the data. Table 5.3 highlights the improvement in acceptance ratio for the EP-SMCMC in this scenario. The increase during the first EP iteration is due to the relative decrease in the number of measurements processed by each computing node. The large increase during the second EP iteration is due to a smarter proposal distribution

Table 5.1: Algorithm computation time per time step.

Algorithms	$M = 500$	
	Time (s)	Computational Gain (%)
SMCMC	114.75	0
AS-SMCMC	69.54	39.4
EP-SMCMC	9.89	91.38
	$M = 5000$	
SMCMC	1087.93	0
AS-SMCMC	274.60	74.76
EP-SMCMC	96.40	91.14

Table 5.2: Acceptance rates for the first refinement step.

Algorithm	Acceptance Rate (Min, Median, Mean, Max)
SMCMC	(30.93, 94.35, 89.72, 96.57)
AS-SMCMC	(30.90, 94.43, 89.70, 96.54)
EP-SMCMC (L = 1)	(34.15, 93.99, 89.47, 94.86)
EP-SMCMC (L = 2)	(30.86, 94.54, 89.77, 96.59)

Table 5.3: Acceptance rates for the second refinement step.

Algorithm	Acceptance Rate (Min, Median, Mean, Max)
SMCMC	(8.82, 23.44, 21.26, 25.78)
AS-SMCMC	(9.04, 24.24, 21.95, 26.75)
EP-SMCMC (L = 1)	(19.76, 42.07, 38.46, 45.01)
EP-SMCMC (L = 2)	(72.11, 76.24, 75.86, 77.69)

which incorporates the information about the measurements from the other computing nodes.

The Kolmogorov-Smirnov (KS) statistic is used to gauge the relative accuracy to correctly approximate empirically the filtering distribution of interest by the algorithms. The KS statistic is given by:

$$KS = \max_{x} \left(\widehat{F}(x) - G(x) \right), \tag{5.39}$$

where $\widehat{F}(x)$ is an empirical cumulative density function (cdf) and $G(x)$ is a continuous cdf. In this setting, $\widehat{F}(x)$ is the empirical cdf of the discrete posterior distribution estimated by the SMCMC algorithms, and $G(x)$ the cdf of a Gaussian distribution with parameters updated by a KF. For EP-SMCMC, the samples from all D computing nodes at the final EP iteration are considered. It is worth while mentioning that the transmission of the samples from the D computing nodes to a single computing node was utilised in this experiment but is not necessary when only estimates are required to be extracted. For example, since the samples in SMCMC are unweighted, the global mean can be established through the averaging of the individual local means. The KS statistic for several different filter configurations is illustrated in Figure 5.1 for both the case of 500 and 5000 measurements. It is first noted that the SMCMC and AS-SMCMC share almost identical performance. This was expected as the goal of AS-SMCMC is to make the same accept or reject decision in the embedded MCMC algorithms as in SMCMC, only while evaluating less measurements. From Figure 5.1a, it is noted that the performance of the EP-SMCMC varies depending on the configuration. Doubling the number of computing nodes, while halving the number of samples, conserves the total number of samples while further increasing the computational efficiency at the cost of an increase in error. While in the other extreme case, increasing the number of samples while keeping the number of computing nodes fixed, significantly increases the accuracy while decreasing the computational gain. The case of N_p equal to 1000, results in the same number of samples for all three algorithms. It is clear that even in this scenario, there is an increase in performance, which can be attributed to the increased acceptance rate which results in a more diverse empirical cdf. The EP-SMCMC algorithm is also well suited in this specific example due to the Gaussian nature of the model and utilisation of the Gaussian density for the approximate likelihood terms.

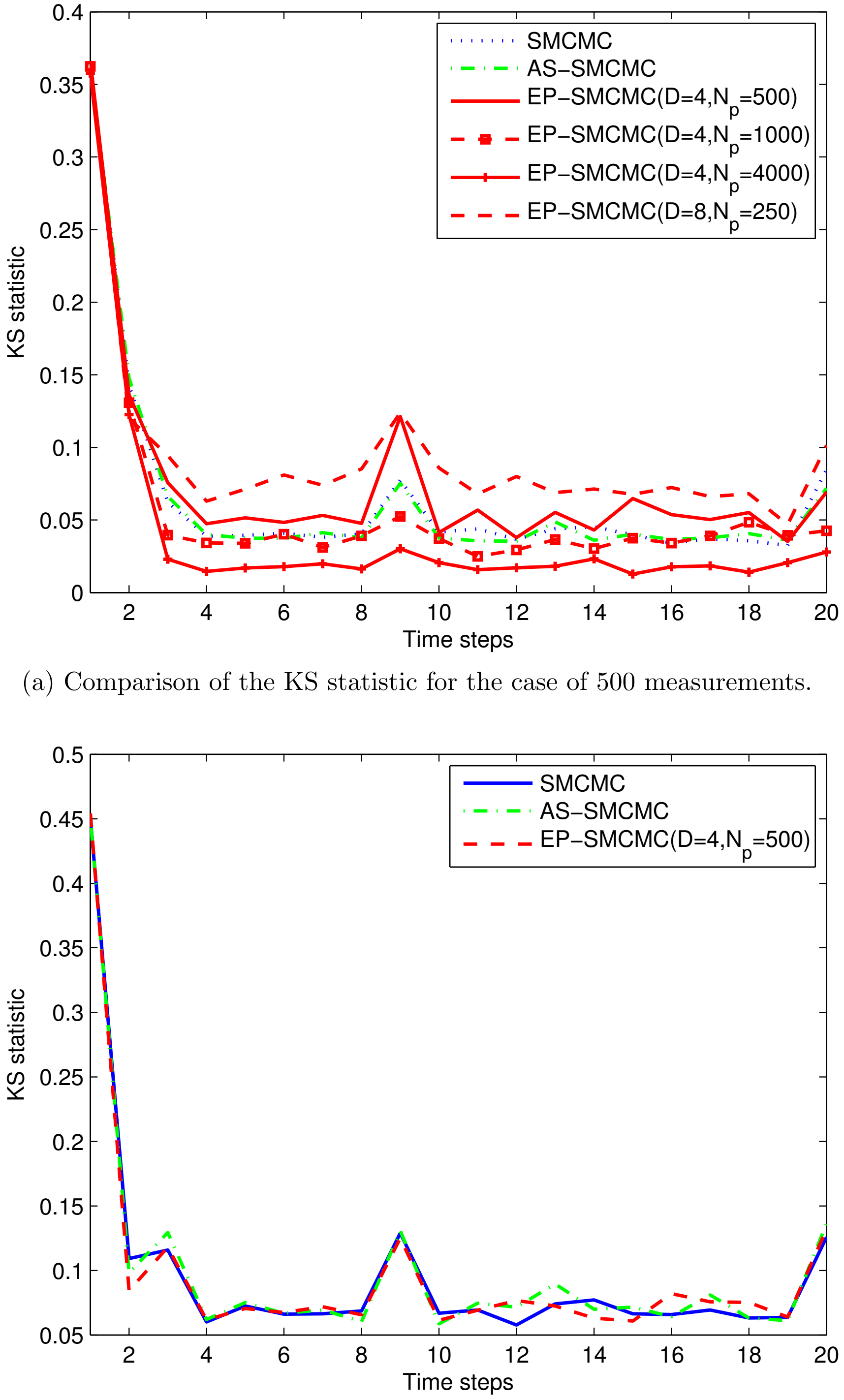
5.5.3 Example 2: Multiple Object Tracking in Clutter

In this example the application of multiple point object tracking in clutter is presented. The state vector consists of the positions and velocities of N_T objects in a two dimensional space, $\boldsymbol{x}_k = (x_{1,k} \dots x_{N_T,k}, y_{1,k} \dots y_{N_T,k}, \dot{x}_{1,k} \dots \dot{x}_{N_T,k}, \dot{y}_{1,k} \dots \dot{y}_{N_T,k})^\top$. In this example it is assumed that the number of objects, N_T, is fixed and known, and that each object evolves independently of the other objects. The motion of each object adheres to the near constant velocity model. This results in the marginal state transition density for object j having the form

$$p(\boldsymbol{x}_{j,k}|\boldsymbol{x}_{j,k-1}) = \mathcal{N}(\boldsymbol{x}_{j,k}|\boldsymbol{A}\boldsymbol{x}_{j,k-1}, \boldsymbol{Q}), \tag{5.40}$$

where $\mathcal{N}(\cdot)$ represents the normal distribution, and matrices $\boldsymbol{A}$ and $\boldsymbol{Q}$ are defined as $\boldsymbol{A} = \begin{bmatrix} \mathrm{I}_2 & T_s\mathrm{I}_2 \\ \boldsymbol{0}_2 & \mathrm{I}_2 \end{bmatrix}$ and $\boldsymbol{Q} = \sigma_x^2 \begin{bmatrix} (T_s^3/3)\mathrm{I}_2 & (T_s^2/2)\mathrm{I}_2 \\ (T_s^2/2)\mathrm{I}_2 & T_s\mathrm{I}_2 \end{bmatrix}$, where $T_s = t_k - t_{k-1}$, and I_2 represents the 2×2 identity matrix.

In a similar fashion to Chapter 3 and 4, the total number of measurements received



(a) Comparison of the KS statistic for the case of 500 measurements.

(b) Comparison of the KS statistic for the case of 5000 measurements.

Figure 5.1: The KS statistic for the several configurations of the SMCMC based algorithms relative to the KF.

is given by $M_k = N_T M_{T,k} + M_{C,k}$. The likelihood density takes the form [49]:

$$p(\boldsymbol{z}_k|\boldsymbol{x}_k) = \frac{e^{-\mu_k}}{M_k!} \prod_{i=1}^{M_k} \left(\lambda_C p_C(\boldsymbol{z}_{i,k}) + \sum_{j=1}^{N_T} \lambda_X p_X(\boldsymbol{z}_{i,k}|\boldsymbol{x}_{j,k}) \right), \tag{5.41}$$

where $\mu_k = \lambda_C + N_T \lambda_T$, $p_X(\cdot)$ and $p_C(\cdot)$ represent the likelihood of an object and clutter measurement respectively. Each individual measurement represents a point in the two dimensional observation space, $\boldsymbol{z}_{i,k} = [z_{i,x,k}, z_{i,y,k}]^\top$. In the case of a measurement from an object, the likelihood is modelled as $p_X(\boldsymbol{z}_{i,k}|\boldsymbol{x}_{j,k}) = \mathcal{N}(\boldsymbol{z}_{i,k}; \boldsymbol{x}_{j,k}, \boldsymbol{\Sigma})$. The clutter measurements are independent of the states of the objects and are uniformly distributed in the visible region of the sensor, resulting in the clutter likelihood taking the form of $p_C(\boldsymbol{z}_{i,k}) = U_{R_x}(z_{i,x,k}) U_{R_y}(z_{i,y,k})$, where $A_c = R_x \times R_y$ represents the clutter area.

The following parameters, unless otherwise specified, were used for all experiments. The filter parameters include: the number of particles, for SMCMC & AS-SMCMC, $N_p = 4000$, and EP-SMCMC, $N_p = 500$ for each computing node (number of computing nodes, $D = 4$); the covariance associated with the proposal for the refinement step, $\boldsymbol{\Sigma}_r = 0.01\mathrm{I}$; the subsampling parameters, $\gamma_s = 1.2$, $\delta_s = 0.1$, and $p_s = 2$. The Simulation parameters include: a total running time, $T = 20$, with sampling time, $T_s = 1$; the variance associated with the motion model $\sigma_x = 0.5$; the object observation model parameters, $\lambda_X = 1500$, and $\boldsymbol{\Sigma} = \mathrm{I}$; the clutter parameters, $\lambda_C = 4000$, and $A_c = 4 \times 10^4$; the number of objects $N_T = 3$.

For this example the SMCMC framework consists of a joint draw and a local refinement step on the current state only, for all three algorithms. The proposal distribution for the joint draw in the SMCMC and AS-SMCMC is given by:

$$q_1\left(\boldsymbol{x}_k, \boldsymbol{x}_{k-1}|\boldsymbol{x}_k^{m-1}, \boldsymbol{x}_{k-1}^{m-1}\right) = \frac{1}{N} p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) \sum_{j=N_b+1}^{N+N_b} \delta(\boldsymbol{x}_{k-1} - \boldsymbol{x}_{k-1}^{(j)}). \tag{5.42}$$

The following proposal distribution was selected for the local refinement step:

$$q_{3,p}\left(\boldsymbol{x}_{\Omega_p,k}|\boldsymbol{x}_k^m, \boldsymbol{x}_{k-1}^m\right) = \mathcal{N}(\boldsymbol{x}_{\Omega_p,k}^m, \boldsymbol{\Sigma}_r), \tag{5.43}$$

where $\boldsymbol{x}_{\Omega_p,k} = (x_{p,k}, y_{p,k}, \dot{x}_{p,k}, \dot{y}_{p,k})^\top$ corresponds to the p-th object. This proposal represents a random walk move with covariance $\boldsymbol{\Sigma}_r$. In the case of EP-SMCMC, the proposal distribution for the joint draw is given by:

$$\begin{aligned} q_1\left(\boldsymbol{x}_k, \boldsymbol{x}_{k-1} | \boldsymbol{x}_k^{m-1}, \boldsymbol{x}_{k-1}^{m-1}\right) &\propto \frac{1}{N} p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) \prod_{i \neq d} \pi(\boldsymbol{x}_k|\boldsymbol{\eta}_i) \sum_{j=N_b+1}^{N+N_b} \delta(\boldsymbol{x}_{k-1} - \boldsymbol{x}_{k-1}^{(j)}) \\ &= \frac{1}{N} \mathcal{N}\left(\boldsymbol{x}_k; \boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q\right) \sum_{j=N_b+1}^{N+N_b} \delta(\boldsymbol{x}_{k-1} - \boldsymbol{x}_{k-1}^{(j)}). \end{aligned} \tag{5.44}$$

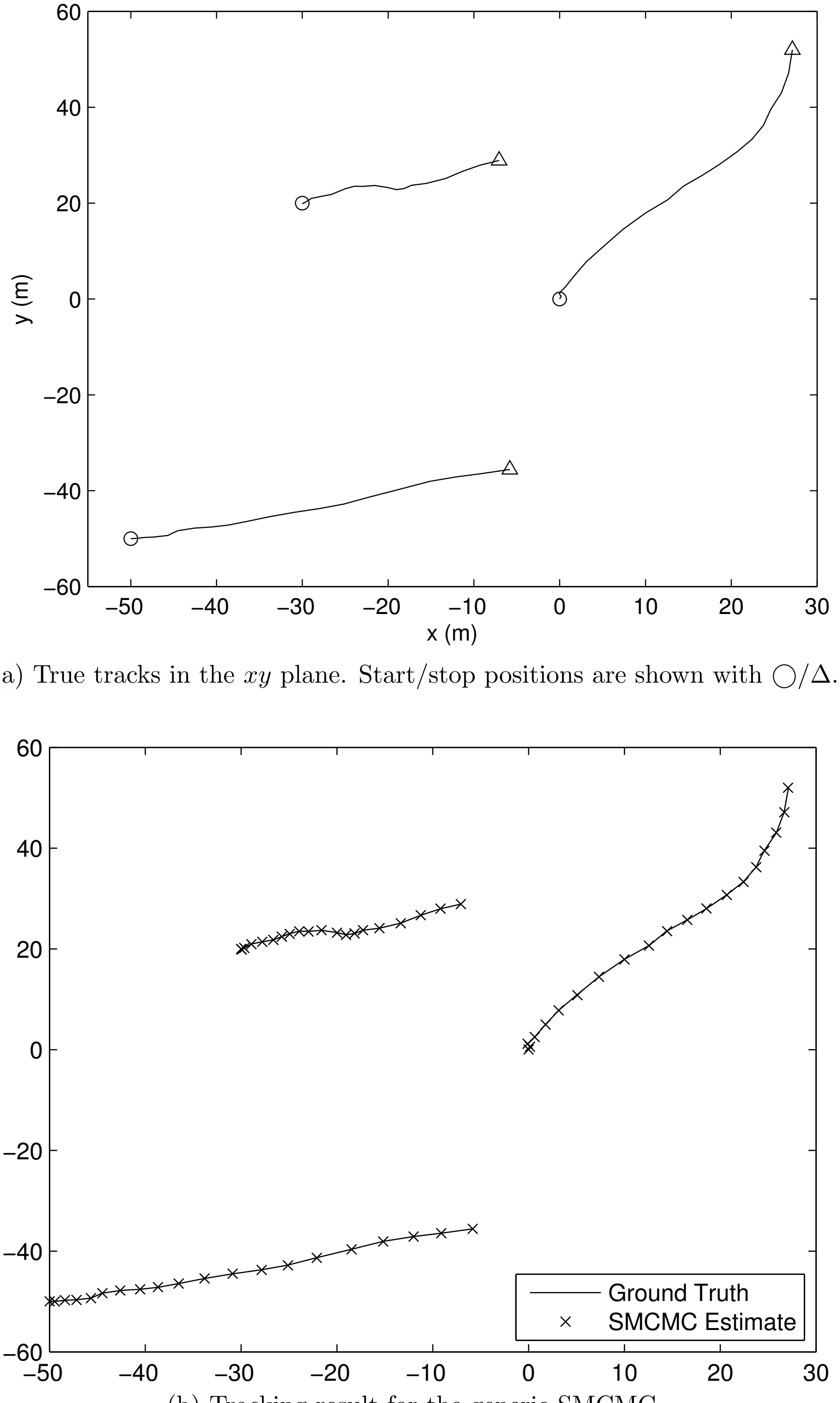
where $\boldsymbol{\mu}_q$ and $\boldsymbol{\Sigma}_q$ are derived from the NPs $\boldsymbol{\eta}_q = \boldsymbol{\eta}_{g,d} + \sum_{i \neq d} \boldsymbol{\eta}_i$, and $\boldsymbol{\eta}_{g,d}$ represents the NPs of the transition density, $p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$. The same local proposal distribution as used in SMCMC and AS-SMCMC, equation (5.43), was selected for the refinement step in EP-SMCMC.

It is interesting to note that in this example the likelihood expression, given in (5.41), is independent of an object's velocities. Therefore, when determining the natural parameters of the approximate likelihood terms using (5.32), the subtraction of the precision terms between the posterior and predictive posterior state pdfs were forced to zero for all the dimensions related to object velocity. This eliminates potential numerical problems that could arise in the empirical estimation of the natural parameters from a finite number of samples.

The estimated tracks for a single simulation run are shown in Figure 5.2. An abundance in measurements results in all algorithms returning accurate estimate results. The RMSE, averaged over all the position dimensions for the three objects, is given in Figure 5.3. There is a negligible increase in RMSE for the positions related to the EP-SMCMC. The computation time per time step for each algorithm is illustrated in Table 5.4. The acceptance rates of the joint draw and refinement steps are illustrated in Table 5.5 and 5.6 respectively.

5.5.4 Example 3: Object Tracking in a Distributed Sensor Network

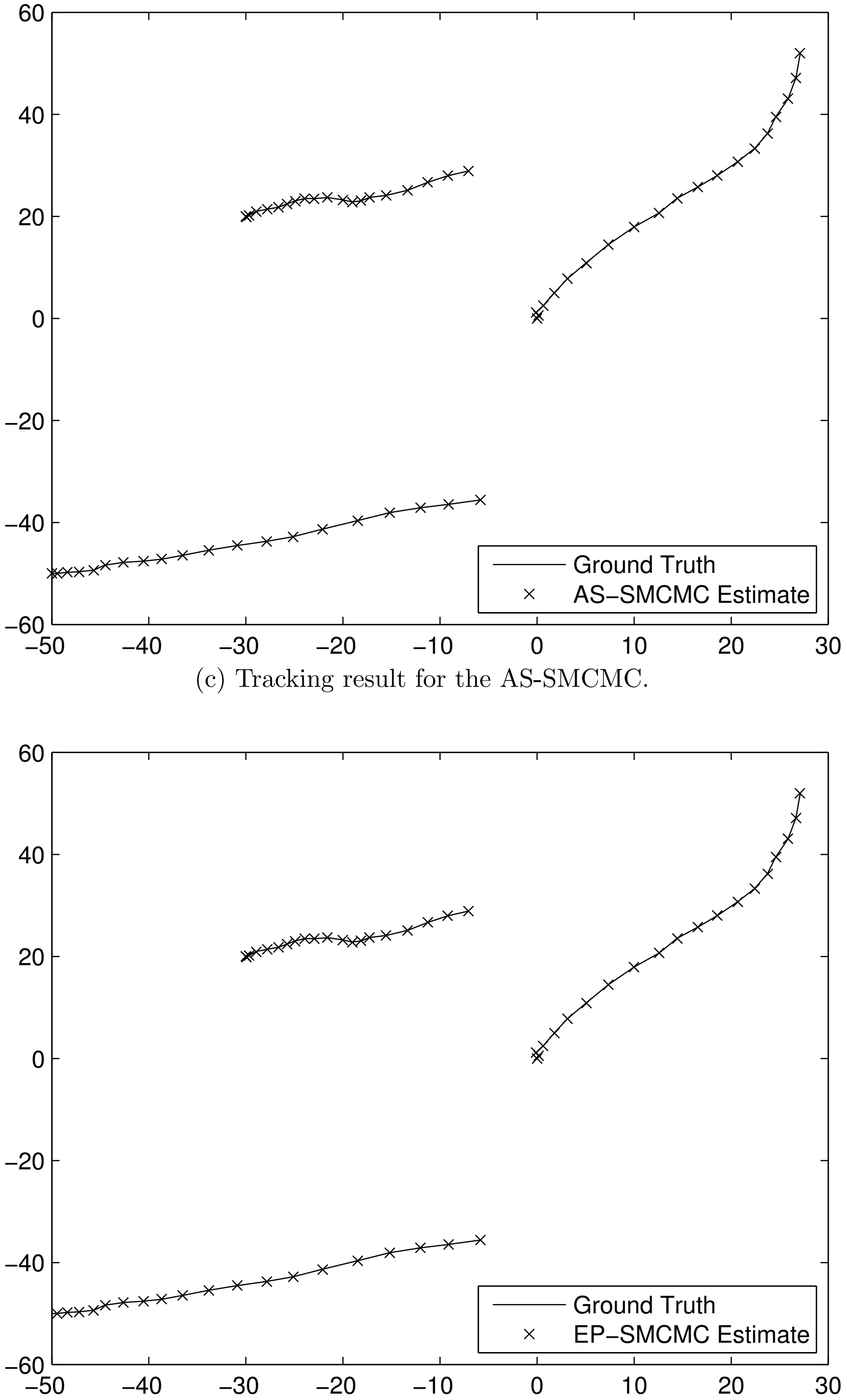
Consider the scenario of an object moving through a highly cluttered environment. A distributed sensor network, consisting of several data rich sensor nodes, is utilised to



(a) True tracks in the xy plane. Start/stop positions are shown with ○/Δ.

(b) Tracking result for the generic SMCMC.

Figure 5.2: Tracking results for a single run of the algorithms.



(c) Tracking result for the AS-SMCMC.

(d) Tracking result for the EP-SMCMC.

Figure 5.2: Tracking results for a single run of the algorithms. (cont.)

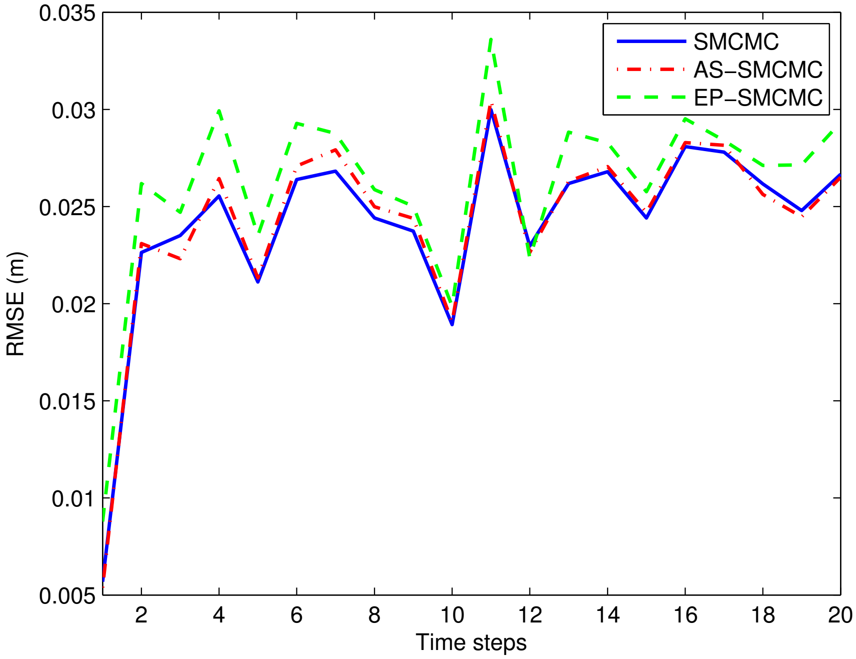


Figure 5.3: The RMSE averaged over the position dimensions for the tracking simulation.

Table 5.4: Algorithm computation time per time step.

Algorithms	Time (min)	Computational Gain (%)
SMCMC	505.42	0
AS-SMCMC	388.82	23.07
EP-SMCMC	59.06	88.31

Table 5.5: Acceptance rates for the joint draw.

Algorithm	Acceptance Rate (Min, Median, Mean, Max)
SMCMC	(0, 0, 0, 0)
AS-SMCMC	(0, 0, 0, 0)
EP-SMCMC (L = 1)	(0, 0 , 0.002, 0.01)
EP-SMCMC (L = 2)	(0.04, 1.35, 1.35, 3.09)

Table 5.6: Acceptance rates for the refinement step.

Algorithm	Acceptance Rate (Min, Median, Mean, Max)
SMCMC	(28.92, 29.45, 29.45, 30.08)
AS-SMCMC	(28.94, 29.31, 29.43, 30.25)
EP-SMCMC (L = 1)	(62.37, 63.60, 63.65, 65.43)
EP-SMCMC (L = 2)	(25.78, 26.76, 26.92, 28.77)

monitor the object which returns multiple object and clutter measurements at each time step and each sensor node. The EP-PF and EP-SMCMC, are well suited for such scenarios since the distributed structure of the scenario matches that of the methods. These methods are compared with the standard PF, described in Algorithm 2, for the inference of the latent states of the object over several experiments. The motion of the object is modelled with the nearly constant velocity model as in (5.40). Similarly to (5.41), the likelihood per sensor node is represented by:

$$p(\boldsymbol{z}_{d,k}|\boldsymbol{x}_k) \propto \prod_{i=1}^{M_{d,k}} \lambda_X p_X(\boldsymbol{z}_{i,d,k}|\boldsymbol{x}_k) + \lambda_C p_C(\boldsymbol{z}_{i,d,k}), \tag{5.45}$$

with $p_X(\boldsymbol{z}_{i,d,k}|\boldsymbol{x}_k) = \mathcal{N}(\boldsymbol{z}_{i,d,k}; h(\boldsymbol{x}_k), \boldsymbol{\Sigma})$, where $h(\boldsymbol{x}_k) = \sqrt{(x_k - S_{d,x})^2 + (y_k - S_{d,y})^2}$, and $(S_{d,x}, S_{d,y})^\top$ represent the position coordinates of sensor node d.

The metrics used to compare the performance of the filters are: the RMSE of the position, which illustrates the tracking accuracy of the filters; the simulation computation time of the distributed filters; and the communication cost for sharing information between the sensor nodes. Due to the many different variables associated with the speed of a communication link, the number of doubles which are required to be transmitted between nodes by each algorithm in order to infer the filtering posterior state pdf is considered. In the PF, it is required to transmit all the measurements from each sensor node at each time step, to a centralised processing node. Assuming that each sensor node is capable of communicating with the processing node in parallel, then the number of doubles required to be transmitted is given by

$$C_{\text{CPF}} = \max_{1 \leq d \leq D} M_{d,k}. \tag{5.46}$$

For an interconnected network, the communication cost of broadcasting the NPs of each sensor node in the EP based methods is related to the number of EP iterations,

$$C_{\text{EP-PF}} = (L-1)N_{\text{NP}}, \tag{5.47}$$

where N_{NP} is the number of doubles used to represent the NPs.

The following parameters were utilised across all simulations, unless otherwise specified. The number of particles for the PF and EP based methods are $N = 10000$, and $N = 5000$, respectively. The number of MC simulation runs is $N_I = 50$. The number of time simulation steps is $T = 70$. The motion model parameters are $T_s = 1$, and $\sigma_x = 0.5$. The number of sensor nodes is $D = 4$. The object observation model parameters are $\lambda_X = 200$, and $\mathbf{\Sigma} = \mathrm{I}$. The clutter parameters are: $\lambda_C = 100$, and $A_c = 4 \times 10^4$. The number of EP iterations is $L = 2$. The object trajectory and sensor node positions relative to the object for the experiments are illustrated in Figure 5.4.

The number of particles was selected to match the number of particles that are required to be processed at each time step for all the algorithms. The EP iteration, L, determines how many times the particle set is required to be re-evaluated. Results are illustrated for the minimum number of EP iterations. The average RMSE for the position is illustrated in Figure 5.5. Overall, there is a negligible loss in tracking accuracy when using the EP based approaches with only 2 EP iterations when compared to the PF.

For the given experimental setup, the communication cost is given in Table 5.7. It is clear from this result that a significant advantage of the EP based methods is the massive reduction in communication cost. This is due to the ability of the EP based methods to transmit the information found within the measurements at each sensor

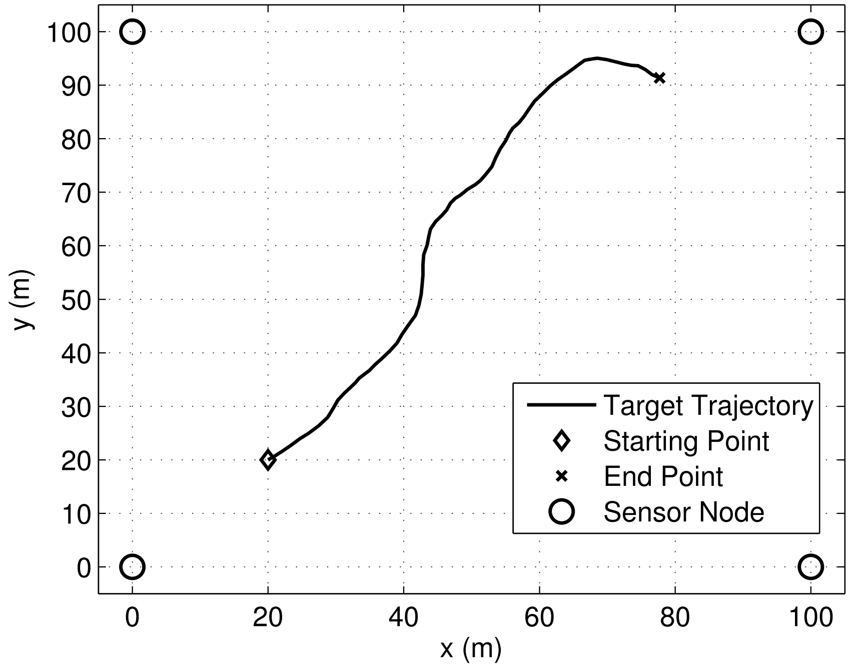


Figure 5.4: Object trajectory and sensor node placement for the experiments.

node in a fixed small number of NPs.

Finally the computation complexity of the distributed methods is illustrated in Table 5.8. The SMCMC based approach incurs a higher cost due to the additional sampling for the state $\boldsymbol{x}_{k-1}$. However, MCMC sampling has been shown to be more efficient in high dimensional state spaces [69, 108], thus the EP-SMCMC is favoured for high dimensional state space models.

5.6 Summary

In this chapter, the problem of object tracking with an excessive number of measurements is considered. A large number of measurements can be detrimental to real time systems due to increases in computational time. To this end, three novel methods were presented to reduce the computational burden.

The first method is based on the introduction of adaptive subsampling into an SMCMC framework. In this case the computational load is reduced by only con-

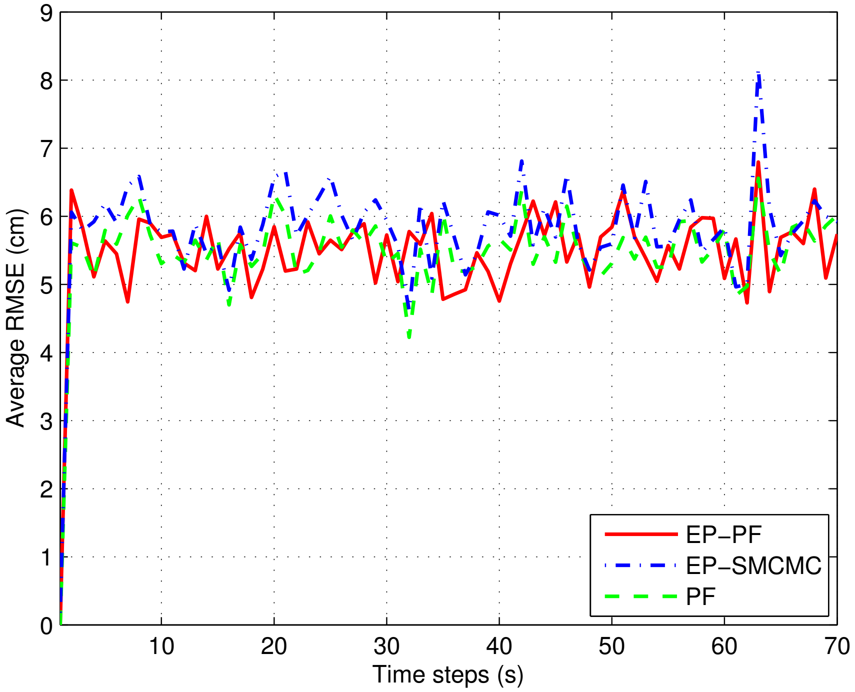


Figure 5.5: Average RMSE for the position of the object.

sidering a subset of all the measurements at each time step. The samples obtained from the SMCMC are a controlled approximation of the posterior state pdf of all measurements.

The second and third methods are based on combining the EP algorithm with an SMCMC and PF framework. In this case the computational load is reduced by separating the measurements into batches and processing the batches in parallel. The samples obtained from the SMCMC and PF are approximations of the posterior state pdf of all measurements due to the manner in which the information from the batches is combined.

The power of the methods was displayed through three examples, with comparisons to a standard SMCMC and PF algorithm. The first example was based on a linear Gaussian model. This has the advantage of having the KF as a benchmark for performance. The second example considers the problem of multiple object tracking. The third example considers the problem of tracking an object in a distributed net-

Table 5.7: Average number of communicated doubles for one time cycle (from k to $k+1$) for each method.

Method	Average number of communicated doubles per sensor node
PF	300
EP-PF	20
EP-SMCMC	20

Table 5.8: Distributed method computational expense for one time cycle (from k to $k+1$).

Method	Time (s)
EP-PF	7.02
EP-SMCMC	10.78

work, which suits the structure of the EP based approaches. The proposed methods greatly reduced the computational time for Bayesian filtering, by up to 90% in the conducted experiments. The experiments also illustrated that the EP based methods have up to a 93% reduction in communication costs when compared with a centralised PF framework in a distributed network.

The flexibility of the proposed methods allows for their application in a wide variety of problem areas, e.g. these methods can be applied to advanced multiple target tracking scenarios through a joint state vector approach with the incorporation of an existence variable as in Chapter 4, or a reversible jump mechanism [23].

Algorithm 14 Adaptive Subsampling Sequential Markov Chain Monte Carlo

```
Initialise particle set: $\{\boldsymbol{x}_0^{(j)}\}_{j=1}^N$
Determine initial proxy parameters.
for $k = 1,...,T$ do
    for $m = 1,...,N+N_b$ do
        if $m = 1 \vee N_b$ then
            Update proxy parameters.
        end if
        Joint Draw
        Propose $\{\boldsymbol{x}_k^*, \boldsymbol{x}_{k-1}^*\} \sim q_1\left(\boldsymbol{x}_k, \boldsymbol{x}_{k-1} | \boldsymbol{x}_k^{m-1}, \boldsymbol{x}_{k-1}^{m-1}\right)$
        Compute $\psi_1(\boldsymbol{x}_k^*, \boldsymbol{x}_{k-1}^*, \boldsymbol{x}_k^{m-1}, \boldsymbol{x}_{k-1}^{m-1}) = \frac{1}{M_k} \log \left[ u \frac{p(\boldsymbol{x}_k^{m-1} | \boldsymbol{x}_{k-1}^{m-1}) q_1\left(\boldsymbol{x}_k^*, \boldsymbol{x}_{k-1}^* | \boldsymbol{x}_k^{m-1}, \boldsymbol{x}_{k-1}^{m-1}\right)}{p(\boldsymbol{x}_k^* | \boldsymbol{x}_{k-1}^*) q_1\left(\boldsymbol{x}_k^{m-1}, \boldsymbol{x}_{k-1}^{m-1} | \boldsymbol{x}_k^*, \boldsymbol{x}_{k-1}^*\right)} \right]$
        Compute $\Lambda_1^{S_{m,k}}(\boldsymbol{x}_k^*, \boldsymbol{x}_k^{m-1})$ and $\{\wp_i(\boldsymbol{x}_k^{m-1}, \boldsymbol{x}_k^*)\}_{i=1}^{M_k}$ with the routine described by
        Algorithm 15.
        if $\Lambda_1^{S_{m,k}}(\boldsymbol{x}_k^*, \boldsymbol{x}_k^{m-1}) > \psi_1(\boldsymbol{x}_k^*, \boldsymbol{x}_{k-1}^*, \boldsymbol{x}_k^{m-1}, \boldsymbol{x}_{k-1}^{m-1}) - \frac{1}{M_k} \sum_{i=1}^{M_k} \wp_i(\boldsymbol{x}_k^{m-1}, \boldsymbol{x}_k^*)$ then
            $\{\boldsymbol{x}_k^m, \boldsymbol{x}_{k-1}^m\} = \{\boldsymbol{x}_k^*, \boldsymbol{x}_{k-1}^*\}$
        else
            $\{\boldsymbol{x}_k^m, \boldsymbol{x}_{k-1}^m\} = \{\boldsymbol{x}_k^{m-1}, \boldsymbol{x}_{k-1}^{m-1}\}$
        end if
        Refinement
        Propose $\boldsymbol{x}_{k-1}^* \sim q_2\left(\boldsymbol{x}_{k-1} | \boldsymbol{x}_k^m, \boldsymbol{x}_{k-1}^m\right)$
        Compute the MH acceptance probability
            $$\rho_2 = \min\left(1, \frac{p(\boldsymbol{x}_{k-1}^* | \boldsymbol{x}_k^m, \boldsymbol{z}_{1:k})}{q_2\left(\boldsymbol{x}_{k-1}^* | \boldsymbol{x}_k^m, \boldsymbol{x}_{k-1}^m\right)} \frac{q_2\left(\boldsymbol{x}_{k-1}^m | \boldsymbol{x}_k^m, \boldsymbol{x}_{k-1}^*\right)}{p(\boldsymbol{x}_k^m | \boldsymbol{x}_{k-1}^m, \boldsymbol{z}_{1:k})}\right)$$
        Accept $\boldsymbol{x}_{k-1}^m = \boldsymbol{x}_{k-1}^*$ with probability $\rho_2$
        Divide $\boldsymbol{x}_k$ into $P$ disjoint blocks $\{\Omega_p\}_{p=1}^P$ such that $\bigcup_p \Omega_p = \{1, ..., N_d\}$ and
        $\Omega_p \cap \Omega_q = \emptyset, \forall p \neq q$
        for $p = 1,...,P$ do
            Propose $\boldsymbol{x}_{\Omega_p,k}^* \sim q_{3,p}\left(\boldsymbol{x}_{\Omega_p,k} | \boldsymbol{x}_k^m, \boldsymbol{x}_{k-1}^m\right)$
            Compute $\psi_{3,p}(\boldsymbol{x}_{\Omega_p,k}^*, \boldsymbol{x}_{\Omega_p,k}^m, \boldsymbol{x}_{k-1}^m) = \frac{1}{M_k} \log \left[ u \frac{p(\boldsymbol{x}_{\Omega_p,k}^m | \boldsymbol{x}_{k-1}^m) q_{3,p}\left(\boldsymbol{x}_{\Omega_p,k}^* | \boldsymbol{x}_k^m, \boldsymbol{x}_{k-1}^m\right)}{p(\boldsymbol{x}_{\Omega_p,k}^* | \boldsymbol{x}_{k-1}^m) q_{3,p}\left(\boldsymbol{x}_{\Omega_p,k}^m | \boldsymbol{x}_k^*, \boldsymbol{x}_{k-1}^m\right)} \right]$
            Compute $\Lambda_{3,p}^{S_{m,k}}(\boldsymbol{x}_{\Omega_p,k}^m, \boldsymbol{x}_{\Omega_p,k}^*)$ and $\{\wp_i(\boldsymbol{x}_{\Omega_p,k}^m, \boldsymbol{x}_{\Omega_p,k}^*)\}_{i=1}^{M_k}$ with the routine described by Algorithm 15.
            if $\Lambda_{3,p}^{S_{m,k}}(\boldsymbol{x}_{\Omega_p,k}^*, \boldsymbol{x}_{\Omega_p,k}^m) > \psi_{3,p}(\boldsymbol{x}_{\Omega_p,k}^*, \boldsymbol{x}_{\Omega_p,k}^m, \boldsymbol{x}_{k-1}^m) - \frac{1}{M_k} \sum_{i=1}^{M_k} \wp_i(\boldsymbol{x}_{\Omega_p,k}^m, \boldsymbol{x}_{\Omega_p,k}^*)$
            then
                $\boldsymbol{x}_{\Omega_p,k}^m = \boldsymbol{x}_{\Omega_p,k}^*$
            end if
        end for
    end for
    Approximation of the marginal posterior state pdf with the following empirical measure:
    $\widehat{p}(\boldsymbol{x}_k | \boldsymbol{z}_{1:k}) = \frac{1}{N} \sum_{j=N_b+1}^{N+N_b} \delta(\boldsymbol{x}_k - \boldsymbol{x}_k^{(j)})$
end for
```

Algorithm 15 Adaptive Subsampling Routine

```
Given: The current and proposed states of the Markov chain, $\{\boldsymbol{x}_k,\ \boldsymbol{x}_k^*\}$, the complete measurement set, $\boldsymbol{z}_k = \{\boldsymbol{z}_{1,k}, ..., \boldsymbol{z}_{M_k,k}\}$, $\delta$, and $\psi(\cdot)$.
Initialise: number of sub-sampled measurements, $S_{m,k} = 0$, Approximate log likelihood ratio subtracted by proxy, $\Lambda = 0$, set of sub-sampled measurements, $\boldsymbol{z}_{*,k} = \emptyset$, initial batchsize, $b = 1$, while loop counter, $w = 0$.
Compute an upper bound for the range, $R_k^B$, according to (5.14).
Compute the proxy, $\{\wp_i(\boldsymbol{x}_k, \boldsymbol{x}_k^*)\}_{i=1}^{M_k}$, according to (5.11).
DONE = FALSE
while DONE == FALSE do
    $w = w + 1$
    $\{\boldsymbol{z}_{S_{m,k}+1,*,k}, ..., \boldsymbol{z}_{b,*,k}\} \sim_{w/repl.} \boldsymbol{z}_k \setminus \boldsymbol{z}_{*,k}$
    $\boldsymbol{z}_{*,k} = \boldsymbol{z}_{*,k} \cup \{\boldsymbol{z}_{S_{m,k}+1,*,k}, ..., \boldsymbol{z}_{b,*,k}\}$
    $\Lambda = \frac{1}{b}\left(S_{m,k}\Lambda + \sum_{i=S_{m,k}+1}^{b}\left[\log\frac{p(\boldsymbol{z}_{i,*,k}|\boldsymbol{x}_k^*)}{p(\boldsymbol{z}_{i,*,k}|\boldsymbol{x}_k)} - \wp_i(\boldsymbol{x}_k, \boldsymbol{x}_k^*)\right]\right)$
    $S_{m,k} = b$
    $\delta_w = \frac{p_s-1}{p_s w^{p_s}}\delta_s$
    Compute $c$ according to (5.6) utilising $\delta_w$.
    $b = \gamma_s S_{m,k} \wedge M_k$
    if $|\Lambda + \frac{1}{M_k}\sum_{i=1}^{M_k}\wp_i(\boldsymbol{x}_k, \boldsymbol{x}_k^*) - \psi(\cdot)| \geq c$ or $S_{m,k} == M_k$ then
        DONE = TRUE
    end if
end while
return $\Lambda$ and $\{\wp_i(\boldsymbol{x}_k, \boldsymbol{x}_k^*)\}_{i=1}^{M_k}$
```

Algorithm 16 Expectation Propagation Sequential Markov chain Monte Carlo

```
Initialise particle set on each computing node: $\{\boldsymbol{x}_0^{(j)}\}_{j=1}^{N}$
for $k = 1, ..., T$ do
    Partition $M_k$ measurements into $D$ sets, $\{\boldsymbol{z}_{d,k}\}_{d=1}^{D}$, and distribute the sets to each corresponding computing node.
    for $L = 1, ..., L$ do {EP iteration index}
        for $d = 1, ..., D$ do {Computing node index (completed in parallel)}
            Follow steps 3 to 19 of Algorithm 13 with (5.17) as the target distribution.
            Determine the NPs of the approximated likelihood term, $\boldsymbol{\eta}_d$, according to (5.21).
            Distribute the NPs of the approximated likelihood term to the set $D \setminus d$ computing nodes.
        end for
    end for
    Filtering distribution approximated with samples from the $D$ computing nodes.
end for
```

Algorithm 17 Expectation Propagation Particle Filter: Algorithm for sensor node d.

```
Initialise particle set: $\{\boldsymbol{x}_0^{(j)}\}_{j=1}^{N}$ according to prior distribution.
for $k = 1,\dots,T$ do
  for $\ell = 1,\dots,L$ do
    if $\ell == 1$ then
      initialise the NPs from the set $D \setminus d$ of sensor nodes: $\{\boldsymbol{\eta}_i\}_{i \neq d}$.
    end if
    for $j = 1,\dots,N$ do
      Sample a particle: $\boldsymbol{x}_k^{(j)} \sim q(\boldsymbol{x}_k | \boldsymbol{x}_{k-1}^{(j)}, \boldsymbol{z}_k)$.
      Update the particle weight:
      $w_k^{(j)} = w_{k-1}^{(j)} \frac{p(\boldsymbol{x}_k^{(j)} | \boldsymbol{x}_{k-1}^{(j)}) p(\boldsymbol{z}_{d,k} | \boldsymbol{x}_k) \prod_{i \neq d} \pi(\boldsymbol{x}_k | \boldsymbol{\eta}_i)}{q(\boldsymbol{x}_k^{(j)} | \boldsymbol{x}_{k-1}^{(j)}, \boldsymbol{z}_k)}$.
    end for
    Normalise the weights: $w_k^{(j)} = \frac{w_k^{(j)}}{\sum_i w_k^{(i)}}$ $j = 1, \dots, N$.
    if Resampling then
      Select $N$ particle indices $j_i \in \{1, \dots, N\}$ according to weights $\{w_k^{(j)}\}_{j=1}^{N}$.
      Set $\boldsymbol{x}_k^{(i)} = \boldsymbol{x}_k^{(j_i)}$, and $w_k^{(i)} = 1/N \quad i = 1, \dots, N$.
    end if
    Estimate the following NPs: $\boldsymbol{\eta}_{a,d}$ and $\boldsymbol{\eta}_{b,d}$.
    Compute the NPs for sensor node $d$:
    $\boldsymbol{\eta}_d = \boldsymbol{\eta}_{a,d} - \boldsymbol{\eta}_{b,d} - \sum_{i \neq d} \boldsymbol{\eta}_i$
    Transmit the NPs for sensor node $d$ to the set $D \setminus d$ of sensor nodes.
    Receive the NPs for the set $D \setminus d$ of sensor nodes .
  end for
  $\widehat{p}_d(\boldsymbol{x}_k | \boldsymbol{z}_{1:k}) = \sum_{j=1}^{N} w_k^{(j)} \delta\left(\boldsymbol{x}_k - \boldsymbol{x}_k^{(j)}\right)$
end for
```

Chapter 6

CONCLUSIONS

The aim of this thesis was to present computationally efficient methods for object tracking in complex systems. The specific complexities considered included the tracking of an overwhelming large number of objects; complexities in the sensor characteristics; and large amounts of measurements from data rich sensors or sensor networks.

A Box PF and CPF framework for tracking a large crowd of objects was presented in Chapter 3. These methods followed a large group approach where the aim was shifted to inferring the kinematic states, measurement rates, and parameters which characterised the shape of the crowd. A theoretical derivation for the generalised likelihood function for the Box PF was presented. The likelihood was calculated based on optimisation, by solving a CSP with multiple measurements. An adaptive CPF was presented that was able to deal with multiple measurements, including a high level of clutter. The CPF was able to resolve the data association problem without the need of estimating the measurement rates. The filters adaptively tracked the envelope of a crowd. Both filters resolved the data association problem in an efficient way. The Box PF and CPF were compared with the SIR PF. The experiments showed that the Box PF and CPF require a significantly smaller number of (box) particles than the SIR PF, and are also more robust to initialisation errors.

The formulation of the crowd tracking problem has been shown to be directly related to the formulation of the extended object tracking problem. In Chapter 4, a Box PF method for multiple extended object tracking was proposed. A theoretical derivation of the generalised likelihood function of the Box PF for this case was presented. The proved equation was further modified to minimise the computational complexity. Experiments on real data from laser rangefinder sensors has shown that the Box PF can work efficiently with four to thirty two box particles, whereas the PF working with point particles needs several thousands of particles to achieve the same accuracy. The Box PF has been shown to have several advantages when compared to the BP PF. This included a significant computational gain, more than 32%, which could potentially be further exploited through an implementation on a platform that is efficient in interval arithmetic. The Box PF exhibited robustness for a significantly smaller number of box particles which completely encompassed the initialisation region.

In the context of object tracking, tall data refers to an excessively large number of measurements from sensors. This causes a large computational expense for sample based methods such as SMCMC and the PF. In Chapter 5, three novel methods were developed based on the introduction of adaptive subsampling into an SMCMC framework, and EP into an SMCMC and PF framework. The power of the algorithms was displayed through three examples, with comparisons to a standard SMCMC and PF algorithms. The proposed algorithms greatly reduced the computational time for Bayesian filtering, by up to 90% in the conducted experiments. The adaptive subsampling approach afforded a computational gain by only evaluating the likelihood function for a subset of the measurements. In contrast, the EP based algorithms were afforded a computational gain through a divide and conquer approach. Both approaches had flexible structures. The EP based methods had several advantages when used in an interconnected sensor network including: *i*) they did not rely on a synchronous random number generator; *ii*) they were scalable to any sized interconnected network of sensor nodes; *iii*) they were capable of intuitively integrating measurement information in the proposal distribution; *iv*) the EP based framework allowed for an approximation of the filtering posterior state pdf at every sensor node in the network; and *v*) they were well suited to handle large volumes of measurements due to significantly reducing communication costs. The experiments illustrated that the EP based methods had up to a 93% reduction in communication costs when compared with a centralised PF framework.

6.1 Directions for Future Work

Object tracking in complex systems remains a challenging task with a large scope for future research. Below is a non-exhaustive list of directions for future research based on the findings in this thesis:

- The focus of this thesis has been on object tracking, however, state space models are capable of representing a wide variety of dynamic systems. Recently, several emerging complex systems have become of interest. This includes smart cities [70], traffic mobility over large scale traffic networks [85], satellite image data [63] etc. Future work could be focused on extending the methods presented in this thesis to these interesting applications.

- The Box PF based methods have been shown to require a significantly lower number of particles to represent the posterior state pdf. However, this reduction in particles has not reduced the computational complexity of the algorithms by the same order of magnitude. This is in part due to the additional interval arithmetic operations, but also due to the use of an inefficient MATLAB toolbox for these operations, INTLAB [101]. INTLAB was originally designed and optimised for estimating rounding errors. Development of a MATLAB toolbox optimised for the interval operations utilised in the Box PF would aid future development of Box PF based methods.

- A unique aspect of the Box PF method is the requirement of contracting the box particle states given the measurements. In Chapters 3 and 4, this was done through the definition of a CSP given by an individual measurement. The contracted box particles for each measurement were then combined to obtain a box particle contracted by all the measurements. An alternative approach would be to consider the constraints from the individual measurements jointly, as a single CSP. However, measurements not originating from the object would lead to a violation of the constraints, resulting in no solution for a standard CSP. Future research could consider a flexible CSP, where certain constraints may be relaxed, such as a weighted CSP [43]. It would also be interesting to see the relation between this proposed approach and the approach presented in this thesis.

- In the adaptive subsampling based method in Section 5.2, future work could be focused on further refining the method for object tracking. This could be achieved by exploring non-uniform subsampling of the measurements, and further theoretical development of the proxy term for the likelihood.

- As stated in Sections 5.3 and 5.4, the performance of the EP based methods were linked to an approximation of the likelihood function at each local processing node by a distribution which was a member of the exponential family. Although this includes a wide variety of distributions, there may be scenarios where this is a poor approximation. Future work could focus on relaxing the restriction of the exponential family of distributions by reformulating the problem with a Gaussian mixture distribution.

Appendix A

LIKELIHOOD FUNCTION DERIVATION AND CLUTTER RATE ESTIMATION FOR CROWD TRACKING

In Section 3.4 the following approximation is presented:

$$\int U_{[\boldsymbol{z}_{m,k}]}\left(\tilde{h}\left(\boldsymbol{y}_{m,k}\right)\right) U_{q(\boldsymbol{x}_k)}\left(\boldsymbol{y}_{m,k}\right) d\boldsymbol{y}_{m,k} \approx U_{r(\boldsymbol{x}_k)}\left(\boldsymbol{z}_{m,k}\right). \tag{A.1}$$

In this Appendix a detailed description supporting this approximation is presented.

In order to evaluate the integral, it is required to transform the domain of the uniform distribution relating a measurement to a measurement source. The explicit expression for the pdf of this distribution is given by:

$$U_{[\boldsymbol{z}_{m,k}]}\left(\tilde{h}\left(\boldsymbol{y}_{m,k}\right)\right) = \begin{cases} \frac{1}{6\boldsymbol{\sigma}} & : \tilde{h}\left(\boldsymbol{y}_{m,k}\right) \in \left[\boldsymbol{z}_{m,k} - 3\boldsymbol{\sigma}, \boldsymbol{z}_{m,k} + 3\boldsymbol{\sigma}\right] \\ 0 & : \text{elsewhere} \end{cases} \tag{A.2}$$

The inverse function of $\tilde{h}(\,\cdot\,)$ is defined as $\tilde{h}^{-1}(\,\cdot\,)$. When the inverse function exists, a change of variable can be straightforwardly made that results in:

$$g\left(\boldsymbol{y}_{m,k}\right) = \begin{cases} \frac{1}{6\boldsymbol{\sigma}} \left| \frac{d\left(\tilde{h}\left(\boldsymbol{y}_{m,k}\right)\right)}{d\boldsymbol{y}_{m,k}} \right| & : \boldsymbol{y}_{m,k} \in \mathcal{X} \\ 0 & : \text{elsewhere} \end{cases} \tag{A.3}$$

where $\mathcal{X} = [\tilde{h}^{-1}(\boldsymbol{z}_{m,k} - 3\boldsymbol{\sigma}), \tilde{h}^{-1}(\boldsymbol{z}_{m,k} + 3\boldsymbol{\sigma})]$. Thus the integral in equation (A.1) is directly solvable in the following form:

$$\begin{aligned}\int U_{[\boldsymbol{z}_{m,k}]}\left(\tilde{h}\left(\boldsymbol{y}_{m,k}\right)\right) U_{q(\boldsymbol{x}_k)}\left(\boldsymbol{y}_{m,k}\right) d\boldsymbol{y}_{m,k} & \\ = \int g\left(\boldsymbol{y}_{m,k}\right) U_{q(\boldsymbol{x}_k)}\left(\boldsymbol{y}_{m,k}\right) d\boldsymbol{y}_{m,k} & \qquad \text{(A.4)}\end{aligned}$$

It is worth noting that after the transformation, the expression in (A.3) is not necessarily uniform.

A.1 The Linear Case

For the linear observation model, given in (3.11), the expression in (A.3) remains uniform:

$$g\left(\boldsymbol{y}_{m,k}\right) = \begin{cases} \frac{1}{6\boldsymbol{\sigma}} & : \boldsymbol{y}_{m,k} \in [\boldsymbol{z}_{m,k} - 3\boldsymbol{\sigma}, \boldsymbol{z}_{m,k} + 3\boldsymbol{\sigma}] \\ 0 & : \text{elsewhere} \end{cases} \qquad \text{(A.5)}$$

The range of the uniform distribution is dependent on the noise characteristics of the sensor. The range of the second uniform distribution, $U_{q(\boldsymbol{x}_k)}\left(\boldsymbol{y}_{m,k}\right)$, in (A.1) is dependent on the extent parameters of the crowd. The integral in (A.4) only exists when the two uniform distributions overlap. The overlapping region is defined by:

$$r(\boldsymbol{x}_k) = \begin{cases} x_k - \frac{a_k}{2} - 3\sigma_1 \leq z_{1,m,k} \leq x_k + \frac{a_k}{2} + 3\sigma_1 \\ y_k - \frac{b_k}{2} - 3\sigma_2 \leq z_{2,m,k} \leq y_k + \frac{b_k}{2} + 3\sigma_2. \end{cases} \qquad \text{(A.6)}$$

The approximation in (A.1) is based upon the assumption that the length of the extent is significantly larger than the sensor noise characteristics. For instance, the case when the extent tends towards an infinite length with fixed sensor noise is also equivalent to an extent with a fixed length size, and with a sensor noise tending towards zero. In this case the uniform distribution in (A.5) tends towards the Dirac delta function, i.e.

$$g\left(\boldsymbol{y}_{m,k}\right) = \begin{cases} +\infty & : \boldsymbol{y}_{m,k} = \boldsymbol{z}_{m,k} \\ 0 & : \text{elsewhere} \end{cases} \qquad \text{(A.7)}$$

Consequently resulting in equivalence in (A.1):

$$\int g\left(\boldsymbol{y}_{m,k}\right) U_{q(\boldsymbol{x}_k)}\left(\boldsymbol{y}_{m,k}\right) d\boldsymbol{y}_{m,k} = U_{r(\boldsymbol{x}_k)}\left(\boldsymbol{z}_{m,k}\right). \tag{A.8}$$

In reality, the extent is not infinite, however in general, it is considered significantly larger than the range of the sensor noise. This is the motivating factor for the result in (A.1).

A.2 The Non-linear Case

A toy example is presented to illustrate the effect of a non-linear relationship between the sensor and a measurement source. Considering a single dimension with the following relationship,

$$z_{m,k} = \tilde{h}(x_{m,k}) + \xi_{1,k} = \left(x_{m,k}\right)^2 + \xi_{1,k}, \tag{A.9}$$

results in the following transformation:

$$g\left(x_{m,k}\right) = \begin{cases} \frac{1}{2\boldsymbol{\sigma}} x_{m,k} & : x_{m,k} \in [\sqrt{z_{m,k} - 3\sigma}, \sqrt{z_{m,k} + 3\sigma}] \\ 0 & : \text{elsewhere} \end{cases} \tag{A.10}$$

In this case, the function $g\left(x_{m,k}\right)$ is clearly no longer uniform. An example of the solution of the integration in (A.4) is illustrated in Figure A.1a.

This example illustrates that although non-linearities may result in the non-uniformity of $g(x)$, when the extent parameters are significantly larger than the measurement error noise, a uniform approximation for equation (A.1) may still be valid. The effect of greater non-linearities is a topic for future research.

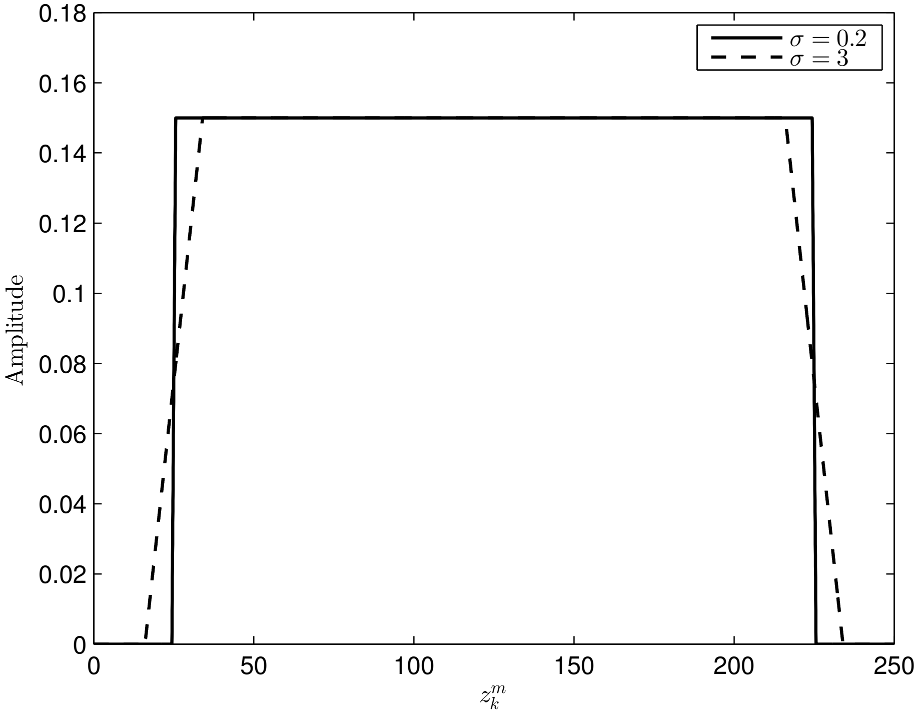


Figure A.1: Example solution of equation (A.1) with $x_k = 10$ and $a_k = 10$ with varying σ.

A.3 Crowd and Clutter Measurement Rate Estimation

The clutter rate λ_C and the crowd rate λ_T can be updated based on the assumption that they can be drawn from the Gamma distribution, similarly to [59]. For λ_C

$$\begin{aligned} p(\lambda_C|\boldsymbol{Z}_k) = \mathcal{GAM}(\lambda_C; \alpha^C_{k|k}, \beta^C_{k|k}) \\ \times \mathcal{L}(\alpha^C_{k|k-1}, \beta^C_{k|k-1}, M_{C,k}), \end{aligned} \tag{A.11}$$

and the updated parameters of the Gamma distribution for the clutter measurement rate are:

$$\begin{aligned} \alpha^C_{k|k} &= \alpha^C_{k|k-1} + M_{C,k}, \\ \beta^C_{k|k} &= \beta^C_{k|k-1} + 1. \end{aligned} \tag{A.12}$$

For λ_T the same relations as (A.12) are valid for the Gamma distribution parameters. In the Box PF implementation, $M_{T,k} = \min_p \left|\mathcal{S}_E^{(p)}\right|$ and for the clutter measurement rate, $M_{C,k} = M_k - \ M_{T,k}$.

Appendix B

EXPANDED GENERALISED LIKELIHOOD FOR MULTIPLE EXTENDED OBJECT TRACKING

According to (4.21), given $\mathcal{J}$, for any value of M_k:

$$\prod_{m=1}^{M_k}\left(\rho+\sum_{i\in\mathcal{J}}\lambda_{T,i}p(\boldsymbol{z}_{m,k}|\boldsymbol{x}_{i,k})\right)=\rho^{M_k}+\sum_{m=1}^{M_k}\sum_{j=1}^{\binom{M_k}{m}}\sum_{n=1}^{|\mathcal{J}|^m}\rho^{M_k-m}\times$$
$$\prod_{\ell=1}^{m}\lambda_{T,(\boldsymbol{b}_{m,n})_\ell}p(\boldsymbol{z}_{(\boldsymbol{a}_{m,j})_\ell,k}|\boldsymbol{x}_{(\boldsymbol{b}_{m,n})_\ell,k}). \tag{B.1}$$

To simplify notations, define: $c_{m,i} = p(\boldsymbol{z}_{m,k}|\boldsymbol{x}_{i,k})$ and $\mathcal{C}(c_{1,\mathcal{J}}, c_{2,\mathcal{J}}, ..., c_{M_k,\mathcal{J}}; \psi)$ represents the summation of all ψ unique combinations of $c_{m,i}$ terms multiplied by the associated object densities, with $\mathcal{C}(c_{1,\mathcal{J}}, c_{2,\mathcal{J}}, ..., c_{M_k,\mathcal{J}}; 0) = 1$, $\mathcal{C}(c_{1,\mathcal{J}}, c_{2,\mathcal{J}}, ..., c_{M_k,\mathcal{J}}; -1) = 0$, and $\mathcal{C}(c_{1,\mathcal{J}}, c_{2,\mathcal{J}}, ..., c_{M_k,\mathcal{J}}; M_k + 1) = 0$. For example, if $\mathcal{J} = \{1, 2\}$, then $\mathcal{C}(c_{1,\mathcal{J}}, c_{2,\mathcal{J}}, c_{3,\mathcal{J}}; 2) = \lambda_{T,1}^2 c_{1,1}c_{2,1} + \lambda_{T,1}\lambda_{T,2}c_{1,1}c_{2,2} + \lambda_{T,1}\lambda_{T,2}c_{1,2}c_{2,1} + \lambda_{T,2}^2 c_{1,2}c_{2,2} + \lambda_{T,1}^2 c_{2,1}c_{3,1} + \lambda_{T,1}\lambda_{T,2}c_{2,1}c_{3,2} + \lambda_{T,1}\lambda_{T,2}c_{2,2}c_{3,1} + \lambda_{T,2}^2 c_{2,2}c_{3,2} + \lambda_{T,1}^2 c_{1,1}c_{3,1} + \lambda_{T,1}\lambda_{T,2}c_{1,1}c_{3,2} + \lambda_{T,1}\lambda_{T,2}c_{1,2}c_{3,1} + \lambda_{T,2}^2 c_{1,2}c_{3,2}$. A useful decomposition of the expression in this form is: $\mathcal{C}(c_{1,\mathcal{J}}, c_{2,\mathcal{J}}, ..., c_{M_k+1,\mathcal{J}}; \psi) = \mathcal{C}(c_{1,\mathcal{J}}, c_{2,\mathcal{J}}, ..., c_{M_k,\mathcal{J}}; \psi) + \sum_{i\in\mathcal{J}} \lambda_{T,i}c_{M_k+1,i}\mathcal{C}(c_{1,\mathcal{J}}, c_{2,\mathcal{J}}, ..., c_{M_k,\mathcal{J}}; \psi - 1)$. The compact form of equation (B.1) is

$$\prod_{m=1}^{M_k}\left(\rho+\sum_{i\in\mathcal{J}}\lambda_{T,i}c_{m,i}\right)=\sum_{m=0}^{M_k}\rho^{M_k-m}\mathcal{C}(c_{1,\mathcal{J}}, c_{2,\mathcal{J}}, ..., c_{M_k,\mathcal{J}}; m). \tag{B.2}$$

Base case: $M_k = 1$:

$$\begin{aligned}\rho + \sum_{i \in \mathcal{I}} \lambda_{T,i} c_{1,i} &= \sum_{m=0}^{1} \rho^{1-m} \mathcal{C}(c_{1,\mathcal{I}}; m) \\ &= \rho^{1-0} \mathcal{C}(c_{1,\mathcal{I}}; 0) + \rho^{1-1} \mathcal{C}(c_{1,\mathcal{I}}; 1) \\ &= \rho + \sum_{i \in \mathcal{I}} \lambda_{T,i} c_{1,i}. \end{aligned} \tag{B.3}$$

Inductive hypothesis: Suppose equation (B.2) holds for all values of M_k.

Inductive step: Consider the scenario with $M_k + 1$ measurements,

$$\begin{aligned}&\prod_{m=1}^{M_k+1} \left(\rho + \sum_{i \in \mathcal{I}} \lambda_{T,i} c_{m,i} \right) = \\ &\quad \left[\prod_{m=1}^{M_k} \left(\rho + \sum_{i \in \mathcal{I}} \lambda_{T,i} c_{m,i} \right) \right] \left(\rho + \sum_{i \in \mathcal{I}} \lambda_{T,i} c_{M_k+1,i} \right) \\ &\quad = \sum_{m=0}^{M_k} \rho^{M_k - m} \mathcal{C}(c_{1,\mathcal{I}}, c_{2,\mathcal{I}}, ..., c_{M_k,\mathcal{I}}; m) \left(\rho + \sum_{i \in \mathcal{I}} \lambda_{T,i} c_{M_k+1,i} \right) \\ &\quad = \sum_{m=0}^{M_k+1} \rho^{M_k+1-m} \mathcal{C}(c_{1,\mathcal{I}}, c_{2,\mathcal{I}}, ..., c_{M_k,\mathcal{I}}; m) \\ &\quad + \sum_{i \in \mathcal{I}} \lambda_{T,i} c_{M_k+1,i} \sum_{m=0}^{M_k+1} \rho^{M_k+1-m} \mathcal{C}(c_{1,\mathcal{I}}, c_{2,\mathcal{I}}, ..., c_{M_k,\mathcal{I}}; m-1) \\ &\quad = \sum_{m=0}^{M_k+1} \rho^{M_k+1-m} \Big(\mathcal{C}(c_{1,\mathcal{I}}, c_{2,\mathcal{I}}, ..., c_{M_k,\mathcal{I}}; m) \\ &\quad + \sum_{i \in \mathcal{I}} \lambda_{T,i} c_{M_k+1,i} \mathcal{C}(c_{1,\mathcal{I}}, c_{2,\mathcal{I}}, ..., c_{M_k,\mathcal{I}}; m-1) \Big) \\ &\quad = \sum_{m=0}^{M_k+1} \rho^{M_k+1-m} \mathcal{C}(c_{1,\mathcal{I}}, c_{2,\mathcal{I}}, ..., c_{M_k+1,\mathcal{I}}; m). \end{aligned} \tag{B.4}$$

By the principle of mathematical induction, the proposition holds for all $M_k \in \mathbb{N}$.

Appendix C

HESSIAN OF THE LOG LIKELIHOOD FOR OBJECT TRACKING IN TALL DATA

The AS-SMCMC algorithm requires an upper bound on the range of the log of the single measurement likelihood ratio, as described in equations (5.13) and (5.14). The upper bound is dependent on the Hessian of the log of the single measurement likelihood. In the examples exhibited in Section 5.5, the upper bound is independent of the data and is computed offline prior to tracking.

In Example 1 the likelihood for the ith measurement is given by:

$$p(\boldsymbol{z}_{i,k}|\boldsymbol{x}_k) = \mathcal{N}\left(\boldsymbol{z}_{i,k}; \boldsymbol{H}\boldsymbol{x}_k, \boldsymbol{R}\right). \tag{C.1}$$

with corresponding log likelihood:

$$\ell_i(\boldsymbol{x}_k) = -\frac{1}{2}\log(|\boldsymbol{R}|) - \frac{N_d}{2} log(2\pi) - \frac{1}{2}(\boldsymbol{z}_{i,k} - \boldsymbol{H}\boldsymbol{x}_k)^\top \boldsymbol{R}^{-1}(\boldsymbol{z}_{i,k} - \boldsymbol{H}\boldsymbol{x}_k). \tag{C.2}$$

The gradient of the log likelihood is:

$$\nabla\ell_i(\boldsymbol{x}_k) = \boldsymbol{H}^\top \boldsymbol{R}^{-1}(\boldsymbol{z}_{i,k} - \boldsymbol{H}\boldsymbol{x}_k). \tag{C.3}$$

Finally, the Hessian of the log likelihood is given by:

$$\nabla^2\ell_i(\boldsymbol{x}_k) = \boldsymbol{H}^\top \boldsymbol{R}^{-1}\boldsymbol{H}, \tag{C.4}$$

In Example 2, the likelihood for the ith measurement is given by:

$$p(\boldsymbol{z}_{i,k}|\boldsymbol{x}_k) = \lambda_C p_C(\boldsymbol{z}_{i,k}) + \sum_{j=1}^{N_T} \lambda_X p_X(\boldsymbol{z}_{i,k}|\boldsymbol{x}_{j,k}). \tag{C.5}$$

with corresponding log likelihood:

$$\ell_i(\boldsymbol{x}_k) = \log\left(\lambda_C p_C(\boldsymbol{z}_{i,k}) + \sum_{j=1}^{N_T} \lambda_X p_X(\boldsymbol{z}_{i,k}|\boldsymbol{x}_{j,k})\right). \tag{C.6}$$

The gradient of the log likelihood is:

$$\nabla\ell_i(\boldsymbol{x}_k) = \frac{\lambda_X \nabla p_X}{\left(\lambda_C p_C(\boldsymbol{z}_{i,k}) + \sum_{j=1}^{N_T} \lambda_X p_X(\boldsymbol{z}_{i,k}|\boldsymbol{x}_{j,k})\right)}. \tag{C.7}$$

where $\nabla p_X = (\nabla p_X(\boldsymbol{z}_{i,k}|\boldsymbol{x}_{1,k}), ..., \nabla p_X(\boldsymbol{z}_{i,k}|\boldsymbol{x}_{N_T,k}))^\top$. Finally, the Hessian of the log likelihood is given by:

$$\nabla^2\ell_i(\boldsymbol{x}_k) = \frac{\left(\lambda_C p_C(\boldsymbol{z}_{i,k}) + \sum_{j=1}^{N_T} \lambda_X p_X(\boldsymbol{z}_{i,k}|\boldsymbol{x}_{j,k})\right) \lambda_X \nabla p_X + \lambda_X^2 \nabla p_X (\nabla p_X)\top}{\left(\lambda_C p_C(\boldsymbol{z}_{i,k}) + \sum_{j=1}^{N_T} \lambda_X p_X(\boldsymbol{z}_{i,k}|\boldsymbol{x}_{j,k})\right)^2}. \tag{C.8}$$

Appendix D

EXPECTATION PROPAGATION

This Appendix outlines the EP algorithm which forms a crucial part of the framework for the EP-SMCMC and EP-PF filters presented in Chapters 5.3 and 5.4, respectively.

Consider an intractable density $f(\boldsymbol{x})$ which can be factorised up to proportion

$$f(\boldsymbol{x}) \propto \prod_{k=0}^{K} f_k(\boldsymbol{x}). \tag{D.1}$$

The EP algorithm addresses the problem of using a tractable density to approximate $f(\boldsymbol{x})$ [86]. The approximation takes the same factorised form,

$$g(\boldsymbol{x}) \propto \prod_{k=0}^{K} g_k(\boldsymbol{x}), \tag{D.2}$$

where each factor $g_k(\boldsymbol{x})$ corresponds to the factor $f_k(\boldsymbol{x})$ in (D.1). To ensure that the EP is computationally tractable, the approximating class of densities is constrained to the exponential family,

$$g_k(\boldsymbol{x}) = h(\boldsymbol{x})g(\boldsymbol{\eta}) \exp\left\{\boldsymbol{\eta}^\top \boldsymbol{u}(\boldsymbol{x})\right\}, \tag{D.3}$$

where $\boldsymbol{\eta}$ represents the natural parameters, $\boldsymbol{u}(\boldsymbol{x})$ is a function which varies depending on the member of the exponential family, and $g(\boldsymbol{\eta})$ is a coefficient that ensures that the distribution is normalised. This constraint means that the overall approximation, $g(\boldsymbol{x})$, will also be from the same exponential family, described by a set of sufficient statistics.

The EP algorithm begins by initialising the terms, $\{g_k(\boldsymbol{x})\}_{k=0}^{K}$. The initial estimate for the approximation is thus given by (D.2). Until convergence, choose a factor to update, $g_i(\boldsymbol{x})$, $i \in \{0, ..., K\}$. Remove the factor from (D.2) according to:

$$g^{\setminus i}(\boldsymbol{x}) \propto \frac{g(\boldsymbol{x})}{g_i(\boldsymbol{x})}. \tag{D.4}$$

The parameters of the factor are then updated based on the minimisation of the following KL divergence:

$$g_i^{\text{new}}(\boldsymbol{x}) = \arg\min \mathrm{KL}(g^{\setminus i}(\boldsymbol{x}) f_k(\boldsymbol{x}) || g^{\setminus i}(\boldsymbol{x}) g_i(\boldsymbol{x})). \tag{D.5}$$

There are no guarentees of convergence for the EP iterations. However, the algorithm has been successfully applied in many applications for models with log-concave factors, $f_k(\boldsymbol{x})$ and initialisation to the prior distribution [45].

REFERENCES

[1] F. Abdallah, A. Gning, and P. Bonnifait. Box particle filtering for nonlinear state estimation using interval analysis. *Automatica*, 44(3):807 – 815, 2008.

[2] M. Ades and P. J. van Leeuwen. The equivalent-weights particle filter in a high-dimensional system. *Quarterly J. Royal Met. Soc.*, 141(687):484–503, 2015.

[3] S. Ali, K. Nishino, D. Manocha, and M. Shah, Eds. *Modeling, Simulation and Visual Analysis of Crowds*. Springer, 2014.

[4] D. Alspach and H. Sorenson. Nonlinear Bayesian estimation using Gaussian sum approximation. *IEEE Trans. Aut. Contr.*, 17(4):439?448, 1972.

[5] C. Andrieu, P. Djuric, and A. Doucet. Model selection by MCMC computation. *Signal Processing*, 81(1):19 – 37, 2001. Special section on Markov Chain Monte Carlo (MCMC) Methods for Signal Processing.

[6] C. Andrieu and G. Roberts. The Pseudo-Marginal Approach for Efficient Monte Carlo Computations. *The Annals of Statistics*, 37(2):pp. 697–725, 2009.

[7] D. Angelova and L. Mihaylova. Extended Object Tracking Using Monte Carlo Methods. *IEEE Trans. on Signal Processing*, 56(2):825–832, Feb. 2008.

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

[9] J.-Y. Audibert, R. Munos, and C. Szepesvári. Exploration-exploitation tradeoff using variance estimates in multi-armed bandits. *Theoretical Computer Science*, 410(19):1876 – 1902, 2009.

[10] Y. Bar-Shalom and T. Fortmann. *Tracking and Data Association*, volume 179 of *Mathematics in Science and Engineering*. Academic Press Professional, Inc., 1987.

[11] Y. Bar-Shalom, X.-R. Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software*. Wiley, John & Sons, 2001.

[12] R. Bardenet, A. Doucet, and C. Holmes. Towards scaling up Markov chain Monte Carlo: an adaptive subsampling approach. In *Proc. of the Int. Conf. on Machine Learning*, pages 405–413, 2014.

[13] R. Bardenet, A. Doucet, and C. Holmes. Markov chain Monte Carlo and tall data. *preprint, http://arxiv.org/abs/1505.02827*, May 2015.

[14] R. Bardenet and O.-A. Maillard. Concentration inequalities for sampling without replacement. *Bernoulli*, 21(3):1361–1385, 2015.

[15] M. Baum and U. Hanebeck. Shape tracking of extended objects and group targets with star-convex RHMs. In *Proc. of the 14th Int. Conf. on Information Fusion*, 2011.

[16] M. Baum and U. D. Hanebeck. Extended Object Tracking Based on Set-Theoretic and Stochastic Fusion. *IEEE Trans. on Aerospace and Electronic Systems*, 48(4):3103–3115, Oct. 2012.

[17] M. Baum and U. D. Hanebeck. Extended object tracking with random hypersurface models. *IEEE Trans. on Aerospace and Electronic Systems*, 50(1):149–159, Jan. 2014.

[18] M. Beard, S. Reuter, K. Granström, B. T. Vo, B. N. Vo, and A. Scheel. Multiple extended target tracking with labeled random finite sets. *IEEE Trans. on Signal Processing*, 64(7):1638–1653, April 2016.

[19] A. Beskos, D. Crisan, A. Jasra, K. Kamatani, and Y. Zhou. A Stable Particle Filter in High-Dimensions. *preprint, http://arxiv.org/abs/1412.3501*, 2014.

[20] M. Betancourt. A general Metric for Riemannian Manifold Hamiltonian Monte Carlo. *Lecture Notes in Computer Science, Geometric Science of Information, Springer*, 8085(327-334), 2013.

[21] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag New York, 2006.

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

[23] M. Bocquel, H. Driessen, and A. Bagchi. Multitarget tracking with IP reversible jump MCMC-PF. In *Proc. of the 16th Int. Conf. on Information Fusion*, pages 556–563, July 2013.

[24] M. Bocquel, F. Papi, M. Podt, and H. Driessen. Multitarget Tracking With Multiscan Knowledge Exploitation Using Sequential MCMC Sampling. *J. Sel. Topics Signal Processing*, 7(3):532–542, 2013.

[25] M. Bocquet, C. A. Pires, and L. Wu. Beyond Gaussian statistical modeling in geophysical data assimilation. *Monthly Weather Review*, 138(8):2997–3023, 2010.

[26] Y. Boers, H. Driessen, J. Torstensson, M. Trieb, R. Karlsson, and F. Gustafsson. Track-before-detect algorithm for tracking extended targets. *IEE Proc. Radar, Sonar and Navigation*, 153(4):345–351, Aug. 2006.

[27] M. Bolic, P. Djuric, and S. Hong. Resampling algorithms and architectures for distributed particle filters. *IEEE Trans. on Signal Processing*, 53(7):2442–2450, July 2005.

[28] P. Bunch and S. Godsill. Approximations of the Optimal Importance Density using Gaussian Particle Flow Importance Sampling. *preprint, http://arxiv.org/abs/1406.3183*, Nov. 2014.

[29] F. Campillo and V. Rossi. Convolution particle filter for parameter estimation in general state-space models. *IEEE Trans. on Aerospace and Electronic Systems*, 45(3):1063 –1072, July 2009.

[30] O. Cappe, S. Godsill, and E. Moulines. An Overview of Existing Methods and Recent Advances in Sequential Monte Carlo. *Proc. IEEE*, 95(5):899 –924, May 2007.

[31] A. Carmi, F. Septier, and S. Godsill. The Gaussian mixture MCMC particle algorithm for dynamic cluster tracking. *Automatica*, 48(10):2454–2467, 2012.

[32] D. Clark and S. Godsill. Group Target Tracking with the Gaussian Mixture Probability Hypothesis Density Filter. In *Int. Conf. on Intelligent Sensors, Sensor Networks and Information*, pages 149–154, Dec. 2007.

[33] F. Cucker and S. Smale. Emergent behavior in flocks. *IEEE Trans. on Automatic Control*, 52(5):852–862, May 2007.

[34] F. Daum and J. Huang. Particle flow for nonlinear filters with log-homotopy. In *SPIE Defense and Security Symposium*, pages 696918–696918. International Society for Optics and Photonics, 2008.

[35] F. Daum and J. Huang. Particle degeneracy: root cause and solution. *Proc. SPIE*, 2011.

[36] P. Del Moral, A. Doucet, and A. Jasra. An adaptive sequential Monte Carlo method for approximate Bayesian computation. *Statistics and Computing*, 22(5):1009–1020, 2012.

[37] R. Deriche and O. Faugeras. Tracking line segments. In O. Faugeras, editor, *Computer Vision ECCV 90*, volume 427 of *Lecture Notes in Computer Science*, pages 259–268. Springer Berlin Heidelberg, 1990.

[38] A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. New York: Spring-Verlag, 2001.

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

[40] O. Dzyubachyk, W. A. van Cappellen, J. Essers, W. J. Niessen, and E. Meijering. Advanced Level-Set-Based Cell Tracking in Time-Lapse Fluorescence Microscopy. *IEEE Trans. on Medical Imaging*, 29(3):852–867, March 2010.

[41] B. Errasti-Alcala and P. Braca. Track before Detect algorithm for tracking extended targets applied to real-world data of X-band marine radar. In *Proc. of 17th Int. Conf. on Information Fusion*, pages 1–8, July 2014.

[42] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. of the 2nd Int. Conf. on Knowledge Discovery and Data Mining*, pages 226–231, 1996.

[43] J. E. Gallardo, C. Cotta, and A. J. Fernández. Solving Weighted Constraint Satisfaction Problems with Memetic/Exact Hybrid Algorithms. *Journal of Artificial Intelligence Research*, 35:533–555, 2009.

[44] F. J. I. Garcia, M. Bocquel, and H. Driessen. Langevin Monte Carlo filtering for target tracking. In *Proc. of the 18th Int. Conf. on Information Fusion*, pages 82–89, July 2015.

[45] A. Gelman, A. Vehtari, P. Jylänki, C. Robert, N. Chopin, and J. P. Cunningham. Expectation propagation as a way of life. *preprint, http://arxiv.org/abs/1412.4869*, 2014.

[46] S. Geman and D. Geman. Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI-6(6):721–741, Nov. 1984.

[47] R. Genuer, J. Poggi, C. Tuleau-Malot, and N. Villa-Vialaneix. Random forests for big data. *preprint, http://arxiv.org/abs/1511.08327*, 2015.

[48] K. Gilholm, S. Godsill, S. Maskell, and D. Salmond. Poisson models for extended target and group tracking. In *Optics & Photonics 2005*, pages 59130R–59130R. International Society for Optics and Photonics, 2005.

[49] K. Gilholm and D. Salmond. Spatial distribution model for tracking extended objects. *IEE Proc. Radar, Sonar and Navigation*, 152(5):364–371, Oct. 2005.

[50] W. R. Gilks and C. Berzuini. Following a moving target-Monte Carlo inference for dynamic Bayesian models. *J. Royal Statist. Soc.: Series B*, 63(1):127–146, 2001.

[51] A. Gning, L. Mihaylova, F. Abdallah, and B. Ristic. Particle filtering combined with interval methods for tracking applications. In M. Mallick, V. Krishnamurthy, and B.-N. Vo, editors, *Integrated Tracking, Classification, and Sensor Management: Theory and Applications*, pages 43–74. John Wiley & Sons, New Jersey, USA, 2012.

[52] A. Gning, L. Mihaylova, S. Maskell, S. K. Pang, and S. Godsill. Group Object Structure and State Estimation With Evolving Networks and Monte Carlo Methods. *IEEE Trans. on Signal Processing*, 59(4):1383–1396, April 2011.

[53] A. Gning, B. Ristic, and L. Mihaylova. Bernoulli Particle/Box-Particle Filters for Detection and Tracking in the Presence of Triple Measurement Uncertainty. *IEEE Trans. on Signal Processing*, 60(5):2138–2151, May 2012.

[54] A. Gning, B. Ristic, L. Mihaylova, and F. Abdallah. An Introduction to Box Particle Filtering [Lecture Notes]. *IEEE Signal Processing Magazine*, 30(4):166–171, July 2013.

[55] S. Godsill and T. Clapp. Improvement Strategies for Monte Carlo Particle Filters. In A. Doucet, J. de Freitas, and N. Gordon, editors, *Sequential Monte Carlo Methods in Practice*. New York: Sringer-Verlag, 2001.

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

[57] K. Granström and M. Baum. Extended object tracking: Introduction, overview and applications. arXiv preprint arXiv:1604.00970, 2016.

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

[59] K. Granström and U. Orguner. A PHD Filter for Tracking Multiple Extended Targets Using Random Matrices. *IEEE Trans. on Signal Processing*, 60(11):5657–5671, Nov. 2012.

[60] K. Granström and U. Orguner. Estimation and maintenance of measurement rates for multiple extended target tracking. In *Proc. of 15th Int. Conf. on Information Fusion*, pages 2170–2176, July 2012.

[61] H. Haj Chadé, A. Gning, F. Abdallah, I. Mougharbel, and S. Julier. Non Parametric Distributed Inference in Sensor Networks Using Box Particles Messages. *Mathematics in Computer Science*, 8(3-4):455–478, 2014.

[62] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

[63] I. Hedhli, G. Moser, J. Zerubia, and S. B. Serpico. A New Cascade Model for the Hierarchical Joint Classification of Multitemporal and Multiresolution Remote Sensing Data. *IEEE Trans. on Geoscience and Remote Sensing*, 54(11):6333–6348, Nov. 2016.

[64] M. Isard and J. MacCormick. BraMBLe: a Bayesian multiple-blob tracker. In *IEEE Int. Conf. on Computer Vision*, volume 2, pages 34–41 vol.2, 2001.

[65] L. Jaulin. Nonlinear bounded-error state estimation of continuous-time systems. *Automatica*, 38(6):1079 – 1082, 2002.

[66] L. Jaulin. Robust set-membership state estimation; application to underwater robotics. *Automatica*, 45(1):202–206, 2009.

[67] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter. *Applied Interval Analysis*. Springer-Verlag, 2001.

[68] R. E. Kalman. A new approach to linear filtering and prediction problems. *Trans. of the ASME-Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[69] Z. Khan, T. Balch, and F. Dellaert. MCMC-based particle filtering for tracking a variable number of interacting targets. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27(11):1805 –1819, Nov. 2005.

[70] R. Kitchin. The real-time city? Big data and smart urbanism. *GeoJournal*, 79(1):1–14, 2014.

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

[72] A. Korattikara, Y. Chen, and M. Welling. Austerity in MCMC land: Cutting the Metropolis-Hastings Budget. In *Proc. of the Int. Conf. on Machine Learning*, 2014.

[73] J. Lan and X. Li. Tracking of Maneuvering Non-Ellipsoidal Extended Object or Target Group Using Random Matrix. *IEEE Trans. on Signal Processing*, 62(9):2450–2463, May 2014.

[74] T. Li, M. Bolic, and P. Djuric. Resampling Methods for Particle Filtering: Classification, implementation, and strategies. *IEEE Signal Processing Magazine*, 32(3):70–86, May 2015.

[75] J. S. Liu. *Monte Carlo Strategies in Scientific Computing.* New York: Springer, Jan. 2008.

[76] C. Lundquist, K. Granström, and U. Orguner. An Extended Target CPHD Filter and a Gamma Gaussian Inverse Wishart Implementation. *IEEE Journal of Selected Topics in Signal Processing*, 7(3):472–483, June 2013.

[77] R. Mahler. Multitarget Bayes filtering via first-order multitarget moments. *IEEE Trans. on Aerospace and Electronic Systems*, 39(4):1152–1178, Oct 2003.

[78] R. Mahler. "Statistics 101" for multisensor, multitarget data fusion. *IEEE Aerospace and Electronic Systems Magazine*, 19(1 II):53–64, 2004.

[79] R. Mahler. *Statistical Multisource-Multitarget Information Fusion.* Artech House, Norwood, MA, USA, 2007.

[80] R. P. S. Mahler. PHD filters for nonstandard targets, I: Extended targets. *Proc. of the 12th Int. Conf. on Information Fusion*, pages 915–921, 2009.

[81] M. Mallick, V. Krishnamurthy, and B.-N. Vo. *Integrated tracking, classification, and sensor management: theory and applications.* John Wiley & Sons, 2012.

[82] R. Mazzon and A. Cavallaro. Multi-camera tracking using a Multi-Goal Social Force Model. *Neurocomputing*, 100:41–50, 2013.

[83] K. Mehrotra and P. R. Mahapatra. A jerk model for tracking highly maneuvering targets. *IEEE Trans. on Aerospace and Electronic Systems*, 33(4):1094–1105, Oct. 1997.

[84] L. Mihaylova, A. Carmi, F. Septier, A. Gning, S. Pang, and S. Godsill. Overview of Bayesian sequential Monte Carlo methods for group and extended object tracking. *Digital Signal Processing: A Review Journal*, 25(1):1–16, 2014.

[85] L. Mihaylova, A. Hegyi, A. Gning, and R. Boel. Parallelized Particle and Gaussian Sum Particle Filters for Large-Scale Freeway Traffic Systems. *IEEE Trans. on Intelligent Transportation Systems*, 13(1):36–48, March 2012.

[86] T. P. Minka. *A family of algorithms for approximate Bayesian inference.* PhD thesis, Massachusetts Institute of Technology, 2001.

[87] S. Minsker, S. Srivastava, L. Lin, and D. Dunson. Robust and Scalable Bayes via a Median of Subset Posterior Measures. *preprint, http://arxiv.org/abs/1403.2660v2*, 2014.

[88] V. Mnih, C. Szepesvári, and J.-Y. Audibert. Empirical Bernstein Stopping. In *Proc. of the Int. Conf. on Machine Learning*, pages 672–679, 2008.

[89] W. Neiswanger, C. Wang, and E. Xing. Asymptotically Exact, Embarrassingly Parallel MCMC. *preprint, http://arxiv.org/abs/1311.4780v2*, 2013.

[90] S. K. Pang, J. Li, and S. Godsill. Detection and Tracking of Coordinated Groups. *IEEE Trans. on Aerospace and Electronic Systems*, 47(1):472–502, Jan. 2011.

[91] S. Pellegrini, A. Ess, K. Schindler, and L. V. Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *Proc. of the IEEE 12th Int. Conf. on Computer Vision*, pages 261–268, 2009.

[92] N. Petrov, L. Mihaylova, A. Gning, and D. Angelova. A novel sequential Monte Carlo approach for extended object tracking based on border parameterisation. In *Proc. of the 14th Int. Conf. on Information Fusion*, pages 306–313, 2011.

[93] M. K. Pitt and N. Shephard. Filtering via Simulation: Auxiliary Particle Filters. *J. Amer. Statist. Assoc.*, 94(446):590–599, June 1999.

[94] M. Quiroz, M. Villani, and R. Kohn. Speeding Up MCMC by Efficient Data Subsampling. *preprint, http://arxiv.org/abs/1404.4178v1*, 2014.

[95] J. Read, K. Achutegui, and J. Míguez. A distributed particle filter for nonlinear tracking in wireless sensor networks. *Signal Processing*, 98(0):121 – 134, 2014.

[96] P. Rebeschini and R. van Handel. Can local particle filters beat the curse of dimensionality? *preprint, http://arxiv.org/abs/1301.6585*, 2013.

[97] D. Reid. An algorithm for tracking multiple targets. *IEEE Trans. Auto. Control*, 24(6):843–854, Dec. 1979.

[98] C. P. Robert and G. Casella. *Monte Carlo statistical methods.* Springer, 2004.

[99] V. Rossi and J.-P. Vila. Nonlinear filtering in discrete time: A particle convolution approach. *Annales de l'Institut de Statistique de l'Université de Paris*, 50(3):71–102, 2006.

[100] D. B. Rubin. A noniterative sampling/importance resampling alternative to the data augmentation algorithm for creating a few imputations when fractions of missing information are modest: the SIR algorithm. *J. Amer. Statist. Assoc.*, 82(398):543–546, June 1987.

[101] S. Rump. INTLAB - INTerval LABoratory. In T. Csendes, editor, *Developments in Reliable Computing*, pages 77–104. Kluwer Academic Publishers, Dordrecht, 1999. `http://www.ti3.tu-harburg.de/rump/`.

[102] M. Schikora, A. Gning, L. Mihaylova, D. Cremers, and W. Koch. Box-particle PHD filter for multi-target tracking. In *Proc. of the 12th Int. Conf. on Information Fusion*, pages 106–113, 2012.

[103] M. Schikora, A. Gning, L. Mihaylova, D. Cremers, W. Koch, and R. Streit. Box-particle intensity filter for multi-target tracking. *IEEE Trans. on Signal Processing*, 50(3):1660–1672, 2014.

[104] D. Schuhmacher, B. T. Vo, and B. N. Vo. A consistent metric for performance evaluation of multi-object filters. *IEEE Trans. on Signal Processing*, 56(8):3447–3457, Aug. 2008.

[105] S. L. Scott, A. W. Blocker, F. V. Bonassi, H. A. Chipman, E. I. George, and R. E. McCulloch. Bayes and big data: The consensus Monte Carlo algorithm. In *EFaB Bayes 250 Conf.*, volume 16, 2013.

[106] F. Septier, A. Carmi, S. K. Pang, and S. Godsill. Multiple Object Tracking Using Evolutionary and Hybrid MCMC-Based Particle Algorithms. In *Proc. of the 15th IFAC Symp. on System Identification*, July 2009.

[107] F. Septier, J. Cornebise, S. Godsill, and Y. Delignon. A comparative study of Monte-Carlo methods for multitarget tracking. In *IEEE Statistical Signal Processing Workshop*, pages 205–208, June 2011.

[108] F. Septier, S. K. Pang, A. Carmi, and S. Godsill. On MCMC-Based particle methods for Bayesian filtering: Application to multitarget tracking. In *Proc. of the IEEE Int. Workshop on Computational Advances in Multi-Sensor Adaptive Processing*, pages 360–363, Dec. 2009.

[109] F. Septier and G. W. Peters. Langevin and Hamiltonian Based Sequential MCMC for Efficient Bayesian Filtering in High-Dimensional Spaces. *IEEE Journal of Selected Topics in Signal Processing*, 10(2):312–327, March 2016.

[110] R. Singer. Estimating Optimal Tracking Filter Performance for Manned Maneuvering Targets. *IEEE Trans. on Aerospace and Electronic Systems*, 6:473–483, July 1970.

[111] V. Smidl and R. Hofman. Adaptive importance sampling in particle filtering. In *Proc. of 16th Int. Conf. on Information Fusion*, pages 9–16, July 2013.

[112] C. Snyder. Particle filters, the "optimal" proposal and high-dimensional systems. In *Proc. of the ECMWF Seminar on Data Assimilation for Atmosphere and Ocean*, 2011.

[113] C. Snyder, T. Bengtsson, P. Bickel, and J. Anderson. Obstacles to high-dimensional particle filtering. *Monthly Weather Review*, 136(12):4629–4640, 2008.

[114] M. Suchard, Q. Wang, C. Chan, J. Frelinger, A. Cron, and M. West. Understanding GPU programming for statistical computation: Studies in massively parallel massive mixtures. *Journal of Computational and Graphical Statistics*, 19(2):419–438, 2010.

[115] A. ur Rehman, S. Naqvi, L. Mihaylova, and J. Chambers. Multi-target Tracking and Occlusion Handling with Learned Variational Bayesian Clusters and a Social Force Model. *IEEE Trans. on Signal Processing*, 64(5):1320 – 1335, 2016.

[116] R. Van Der Merwe, A. Doucet, N. De Freitas, and E. Wan. The unscented particle filter. In *Adv. Neural Inform. Process. Syst.*, pages 584–590, Dec. 2000.

[117] Velodyne LIDAR. *High Definition LiDAR HDL-64E S2 Specifications.* Available: http://velodynelidar.com/lidar/hdlproducts/hdl64e.aspx.

[118] J. Vermaak, N. Ikoma, and S. Godsill. Sequential Monte Carlo framework for extended object tracking. *IEE Proc. Radar, Sonar and Navigation*, 152(5):353–363, 2005.

[119] J.-P. Vila. Enhanced consistency of the Resampled Convolution Particle Filter. *Statistics and Probability Letters*, 82(4):786 – 797, 2012.

[120] B.-N. Vo and W.-K. Ma. The Gaussian Mixture Probability Hypothesis Density Filter. *IEEE Trans. on Signal Processing*, 54(11):4091–4104, Nov. 2006.

[121] B.-N. Vo, S. Singh, and A. Doucet. Sequential Monte Carlo methods for multitarget filtering with random finite sets. *IEEE Trans. on Aerospace and Electronic Systems*, 41(4):1224–1245, Oct. 2005.

[122] N. Wahlström and E. Özkan. Extended Target Tracking Using Gaussian Processes. *IEEE Trans. on Signal Processing*, 63(16):4165–4178, Aug. 2015.

[123] E. A. Wan and R. Van Der Merwe. The unscented Kalman filter for nonlinear estimation. In *IEEE Com. & Control Symp. Adaptive Systems Signal Proc.*, pages 153–158. IEEE, Oct. 2000.

[124] X. Wang and D. Dunson. Parallel MCMC via Weierstrass sampler. *preprint, http://arxiv.org/abs/1312.4605*, 2014.

[125] M. Wieneke, K. Safenreiter, and W. Koch. Combined person tracking and classification in a network of chemical sensors. In *Proc. of the 11th Int. Conf. on Information Fusion*, pages 1–8, June 2008.

[126] M. Xu, Y. W. Teh, J. Zhu, and B. Zhang. Distributed Context-Aware Bayesian Posterior Sampling via Expectation Propagation. In *Advances in Neural Information Processing Systems*, 2014.

[127] J. Yang, D. Zhang, A. Frangi, and J.-Y. Yang. Two-dimensional PCA: a new approach to appearance-based face representation and recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(1):131–137, Jan. 2004.