Faculty of Science and Engineering
Department of Electrical and Computer Engineering

Estimation and Control of Multi-object Systems with High-fidelity Sensor Models:

A Labelled Random Finite Set Approach

Michael Anthony Beard

This thesis is presented for the degree of
Doctor of Philosophy
of
Curtin University

June 2016

Michael Anthony Beard: *Estimation and Control of Multi-object Systems with High-fidelity Sensor Models,* A Labelled Random Finite Set Approach, © June 2016

SUPERVISORS: Prof. Ba-Ngu Vo, Assoc. Prof. Ba Tuong Vo, Dr. Sanjeev Arulampalam

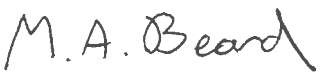
To my late father Lindsay (1949-2016),

and my mother Jennet.

Their love and support will never be forgotten.

DECLARATION

To the best of my knowledge and belief, this thesis contains no material previously published by any other person, except where due acknowledgment has been made. This thesis contains no material which has been accepted for the award of any other degree or diploma in any university.



Michael Anthony Beard

June 2016

ABSTRACT

An essential component of all multi-object estimation systems is a mathematical model that captures the expected behaviour of the received measurements. The majority of multi-object estimation algorithms proposed in the literature have focussed on a measurement model known as the *standard* or *point-detection model*. Fundamentally, this model consists of the following two assumptions; each target can generate at most one measurement, and each measurement is generated by at most one target. These assumptions lead to simplified derivation and lower computational complexity of tracking algorithms. However, most sensors do not satisfy these modelling assumptions at all times, and algorithms that are derived from the standard model may not be able to cope when faced with real world sensor data. It is therefore important to develop trackers that are not restricted to the standard model, but can instead accommodate more general models in which the aforementioned assumptions may not hold true.

Another issue in many tracking systems is that of how to control the sensor. It is often the case that sensors can be manipulated in ways that affect the quality of the observed measurements, thereby affecting the estimation performance of the system. For example, sensors can often be moved or oriented in different ways, or their operating parameters changed to influence how they detect and measure the targets. Such control is often carried out manually, relying on human judgement to provide the control decisions. On the other hand, automating these decisions offers the opportunity for improved sensor control, by accounting for the prevailing conditions in a more reliable way.

The theory of *random finite sets* (RFS) provides a mathematically rigorous framework for addressing multi-object estimation and control problems. A recently proposed technique using *labelled random finite sets* is particularly appealing, as it has led to the derivation of the first Bayes optimal multi-target tracker that can estimate the trajectories of an unknown and time-varying number of targets. This algorithm is called the *generalised labelled multi-Bernoulli* (GLMB) filter. So far, the GLMB filter has

been restricted to the standard measurement model. This dissertation addresses this limitation by generalising the GLMB filter to accommodate sensor models that do not adhere to the standard assumptions, thereby broadening its range of potential real world applications. Also presented is a new technique for multi-target sensor control, based on maximising a measure of the multi-target information gain in the context of the GLMB filter.

The first key contribution of this dissertation is the first labelled RFS-based tracker to accommodate possible merging of target measurements. This is significant for applications involving low resolution sensors that are incapable of resolving closely spaced targets into separate detections. The second contribution is the first labelled RFS-based algorithm for multiple extended target tracking, which has applications to processing high resolution sensor data containing multiple simultaneous detections per target. The third contribution is the derivation of analytical expressions for two new statistical properties of GLMB random finite sets, namely the Cauchy-Schwarz divergence and the void probability functional. Both of these properties have applications in multi-object sensor control.

ACKNOWLEDGMENTS

First of all, I would like to express my sincere appreciation and gratitude to my supervisors, Professor Ba-Ngu Vo and Associate Professor Ba Tuong Vo. It has been an absolute pleasure working with them over the past few years. Their tremendous dedication and knowledge have been an endless source of inspiration for me, and without their exceptional guidance, this work would not have come to fruition. Furthermore, their unique sense of humor has often provided me with some welcome respite from the rigours of academic study.

I am deeply indebted to Dr Sanjeev Arulampalam, my supervisor at the Defence Science and Technology Organisation (DSTO). It was his expertise that motivated me to change the course of my career, and pursue research in the area of multi-target tracking. In retrospect, this was undoubtedly one of the best decisions I have made, and I feel privileged to have Sanjeev as my supervisor, colleague, and friend.

Special thanks go to Dr Stephan Reuter and Dr Karl Granström for their valuable collaboration regarding the work on extended target tracking. I particularly thank them for their contributions to the simulation and experimental results in Chapter 4.

I gratefully acknowledge the generous financial and administrative support I have received from the Department of Electrical and Computer Engineering of Curtin University, and Maritime Division of DSTO. In particular, I am extremely thankful for the time that DSTO has allowed me to devote to my studies. I am also very appreciative of the opportunities to attend several conferences during my candidature, which has vastly enriched my experience as a PhD student.

My deep and heartfelt appreciation goes to my amazing partner Bronwyn, for all her love and devotion over the course of my studies. I will always be grateful for her enduring optimism and encouragement, which has helped me enormously during this busy period of my life. Finally, I would like to thank my wonderful parents Lindsay and Jennet for all their loving support, and for providing me with every opportunity to succeed in my education and career. I dearly wish that my dad could have lived to see me graduate, and more importantly, to enjoy his retirement. He was taken from us far too soon.

PUBLICATIONS

The following is a list of publications arising from the work in this dissertation.

JOURNAL ARTICLES

1. **M. Beard**, B.-T. Vo and B-.N. Vo. "Bayesian Multi-Target Tracking with Merged Measurements Using Labelled Random Finite Sets," *IEEE Transactions on Signal Processing*, vol. 63, no. 6, pp. 1433-1447, March 2015.

2. **M. Beard**, S. Reuter, K. Granström, B.-T. Vo, B.-N. Vo and A. Scheel. "Multiple Extended Target Tracking with Labelled Random Finite Sets," *IEEE Transactions on Signal Processing*, vol. 64, no. 7, pp. 1638-1653, April 2016.

3. **M. Beard**, B.-N. Vo, B.-T. Vo and S. Arulampalam. "Void Probabilities and Cauchy-Schwarz Divergence for Generalized Labeled Multi-Bernoulli Models," submitted to *IEEE Transactions on Information Theory* in August 2015, arXiv preprint arXiv:1510.05532.

4. F. Papi, B.-N. Vo, B.-T. Vo, C. Fantacci and **M. Beard**. "Generalized Labeled Multi-Bernoulli Approximation of Multi-Object Densities," *IEEE Transactions on Signal Processing*, vol. 63, no. 20, pp. 5487-5497, October 2015.

5. **M. Beard**, B.-T. Vo and B.-N. Vo. "Multitarget Filtering with Unknown Clutter Density Using a Bootstrap GMCPHD Filter," *IEEE Signal Processing Letters*, vol. 20, no. 4, pp. 323-326, February 2013.

6. **M. Beard**, B.-T. Vo, B.-N. Vo and S. Arulampalam, "A Partially Uniform Target Birth Model for Gaussian Mixture PHD/CPHD Filtering", *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 4, pp. 2835-2844, October 2013.

7. B. Ristic, **M. Beard**, C. Fantacci, "An Overview of Particle Methods for Random Finite Set Models," *Information Fusion*, vol. 31, pp. 110-126, September 2016.

CONFERENCE PAPERS

1. **M. Beard**, B.-T. Vo and B.-N. Vo. "Multi-target Tracking with Merged Measurements Using Labelled Random Finite Sets," in *Proceedings of the 17th International Conference on Information Fusion*, Salamanca, Spain, July 2014.

2. **M. Beard**, B.-T. Vo, B.-N. Vo and S. Arulampalam. "Gaussian Mixture PHD and CPHD Filtering with Partially Uniform Target Birth," in *Proceedings of the 15th International Conference on Information Fusion*, Singapore, July 2012.

3. **M. Beard** and S. Arulampalam. "Performance of PHD and CPHD Filtering Versus JIPDA for Bearings-only Multi-target Tracking", in *Proceedings of the 15th International Conference on Information Fusion*, Singapore, July 2012.

4. **M. Beard**, S. Reuter, K. Granström, B.-T. Vo, B.-N. Vo and A. Scheel. "A Generalised Labelled Multi-Bernoulli Filter for Extended Multi-target Tracking," in *Proceedings of the 18th International Conference on Information Fusion*, Washington DC, USA, July 2015.

5. **M. Beard**, B.-T. Vo, B.-N. Vo and S. Arulampalam. "Sensor Control for Multi-target Tracking using Cauchy-Schwarz Divergence," in *Proceedings of the 18th International Conference on Information Fusion*, Washington DC, USA, July 2015.

6. S. Reuter, **M. Beard**, K. Granström and K. Dietmayer. "Tracking Extended Targets in High Clutter Using a GGIW-LMB Filter," in *Proceedings of the 10th Workshop on Sensor Data Fusion: Trends, Solutions, Applications*, Bonn, Germany, October 2015.

7. S. Reuter, A. Scheel, **M. Beard**, K. Granström, B.-T. Vo, B.-N. Vo and K. Dietmayer. "Environment Perception for Companion Systems", in *Proceedings of the International Symposium on Companion Technology*, Ulm University, Germany, September 2015.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

LIST OF ACRONYMS

CBMeMBer	Cardinality balanced multi-object multi-Bernoulli.
CPHD	Cardinalised probability hypothesis density.
CSD	Cauchy-Schwarz divergence.
EKF	Extended Kalman filter.
FISST	Finite set statistics.
GGIW	Gamma Gaussian inverse-Wishart.
GIW	Gaussian inverse-Wishart.
GLMB	Generalised labelled multi-Bernoulli.
GM	Gaussian mixture.
GMCPHD	Gaussian mixture cardinalised probability hypothesis density.
GMPHD	Gaussian mixture probability hypothesis density.
i.i.d	Independent and identically distributed.
IPDA	Integrated probabilistic data association.
ITS	Integrated track splitting.
JIPDA	Joint integrated probabilistic data association.
JPDA	Joint probabilistic data association.
LMB	Labelled multi-Bernoulli.
MHT	Multiple hypothesis tracking.
OSPA	Optimal sub-pattern assignment.
p.g.fl.	Probability generating functional.
PDA	Probabilistic data association.
pdf	Probability density function.
PHD	Probability hypothesis density.
pmf	Probability mass function.
PMHT	Probabilistic multiple hypothesis tracking.
POMDP	Partially observed Markov decision process.
PUB	Partially uniform birth.
RFS	Random finite set.
SMC	Sequential Monte Carlo.
UKF	Unscented Kalman filter.

LIST OF SYMBOLS

x	Unlabelled vector (or scalar).
$\boldsymbol{x}$	Labelled vector.
K, W, X, Y, Z	Unlabelled random finite sets.
$\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{S}, \boldsymbol{B}$	Labelled random finite sets.
$\mathcal{L}(\cdot)$	Label set extraction function.
$\Delta(\cdot)$	Distinct label indicator function.
$\delta_X(\cdot)$	Generalised Kronecker delta function, see (151).
$1_X(\cdot)$	Set inclusion function, see (152).
$\mathbb{L}$	Label space (discrete).
$\mathbb{X}$	State space.
$\mathbb{Z}$	Measurement space.
$\mathbb{C}$	Componenent space (discrete).
$\mathbb{A}$	Action space (discrete).
$f(\cdot\|\cdot)$	Transition kernel.
$g(\cdot\|\cdot)$	Likelihood function.
$p(\cdot)$	Probability density function (vector or scalar).
$\pi(\cdot)$	Unlabelled RFS density.
$\boldsymbol{\pi}(\cdot)$	Labelled RFS density.
$P_D(\cdot)$	Detection probability.
$P_S(\cdot)$	Survival probability.
Θ	Space of mappings.
$\mathcal{P}$	Set of all partitions.
$\mathcal{P}_n$	Set of all partitions dividing a set into n subsets.
$\mathcal{U}$	A partition of a set.

$\mathcal{F}_n(\cdot)$	Set of n-element subsets.
$\langle \cdot, \cdot \rangle$	Inner product of two functions.
$[h(\cdot)]^X$	Multi-object exponential.
$\lvert X \rvert$	Cardinality of the set X.
$\otimes$	Kronecker product.
$\mathbb{N}$	Space of non-negative integers.
$\mathbb{R}^+$	Space of positive real numbers.
$\mathbb{R}^n$	Space of n-dimensional real vectors.
$\mathbb{S}^n_{++}$	Space of $n \times n$ positive definite matrices.
$\mathbb{S}^n_{+}$	Space of $n \times n$ positive semi-definite matrices.
$\mathcal{GAM}(\cdot; \alpha, \beta)$	Gamma probability density function with shape α and scale β.
$\mathcal{N}(\cdot; m, P)$	Gaussian probability density function with mean m and covariance P.
$\mathcal{PS}(\cdot; \lambda)$	Poisson probability mass function with mean λ.
$\mathcal{IW}_d(\cdot; v, V)$	Inverse Wishart distribution on $\mathbb{S}^d_{++}$, with degrees of freedom v and scale V.
I_d	Identity matrix of dimension d.
$E[\cdot]$	Expected value.
$\lVert f \rVert$	L^2-norm ($\sqrt{\langle f, f \rangle}$).

1 INTRODUCTION

1.1 A BRIEF HISTORY OF ESTIMATION AND TRACKING

Since prehistoric times, humankind has been observing the night sky, examining its objects and their behaviour in ever greater detail. The study of astronomy has had a tremendous influence on the development of human civilisation, so it is perhaps fitting that this oldest of natural sciences provided the catalyst for a branch of mathematics that continues to play such an important role in many modern technologies. In 1795, an 18-year-old by the name of Carl Fredrich Gauss, now widely considered to have been one of the greatest mathematicians of all time, faced the problem of determining the motion of celestial bodies based on data collected from telescope observations. In addressing this problem, Gauss developed a technique known as the *method of least squares*, which went on to become one of the fundamental building blocks of modern estimation theory [57, 179].

It was not until 1922 that the term "estimation" came to be be used in the statistical literature. It was first introduced by Ronald Fisher in [55], who was one of the most influential statisticians of the 20th century. Fisher's most notable contributions to estimation theory were his work on the *maximum likelihood method*, and on characterising the information content of observed signals using what became known as the *Fisher information*. His pioneering work, along with the work of his student Calyampudi Rao, led to the discovery of the *Cramer-Rao bound*, which is arguably one of the most important results in modern estimation theory, as it provides a fundamental tool for evaluating the quality of unbiased estimators.

The idea of "filtering" as a method of estimating a signal's true value in the presence of noise, was introduced in the 1940's with the work of Norbert Wiener and

Andrey Kolmogorov. A filter for estimating continuous-time stationary signals based on the minimum mean square error criterion was developed by Wiener during 1940's (published in 1949) [204], and an equivalent method for discrete-time signals was independently proposed by Kolmogorov in 1941 [109]. The *Wiener filter*, as it became known, emerged as a highly popular technique for signal noise reduction, and paved the way for the later developments in estimation and signal processing.

The methods proposed by Weiner and Kolmogorov were based on the concepts of classical control theory, making use of frequency domain techniques in order to characterise the relationship between a system's inputs and outputs. Such an approach was necessary to develop methods that were tractable for manual computation, since computers had not yet been invented. In the 1950's and 60's, researchers in control and estimation theory turned their attention towards the modelling of systems directly in the time domain, known as *state-space modelling*. Such a change, which was made possible by the development of computers, afforded far greater flexibility in system design and analysis, alleviating some of the limitations of classical techniques.

In 1960, Rudolf Kalman used the concept of state-space models to develop his celebrated technique for estimating non-stationary signals from noisy observations [100]. The *Kalman filter* is an exact solution to the Bayesian estimation problem under the assumptions that the process and measurement models are linear, and the state and measurement distributions are Gaussian. It has since been extended to handle nonlinear process and measurement models, using various approximation techniques (see for example [86, 97]).

By the early 1990's, it had become clear that there were many applications in which the modelling assumptions of the Kalman filter were too restrictive. In particular, for the nonlinear variants of the Kalman filter, it has been shown that the Gaussian assumption on the state distribution becomes increasingly erroneous as the non-linearity in the system models grows more severe. In response to this problem, an approach for handling nonlinear non-Gaussian models was developed using the concept of sequential Monte Carlo sampling [63, 50, 168], a technique which became known as *particle filtering*. Algorithms based on particle filtering demand a great deal more numerical computation than Kalman filters, however, due to the advent of powerful computing technology, and their tremendous modelling flexibility, particle filters have become an extremely popular estimation technique.

The aforementioned Weiner, Kalman and particle filters were all originally developed for the class of estimation problems in which the number of parameters to be estimated is fixed and known, and the origin of the observations is also known. For applications where there exists the potential for uncertainty in the origin of the observations, due to either false alarms or the presence of multiple targets, additional techniques must be brought to bear, in order to resolve this uncertainty. Since the 1970's, there has been a great deal of research effort devoted to solving such problems, as they are becoming increasingly important in modern signal processing and estimation systems.

For tracking a single target in the presence of false alarms, the most basic approach is called *nearest neighbour data association*. This algorithm assumes that the measurement generated by the target is the one that lies closest to the predicted estimate, and all others are ignored as false alarms. Although this makes intuitive sense, it has been shown to perform poorly in practice, especially when the false alarm rate is high [9]. An improved technique was proposed in 1975, which was called *probabilistic data association* (PDA) [11]. The PDA method avoids associating a single measurement to the target, and instead updates the estimate using a probabilistically weighted combination of all measurements received on a given scan. In 1994, an extension to PDA was proposed, which accommodates cases where the existence of the target is uncertain. This algorithm was called *integrated probabilistic data association* (IPDA) [151], and was so named for the fact that it integrates a calculation of the target's existence probability into the filtering recursion.

Perhaps unsurprisingly, multiple target tracking has undergone very similar developments to single object tracking. The simplest method for estimating multiple targets is called *global nearest neighbour* (GNN) data association, which operates by making hard decisions about the association of measurements to targets on each scan. The decisions are based on minimising a global association cost, such as the sum of the distances between predicted target states and their associated measurements. Like its single-target counterpart, GNN also performs poorly under high false alarm rates, and also when there are clusters of closely spaced targets [9]. An improved method is called *joint probabilistic data association* (JPDA) [56], which updates each target using a different weighted combination of all measurements, where the weights are the association probabilities computed jointly across all targets. Calculating these probabilities involves considering every possible joint association event,

the number of which grows exponentially as the number of measurements and targets increases. JPDA can therefore become computationally infeasible, and it also restricted in the sense that the number of targets must be fixed and known. A useful extension to JPDA for tracking a time-varying and unknown number of targets is called *joint integrated probabilistic data association* (JIPDA) [148], but this also suffers from the problem of exponential complexity. A cheaper technique is known as *linear multi-target integrated probabilistic data association* (LMIPDA) [152], which makes use of a heuristic approximation to reduce the complexity.

Perhaps the most well established technique for estimating an unknown and time-varying number of targets is called *multiple hypothesis tracking* (MHT), which was first proposed by Reid in 1979 [162]. This algorithm takes a deferred decision approach to the data association problem, by generating a potentially very large collection of tentative tracks, each representing a different association hypotheses. The algorithm then waits for more data to be collected in order to build evidence for the correct hypothesis, before making a hard decision on which one is correct. This method has gained considerable popularity in the decades since it was first proposed, and it has found widespread application in operational multi-target tracking systems. However, MHT also has some drawbacks, including the fact that implementations often require heuristic methods to reduce computational complexity, and it is difficult to establish whether MHT is Bayes optimal, in the sense of minimising the posterior Bayes risk. It is also notoriously challenging to implement, requiring a great deal of programming effort to achieve a useful system.

More recently, the concept of random finite sets (RFS), has emerged as a principled technique for tackling multi-target estimation problems in a mathematically consistent and rigorous manner [135]. Fundamentally, it is based on the idea of modelling the multi-target states, and the multi-target observations, as set-valued random variables. The framework known as finite set statistics (FISST) [62, 135], provides the mathematical tools required to work with random finite sets and to characterise their statistics. The cornerstone of this framework is a notion of multi-target density and integration that is consistent with point process theory [194]. Since its inception in the mid 1990's, FISST has been used to derive several types of multi-target filtering algorithms, such as the probability hypothesis density (PHD) filter [126], cardinalised PHD (CPHD) filter [129], multi-object multi-Bernoulli (MeMBer) filter [135],

and cardinality balanced multi-object multi-Bernoulli (CBMeMBer) filter [200]. These are briefly summarised in Chapter 2 of this dissertation.

At first, the development of RFS-based techniques was concentrated on the case of multi-target filtering without target labels. This meant that on each time step, the algorithms would produce sets of unlabelled target estimates, and there was no explicit correspondence between the estimates across time steps. For this reason, RFS methods were considered useful only for multi-target filtering, and not for multi-target tracking (i.e. they could not estimate continuous target trajectories over time). For a number of years, this was seen as a weakness of the RFS approach, and attempts at solving it were based on various heuristics and post-processing techniques. However, in 2013, a new approach was proposed, based on modelling the multi-object state as a labelled random finite set [198]. This was the first multi-target tracker to be derived based on a rigorous mathematical formulation, which is naturally capable of estimating the trajectories of an unknown and time varying number of targets, under the standard models of multi-target dynamics and multi-target observation. The goal of this dissertation is to build upon this state-of-the-art technique, by generalising it in a way that will enhance its potential for use in a wider variety of real-world multi-target tracking applications.

1.2 MOTIVATION

In recent decades, the proliferation of sensors across many areas of technology has resulted in estimation theory becoming more vital than ever before. Every year, governments, corporations and individuals around the world spend many billions of dollars on systems and technologies which, at their core, rely on the principles of estimation theory to process sensor data. Indeed, systems that monitor or interact with their environment are becoming increasingly commonplace, and they all require the use of sensors to gather the information necessary to perform their task. Applications of such systems span a diverse range of technological and scientific disciplines, of which just a few of examples are; robotics [87, 117], autonomous systems [205], surveillance [120, 25], military combat/weapons systems [29], and biomedical research [61, 182, 139].

In all of these applications, the sensors alone are usually not capable of providing measurements of sufficient quality to be used directly. Although useful information

may be present, it can easily become obscured by unwanted effects such as noise, false alarms, misdetections, and measurement origin uncertainty. To address these problems, further processing is usually required in order to distill the relevant information, and eliminate any undesirable effects corrupting the data. After such processing, the "filtered" information can be passed to the rest of the system, leading to improved reliability and performance in the desired task.

Sensors come in a wide variety of different types and capabilities, and it is unreasonable to expect a single type of observation model to capture these different capabilities. The so-called "standard" sensor model accommodates different types of observed quantities (e.g. position, bearing, range, frequency, etc.), however, that is where its flexibility ends. The standard model is limited to cases where each measurement is generated by at most one target, and each target generates at most measurement. In practice, real world sensors often violate these assumptions, leading to poor estimation performance if the observation data is processed according to this model. For example, some types of radar and sonar systems have relatively low spatial resolution, and thus cannot produce separate detections for targets that are very close together [44, 29]. On the other hand, sensors such as cameras and laser range-finders can have very high resolution, and therefore may produce multiple simultaneous detections originating from a single target [66]. In practical systems, effects such as finite sensor resolution and extended objects are unavoidable, so it is fundamentally important to investigate techniques for processing this type of data in a principled manner. This dissertation is the first to tackle this problem using a rigorous formulation based on labelled random finite sets.

In many sensing systems, performing estimation using the received data is only part of the problem. Often it is possible to exert some level of control over the system, which can potentially influence the estimation performance. A prime example of this is when a sensor is mounted on a mobile platform, such as a vehicle, ship, or aircraft. Different trajectories taken by the platform can affect the quality of the measurement data, thus planning an optimal future trajectory for the sensor platform is an important consideration. These types of control problems have been studied extensively for cases involving a single target, see for example [159, 156, 101, 175, 165]. The case of multiple targets is significantly more challenging, and although some research has been done, for example [110, 127, 167, 65, 78], it has not yet received the same level of attention as single-target sensor control. Multi-target control problems are becoming

increasingly important, particularly in areas such as autonomous systems, robotics, and military command and control [75]. In addition to the aforementioned results on improved sensor models, this dissertation establishes new tools that have clear applications to sensor control in a multi-target environment, again, using the principles of labelled random finite sets.

1.3 KEY CONTRIBUTIONS

The first contribution of this dissertation is the *first multi-target tracking algorithm based on RFS principles to accommodate merged measurements*. Sensors with low spatial resolution are often incapable of generating separate detections for targets that are closely spaced. In these cases, the sensor might only deliver a single detection that is effectively generated by a group of targets, i.e. a merged measurement. Traditional sensor models do not accommodate this effect, and instead assume that each measurement is produced by at most one target. In Chapter 3, a labelled random finite set-based tracker is derived that can handle this effect in a rigorous manner. In doing so, a method for approximating an arbitrary labelled RFS density as a generalised labelled multi-Bernoulli density is developed, which has applications beyond the problem of tracking in the presence of merged measurements. This contribution has appeared in the author's conference paper [19] and journal article [21], and partially in the co-authored journal article [157].

The second contribution is the *first algorithm based on the labelled RFS technique for tracking multiple extended targets*. Most traditional tracking algorithms model the targets as points in the state space, and the sensor models assume that each target generates a maximum of one measurement at any given time. In practice, such an assumption is often inaccurate, especially when the targets are physically large relative to the size of a sensor resolution cell. When this occurs, a single target can give rise to multiple detections simultaneously, causing standard tracking algorithms to fail. In a sense, the extended target problem can be viewed as the dual of the merged measurement problem, with the former allowing multiple measurements per target, and the latter allowing multiple targets per measurement. In Chapter 4, labelled random finite sets are used to derive an algorithm which can track an unknown and time-varying number of extended targets. This contribution has appeared in part in the author's conference paper [20], and journal article [24].

The third contribution is the *derivation of analytical forms for the Cauchy-Schwarz divergence (CSD) and void probability for generalised labelled multi-Bernoulli RFSs, and their application to multi-target sensor control*. The Cauchy-Schwarz divergence provides a measure of information divergence between two probability densities, and it is demonstrated that for two GLMBs composed of Gaussian mixture single-object densities, the CSD can be computed tractably in closed form. The void probability functional is another useful statistical tool, which, in addition to providing a complete characterisation of an RFS, also allows one to calculate the probability that any given region of the state space is empty. It is shown that in certain cases, this can also be computed analytically for GLMBs. In addition to these theoretical results, Chapter 5 presents an application of the CSD and void probability to a multi-target sensor control problem, where the goal is to plan a sensor trajectory that optimises the tracking performance, whilst maintaining a safe distance from all targets. This contribution has appeared in part in the author's conference paper [23], and at the time of writing, a journal submission on this work is currently under review [22].

Finally, this dissertation contains two supplementary contributions, which were developed during the early stages of the author's research. The first of these is a *Gaussian mixture (C)PHD filter with a partially uniform birth intensity*, which is useful for scenarios where the true birth intensity is unknown, and also for reducing the number of birth parameters that need to be specified by the user. This contribution is presented in Section 2.4.6, and has previously appeared in the the author's conference paper [16] and journal article [18]. The second is a *simple improvement to a method for CPHD filtering with unknown clutter rate*, which is useful in practice, since the true clutter rate often cannot be known in advance and may be subject to variations over time. This is presented in Section 2.4.7, and has previously appeared in the author's letter [17].

2 BACKGROUND: SINGLE AND MULTI-OBJECT BAYESIAN ESTIMATION

The aim of this chapter is to provide a general background on Bayesian estimation for single and multi-object tracking, and to motivate the use of labelled random finite sets for multi-object trajectory estimation. Section 2.1 provides a brief survey of filtering techniques for applications in which the system state and the observations can be modelled as vector-valued random variables with fixed and known dimension. Section 2.2 introduces the notion of random finite sets (RFS) as a more flexible alternative to vector-valued random variables, and highlights some common classes of RFS that can be used to model multi-object systems. Section 2.3 introduces the multi-object Bayes filter for estimating the number and states of a time-varying and unknown number of objects, which is a general but intractable recursion based on the RFS formulation. Section 2.4 presents a survey of multi-object filters based on the unlabelled RFS formulation. This section also includes two preliminary contributions, one in Section 2.4.6 concerning filtering with an unknown target birth model, and another in Section 2.4.7 which considers filtering with an unknown clutter rate. These have been previously published in the author's conference paper [16] and journal articles [18, 17]. Section 2.5 reviews existing multi-object tracking techniques based on the formulation of labelled random finite sets. Finally, Section 2.6 presents detailed pseudo-code for two labelled RFS-based filters, which form the basis for the tracking algorithms developed in the remainder of this dissertation.

2.1 BAYESIAN ESTIMATION

The simplest form of Bayesian state estimation involves cases in which the number of objects is fixed and known, and it is known which observations belong to each object. Under such conditions, it suffices to model the system and the observations as vectors with fixed dimension, and on this basis, filtering algorithms can be derived using traditional statistical techniques. The most well known of these is the Kalman filter [100], which provides an exact closed-form solution to the Bayes recursion when the system and observation models are both linear and Gaussian. Approximate methods for non-linear Gaussian models include the extended Kalman filter (EKF) [86] and unscented Kalman filter (UKF) [96], and a general approximate method for non-linear non-Gaussian models is known as the particle filter [63]. This section provides a brief overview of these techniques, since they form the basis for more general multi-object filters that can handle effects such as time-varying numbers of objects, false detections, missed detections, and measurement origin uncertainty.

2.1.1 *The Bayes Recursion*

Suppose that a system can be represented by a discrete-time state-space model, and the state at time k is completely captured by a state vector $x_k \in \mathbb{X} \subseteq \mathbb{R}^{n_x}$. The state vector evolves in time according to a Markov transition density $f_{k|k-1}(x_k|x_{k-1})$, which is the probability density of the state x_k at time k, conditioned on the event that the state at time $k-1$ was x_{k-1}. At each time k, a noisy observation process yields a measurement vector $z_k \in \mathbb{Z} \subseteq \mathbb{R}^{n_z}$, which is related to the state vector via a likelihood $g_k(z_k|x_k)$, i.e. the probability density of the measurement z_k, conditioned on the event that the true state is x_k. The collection of all measurements from time 1 up to k is denoted as $z_{1:k} = (z_1, \ldots, z_k)$.

Suppose that at time $k-1$, our knowledge of the system state, conditioned on all the measurements received so far, is captured by a probability density $p_{k-1}(x_{k-1}|z_{1:k-1})$. At time k, a new measurement z_k is received. The goal of the Bayes recursion is to fuse the information in z_k with the information contained in the previous density $p_{k-1}(x_{k-1}|z_{1:k-1})$, to yield a new density $p_k(x_k|z_{1:k})$, known as the posterior density.

The posterior density is computed from the previous density, transition density and measurement likelihood as follows

$$p_{k|k-1}\left(x_k|z_{1:k-1}\right) = \int f_{k|k-1}\left(x_k|x_{k-1}\right) p_{k-1}\left(x_{k-1}|z_{1:k-1}\right) dx_{k-1} \tag{1}$$

$$p_k\left(x_k|z_{1:k}\right) = \frac{g_k\left(z_k|x_k\right) p_{k|k-1}\left(x_k|z_{1:k-1}\right)}{\int g_k\left(z_k|x\right) p_{k|k-1}\left(x|z_{1:k-1}\right) dx}. \tag{2}$$

In the above, (1) is known as the Chapman-Kolmogorov equation, (2) is called Bayes rule, and collectively they are referred to as the Bayes recursion. Using this technique, the probability density of the state can be recursively propagated in time, and new measurement data can be incorporated as it becomes available. Except for certain specific cases, the Bayes recursion does not have an analytical solution. Hence, approximations are usually required to facilitate the development of analytically and computationally tractable Bayesian filtering algorithms.

2.1.2 *Linear Gaussian Filtering*

The special case of the Bayes recursion involving a Gaussian filtering density, and linear-Gaussian system dynamics and observations, is known as the Kalman Filter [100, 76, 86, 178, 1]. Let us assume that the posterior density of the state at time $k-1$ is given by

$$p_{k-1}\left(x_{k-1}|z_{1:k-1}\right) = \mathcal{N}\left(x_{k-1}; m_{k-1}, P_{k-1}\right), \tag{3}$$

and the dynamics and observations are modelled according to

$$x_k = F_{k|k-1}x_{k-1} + v_{k-1}, \tag{4}$$

$$z_k = H_k x_k + w_k, \tag{5}$$

where $F_{k|k-1}$ is an $n_x \times n_x$ state transition matrix, H_k is an $n_z \times n_x$ observation matrix, $v_{k-1} \sim \mathcal{N}\left(0, Q_{k-1}\right)$ is a Gaussian distributed n_x-dimensional process noise vector with zero mean and covariance Q_{k-1}, and $w_k \sim \mathcal{N}\left(0, R_k\right)$ is a Gaussian distributed n_z-dimensional measurement noise vector with zero mean and covariance R_k. Based

upon these models, the state transition density and measurement likelihood function become

$$f_{k|k-1}\left(x_k|x_{k-1}\right) = \mathcal{N}\left(x_k; F_{k|k-1}x_{k-1}, Q_{k-1}\right), \tag{6}$$

$$g_k\left(z_k|x_k\right) = \mathcal{N}\left(z_k; H_k x_k, R_k\right). \tag{7}$$

Substituting (3) and (6) into (1) yields the predicted density at time k,

$$p_{k|k-1}\left(x_k|z_{1:k-1}\right) = \mathcal{N}\left(x_k; m_{k|k-1}, P_{k|k-1}\right), \tag{8}$$

where

$$m_{k|k-1} = F_{k|k-1}m_{k-1}, \tag{9}$$

$$P_{k|k-1} = F_{k|k-1}P_{k-1}F_{k|k-1} + Q_{k-1}. \tag{10}$$

Substituting (7) and (8) into (2) yields the posterior density at time k,

$$p_k\left(x_k|z_{1:k}\right) = \mathcal{N}\left(x_k; m_k, P_k\right), \tag{11}$$

where

$$m_k = m_{k|k-1} + K_k\left(z_k - H_k m_{k|k-1}\right), \tag{12}$$

$$P_k = P_{k|k-1} - K_k S_k K_k^T, \tag{13}$$

$$K_k = P_{k|k-1}H_k^T S_k^{-1}, \tag{14}$$

$$S_k = H_k P_{k|k-1}H_k^T + R_k. \tag{15}$$

Equations (9)-(10) are known as the *Kalman prediction*, equations (12)-(15) are known as the *Kalman update*, and collectively they are called the *Kalman filter*. The fact that the Gaussian distribution is both closed under the Chapman-Kolmogorov equation with linear-Gaussian transition density, and a conjugate prior with respect to a linear-Gaussian likelihood function, means that the Kalman filter is an exact closed form solution for the Bayes recursion under these modelling assumptions.

2.1.3 *Non-linear Gaussian Filtering*

Consider the more general case in which the system dynamics and observations cannot be modelled according to simple linear transformations as in (4) and (5). Instead, these models are of the form

$$x_k = s_{k-1}\left(x_{k-1}\right) + v_{k-1}, \tag{16}$$

$$z_k = h_k\left(x_k\right) + w_k, \tag{17}$$

where v_{k-1} and w_k are the same as defined in (4) and (5), and $s_{k-1}\left(\cdot\right)$ and $h_k\left(\cdot\right)$ are non-linear functions. In this case, there is no exact analytical solution to the Bayes recursion. However, it is possible to make some assumptions to facilitate an approximate analytical solution.

2.1.3.1 *Extended Kalman Filter*

The simplest technique for accommodating non-linear models is called the extended Kalman filter (EKF) [86, 1, 12]. In the EKF, the predicted and posterior densities at time k are given by (8) and (11) where

$$m_{k|k-1} = s_{k-1}\left(m_{k-1}\right), \tag{18}$$

$$P_{k|k-1} = \hat{F}_{k-1} P_{k-1} \hat{F}_{k-1}^T + Q_{k-1}, \tag{19}$$

$$m_k = m_{k|k-1} + \hat{K}_k\left(z_k - h_k\left(m_{k|k-1}\right)\right), \tag{20}$$

$$P_k = P_{k|k-1} - \hat{K}_k S_k \hat{K}_k^T, \tag{21}$$

in which

$$\hat{S}_k = \hat{H}_k P_{k|k-1} \hat{H}_k^T + R_k, \tag{22}$$

$$\hat{K}_k = P_{k|k-1} \hat{H}_k^T \hat{S}_k^{-1}, \tag{23}$$

$$\hat{F}_{k-1} = \left.\frac{\partial s_{k-1}\left(x\right)}{\partial x}\right|_{x=m_{k-1}}, \tag{24}$$

$$\hat{H}_k = \left.\frac{\partial h_k\left(x\right)}{\partial x}\right|_{x=m_{k|k-1}}. \tag{25}$$

The state transition matrix is replaced with a first order analytical linearisation (i.e. the Jabobian) of the non-linear state transition function, evaluated at the mean of the previous density. Similarly, the measurement matrix is replaced by the Jacobian of the

non-linear observation function, evaluated at the mean of the predicted density. Note that if either the transition or observation function is linear, then the corresponding step from the standard Kalman filter can be applied. For example, for a linear dynamic model and nonlinear observation model, the standard Kalman prediction can be used, followed by an EKF measurement update.

The underlying assumption of the EKF is that the nonlinear models can be adequately approximated by their linearised representations in the vicinity of the previous/predicted mean. This approximation often works well, however, in situations where the non-linearity is too severe, the EKF can be prone to failure [168].

2.1.3.2 *Unscented Kalman Filter*

In contrast to the EKF, which carries out an analytical linearisation of the non-linear functions, the UKF [97, 96, 201] performs a statistical linearisation by representing the filtering density as a deterministically chosen set of weighted sample points, also known as sigma points. These sigma points are propagated through the non-linear transition and measurement equations in order to compute the predicted and posterior densities. This process has been shown to be more robust than the EKF when faced with higher levels of model non-linearity.

Suppose that the density at time $k-1$ is represented by a weighted set of sigma points $\left\{\left(\mathcal{X}_{k-1}^{(i)}, w^{(i)}\right)\right\}_{i=1:N}$, where $N = 2n_x + 1$, chosen according to the unscented transform

$$\mathcal{X}_{k-1}^{(i)} = \begin{cases} m_{k-1}, & i = 1 \\ m_{k-1} + \left[\sqrt{(n_x+\kappa)\, P_{k-1}}\right]_{i-1}, & i = 2, \ldots, n_x + 1 \\ m_{k-1} - \left[\sqrt{(n_x+\kappa)\, P_{k-1}}\right]_{i-n-1}, & i = n+2, ..., 2n_x + 1 \end{cases}, \tag{26}$$

$$w^{(i)} = \begin{cases} \frac{\kappa}{n_x+\kappa}, & i = 1 \\ \frac{1}{2(n_x+\kappa)}, & i = 2, \ldots, 2n_x + 1 \end{cases}, \tag{27}$$

where κ is a scaling parameter such that $\kappa + n_x \neq 0$. In (26), the subscripted square brackets $[A]_i$ denote the i-th row of the matrix A, and the square root is the matrix square root, such that for a matrix A, $\left(\sqrt{A}\right)^T \left(\sqrt{A}\right) = A$.

The predicted density is given by (8) where

$$m_{k|k-1} = \sum_{i=1}^{N} w^{(i)} s_{k-1}\left(\mathcal{X}_{k-1}^{(i)}\right), \tag{28}$$

$$P_{k|k-1} = \sum_{i=1}^{N} w^{(i)} \left[s_{k-1}\left(\mathcal{X}_{k-1}^{(i)}\right) - m_{k|k-1}\right]\left[s_{k-1}\left(\mathcal{X}_{k-1}^{(i)}\right) - m_{k|k-1}\right]^T + Q_{k-1}, \tag{29}$$

which is also represented by the set of sigma points $\left\{\left(\mathcal{X}_{k|k-1}^{(i)}, w^{(i)}\right)\right\}_{i=1:N}$, where $\mathcal{X}_{k|k-1}^{(i)} = s_{k-1}\left(\mathcal{X}_{k-1}^{(i)}\right)$, i.e. each sigma point from the previous density is propagated through the state transition function. The predicted measurement is computed by taking the weighted sum of the measurement function evaluated at each sigma point as follows

$$\hat{z}_{k|k-1} = \sum_{i=1}^{N} w^{(i)} h_k\left(\mathcal{X}_{k|k-1}^{(i)}\right). \tag{30}$$

The posterior density is then given by (11) where

$$m_k = m_{k|k-1} + \hat{K}_k\left(z_k - \hat{z}_{k|k-1}\right), \tag{31}$$

$$P_k = P_{k|k-1} - \hat{K}_k S_k \hat{K}_k^T, \tag{32}$$

$$\hat{K}_k = P_{xz}\hat{S}_k^{-1}, \tag{33}$$

$$\hat{S}_k = P_{zz} + R_k, \tag{34}$$

$$P_{xz} = \sum_{i=1}^{N} w^{(i)} \left[\mathcal{X}_{k|k-1}^{(i)} - m_{k|k-1}\right]\left[h_k\left(\mathcal{X}_{k|k-1}^{(i)}\right) - \hat{z}_{k|k-1}\right]^T, \tag{35}$$

$$P_{zz} = \sum_{i=1}^{N} w^{(i)} \left[h_k\left(\mathcal{X}_{k|k-1}^{(i)}\right) - \hat{z}_{k|k-1}\right]\left[h_k\left(\mathcal{X}_{k|k-1}^{(i)}\right) - \hat{z}_{k|k-1}\right]^T. \tag{36}$$

Unlike the EKF, the UKF does not require the Jacobians of the non-linear functions s and h, thus it can be applied to problems in which these functions may be discontinuous. The use of the unscented transform yields predicted/posterior densities that are accurate up to the second order Taylor series expansion of the corresponding non-linear function, or the third order when the underlying random variable is Gaussian [96].

2.1.4 *Non-linear Non-Gaussian Filtering*

The main advantage of the Kalman filter and its non-linear variants is that they have analytical forms, making them efficient to compute. However, they are limited to the use of Gaussian models to represent the filtering density, as well as the process and and measurement noise distributions. An alternative approach that affords greater modelling flexibility is known as the particle filter [50, 63, 168, 6]. The fundamental principle behind this technique is the use of Monte Carlo integration to approximate the posterior density.

In particle filtering, the posterior density p_{k-1} of the state at time $k-1$ is represented by a weighted collection of points (or particles) $\left\{\left(x_{k-1}^{(i)}, w_{k-1}^{(i)}\right)\right\}_{i=1}^{N}$, leading to the following approximation for p_{k-1},

$$p_{k-1}\left(x_{k-1}|z_{1:k-1}\right) \approx \sum_{i=1}^{N} w_{k-1}^{(i)} \delta_{x_{k-1}^{(i)}}\left(x_{k-1}\right). \tag{37}$$

At time k, the posterior density p_k is approximated by a new collection of particles $\left\{\left(x_k^{(i)}, w_k^{(i)}\right)\right\}_{i=1}^{N}$, whose locations are drawn from a proposal density $q_k\left(\cdot|x_{k-1}^{(i)}, z_k\right)$. This leads to the posterior being approximated as

$$p_k\left(x_k|z_{1:k}\right) \approx \sum_{i=1}^{N} w_k^{(i)} \delta_{x_k^{(i)}}\left(x_k\right), \tag{38}$$

where

$$x_k^{(i)} \sim q_k\left(\cdot|x_{k-1}^{(i)}, z_z\right), \tag{39}$$

$$\tilde{w}_k^{(i)} = w_{k-1}^{(i)} \frac{g_k\left(z_k|x_k^{(i)}\right) p_{k|k-1}\left(x_k^{(i)}|x_{k-1}^{(i)}\right)}{q_k\left(x_k^{(i)}|x_{k-1}^{(i)}, z_k\right)}, \tag{40}$$

$$w_k^{(i)} = \frac{\tilde{w}_k^{(i)}}{\sum_{i=1}^{N} \tilde{w}_k^{(i)}}. \tag{41}$$

In principle, the importance density $q_k\left(\cdot|x_{k-1}^{(i)}, z_k\right)$ can be any function, provided that its support adequately covers the region of high posterior density. However, due to the finite number of particles, this choice can have a significant impact on the performance of the algorithm. In practice, it is necessary to use an importance density that concentrates most of the samples in regions of the state space that yield high posterior weights. This is required to ensure that the sampled representation

faithfully reflects the shape of the true posterior density. The optimal choice for the importance density, i.e. that which minimises the variance of the posterior weights is [49]

$$q_k\left(x_k|x_{k-1}^{(i)}, z_k\right) = \frac{p\left(z_k|x_k, x_{k-1}^{(i)}\right) p\left(x_k|x_{k-1}^{(i)}\right)}{\int p\left(z_k|x_k\right) p\left(x_k|x_{k-1}^{(i)}\right) dx}, \tag{42}$$

but in many applications it is difficult to draw samples directly from this distribution. Most often, suboptimal importance densities are used, and a popular choice is the transition density

$$q_k\left(x_k|x_{k-1}^{(i)}, z_k\right) = p\left(x_k|x_{k-1}^{(i)}\right), \tag{43}$$

because this leads to a simplified weight calculation, with the unnormalised posterior weight (40) reducing to

$$\tilde{w}_k^{(i)} = w_{k-1}^{(i)} g\left(z_k|x_k^{(i)}\right). \tag{44}$$

It has been shown that this method can lead to rapid particle depletion, because the transition density is usually much more diffuse than the measurement likelihood function. Improved choices for the importance density have been studied in works such as [160, 49, 121, 202]. More recently, a technique called particle Markov chain Monte Carlo has been developed [3], which allows for improved importance sampling in higher dimensional state spaces.

An ideal Monte Carlo representation of the posterior should have all particles equally weighted. In this case, the shape of the posterior is captured simply by variations in the concentration of particles in different regions of the state space. However, as the particle filter recursion is repeatedly applied, the variance of the weights begins to increase, depleting the effectiveness of the Monte Carlo representation. The most popular technique for addressing this is known as resampling [63], which generates an equally weighted set of samples from a sample set with unequal weights. This can be achieved in a number of ways (see for example [47, 31, 79]), however, the basic idea is that particles with higher weights are replicated, and those with lower weights are removed.

Resampling is a computationally intensive operation, and is therefore the main bottleneck in many particle filter implementations. Given this, it is often desirable to

avoid resampling on every iteration, and adaptive resampling strategies can be used to achieve this whilst still maintaining sufficient sample diversity [144]. Furthermore, speeding up the resampling itself has the potential to significantly improve execution time. Some research effort has been devoted to this problem, including recent works such as [115] and [146], which have investigated the use of graphics processing units (GPUs) to accelerate the resampling process.

Sequential Monte Carlo methods can also be used to perform smoothing, where future data is used to obtain improved state estimates at some time in the past. For example, a forward-backward smoother based on SMC methods was proposed in [60], and a two-filter particle smoother was proposed in [32]. These techniques have a computational complexity which is quadratic in the number of samples, however methods such as those proposed in [105] and [53] can alleviate this problem.

Another interesting application where SMC techniques have been used is in static parameter estimation, see for example [48, 2, 143]. These methods can be applied in situations where prior knowledge of the system parameters is lacking, and instead their values need to be estimated from the data.

2.2 RANDOM FINITE SETS

A random variable is defined as variable that can take on different values due to the effects of random chance. In traditional probability and statistics, a random variable is usually assumed to have a fixed and known dimension (i.e. vector-valued), and the chances of it taking on different values is characterised by its probability distribution. There are many important problems that cannot be adequately modelled using vector-valued random variables alone. Falling into this category are estimation problems in which the number of items to be estimated is unknown and time-varying, a common example of which is multi-target tracking. In problems such as this, the probabilistic model of the underlying random process must accommodate variations in the number of items, as well as their values. A more natural model for this type of system is called a *point process*, which is the subject of a branch of probability theory known as stochastic geometry [181]. For many practical applications, a more useful and realistic model is the *simple finite point process*, also known as a *random finite set* (RFS). A mathematical framework known as *finite set statistics* (FISST) [135] provides techniques for working with RFSs in a rigorous and consistent manner.

In the framework of FISST, a random finite set is defined as a *finite set-valued random variable*. Whereas a random vector has a fixed and known number of random elements appearing in a specific order, the key feature distinguishing an RFS is that the number of elements (or cardinality) is random, and the elements themselves are random, distinct and unordered. A random finite set has a cardinality distribution, which is a discrete distribution $\rho(\cdot)$ defined on the non-negative integers $\mathbb{N}$, where $\rho(n)$ is the probability that the RFS has exactly n elements. For each cardinality $n \in \mathbb{N}$, there is a joint probability distribution $P_n(\cdot)$ specifying the conditional density of the elements given that the cardinality of the set is equal to n. The collection of all such joint distributions, together with the cardinality distribution, provides a complete specification of a random finite set.

2.2.1 *Random Finite Set Densities*

An random finite set X, defined on the space $\mathbb{X}$, is a random variable taking values in $\mathcal{F}(\mathbb{X})$, i.e. a finite set-valued random variable. Both the number of elements and the value of the elements of an RFS are random. There are several constructs for specifying the probability law of an RFS, one of which is the *belief* (or *containment*) *functional* B, given by

$$B(S) = \Pr(X \subseteq S) \tag{45}$$

for any (closed) $S \subseteq \mathbb{X}$ [181, 140, 136, 138]. In fact, the belief functional uniquely determines the probability law of a random closed set (and hence an RFS) via Choquet's capacity theorem.

Since the belief functional is not a measure, the standard notion of density as a Radon-Nikodym derivative is not applicable. Nevertheless, an alternative notion of density can be defined via Mahler's set calculus [126, 136]. The *belief density* of an RFS X is a non-negative function π on $\mathcal{F}(\mathbb{X})$ such that for any $S \subseteq \mathbb{X}$,

$$B(S) = \int_S \pi(X)\,\delta X, \tag{46}$$

where the integral above is Mahler's *set integral* defined by [126, 136]

$$\int_S \pi(X)\,\delta X = \sum_{i=0}^{\infty} \frac{1}{i!} \int_{S^i} \pi(\{x_1, ..., x_i\})\, d(x_1, ..., x_i) \tag{47}$$

(note that since $S^0 = \{\emptyset\}$, the integral over S^0 is simply $\pi(\emptyset)$). That is, the set integral of the belief density π over a region S, yields the probability that X is contained in S. Note that $\pi(X)$ has dimension $K^{-|X|}$, where K denotes the unit of hyper-volume on $\mathbb{X}$.

While the belief density π is not a probability density, the dimensionless function on $\mathcal{F}(\mathbb{X})$ defined by $\pi(X) K^{|X|}$ is indeed a probability density with respect to the measure μ given by [194]

$$\mu(\mathcal{T}) = \sum_{i=0}^{\infty} \frac{1}{i! K^i} \int_{\mathbb{X}^i} 1_{\mathcal{T}}(\{x_1, ..., x_i\}) \, d(x_1, ..., x_i) \tag{48}$$

for any measurable $\mathcal{T} \subseteq \mathcal{F}(\mathbb{X})$. A belief density and its corresponding probability density are equivalent representations of the probability law of an RFS, which differ only by the term $K^{|X|}$. For the sake of simplicity, from this point on, the term 'density' when used in reference to an RFS, refers to either the belief density or its equivalent probability density.

2.2.2 *Common Types of Random Finite Set*

In finite set statistics, there are several important classes of RFS, each having its own unique properties. This section provides a brief description of the four most common types of RFS used in multi-object system modelling, namely the Poisson, independent and identically distributed cluster, Bernoulli and multi-Bernoulli.

Poisson

An RFS with intensity function $v(\cdot)$ is called a *Poisson RFS* if its cardinality follows a Poisson distribution with mean $\lambda = \int v(x) \, dx$, i.e.

$$\rho(n) = \frac{e^{-\lambda} \lambda^n}{n!}. \tag{49}$$

This means that a Poisson RFS is completely specified by its intensity function v, since knowledge of the intensity also gives us knowledge of the cardinality distribution. For a given cardinality, the elements of the set are all distributed according to the probability density $v(\cdot)/\lambda$, thus the density of a Poisson RFS X is given by

$$\pi(X) = e^{-\lambda} \prod_{x \in X} v(x). \tag{50}$$

The Poisson RFS is often used to represent processes exhibiting complete spatial randomness. For example, a common assumption in tracking applications is that the set of false alarms produced by a sensor is completely spatially random. Thus, the false alarms are usually modelled by a Poisson RFS, often with a uniform intensity function.

Independent Identically Distributed Cluster

An RFS that can be uniquely specified by its cardinality distribution $\rho(\cdot)$ and a matching intensity function $v(\cdot)$ is called an *independently and identically distributed (i.i.d.) cluster RFS*. In contrast to a Poisson RFS (whose cardinality distribution is always Poisson), the cardinality distribution of an i.i.d. cluster RFS can have an arbitrary form, with the only restriction being that its mean must match the integral of the intensity function ($N = \sum_{n=0}^{\infty} n\rho(n) = \int v(x)\,dx$). The individual elements of the set are all distributed according to $v(\cdot)/N$, thus the density of an i.i.d. cluster RFS is given by

$$\pi(X) = n!\rho(|X|) \prod_{x \in X} \frac{v(x)}{N}. \tag{51}$$

It is worth noting here that a Poisson RFS is a special case of an i.i.d. cluster RFS, where the cardinality distribution is constrained to the form of a Poisson.

Bernoulli

An RFS which has probability $1 - r$ of being empty, and probability r of containing a single element distributed according to $p(\cdot)$, is called a *Bernoulli RFS*. This type of RFS is completely specified by the parameters r and p, and is distributed according to the density

$$\pi(X) = \begin{cases} 1 - r, & X = \emptyset \\ r.p(x), & X = \{x\} \\ 0, & |X| > 1 \end{cases}. \tag{52}$$

The Bernoulli RFS is useful in situations involving at most one object, but it is not certain whether the object is present. A prime example of this is in single-target tracking, where the target may or may not be present, thus the target density is naturally modelled as a Bernoulli RFS.

Multi-Bernoulli

A *multi-Bernoulli RFS* is an RFS in which the objects are statistically independent of each other, and the i-th object has probability of existence $r^{(i)}$ and spatial probability density $p^{(i)}$, for $i = 1, \ldots, M$. Thus, a multi-Bernoulli RFS is the union of a fixed number of independent Bernoulli RFSs. This type of RFS is completely specified by the set of parameters $\left\{ \left(r^{(i)}, p^{(i)} \right) \right\}_{i=1}^{M}$, and its density is given by

$$\pi\left(\{x_1, \ldots, x_n\}\right) = \begin{cases} \prod_{j=1}^{M} \left(1 - r^{(j)}\right), & n = 0 \\ \prod_{j=1}^{M} \left(1 - r^{(j)}\right) \sum_{1 \leq i_1 \neq \cdots \neq i_n \leq M} \prod_{j=1}^{n} \frac{r^{(i_j)} p^{(i_j)}(x_j)}{1 - r^{(i_j)}}, & 1 \leq n \leq M \cdot \\ 0, & n > M \end{cases} \tag{53}$$

The multi-Bernoulli RFS is useful in cases involving multiple objects, but the presence of any particular object is not certain. One example of this is in a multi-target system model, where the individual targets have independent probabilities of existence. Another example is a multi-target detection model, where the targets have independent probabilities of generating a detection.

2.3 MULTI-TARGET BAYES FILTER

The goal of recursive multi-object Bayesian estimation is to estimate a finite set of states $X_k \subset \mathbb{X}$, called the *multi-object state*, at each time k. The multi-object states X_k and *multi-object observations* $Z_k \subset \mathbb{Z}$ are modelled as random finite sets. At the previous time step $k-1$, the multi-object state is assumed to be distributed according to a *multi-object density* $\pi_{k-1}\left(\cdot | Z_{1:k-1}\right)$, where $Z_{1:k-1}$ is an array of finite sets of measurements received up to time $k-1$. Each Z_k is assumed to be generated through a process of thinning of misdetected objects, Markov shifts of detected objects, and superposition of false measurements. The *multi-object prediction* to time k is given by the Chapman-Kolmogorov equation

$$\pi_{k|k-1}\left(X_k | Z_{1:k-1}\right) = \int f_{k|k-1}\left(X_k | X\right) \pi_{k-1}\left(X | Z_{1:k-1}\right) \delta X, \tag{54}$$

where $f_{k|k-1}(X_k|X)$ is the multi-object transition kernel from time $k-1$ to time k, and the integral is the set integral,

$$\int f(X)\,\delta X = \sum_{i=0}^{\infty} \frac{1}{i!} \int_{\mathbb{X}^i} f(\{x_1,\ldots,x_i\})\,d(x_1,\ldots,x_i) \tag{55}$$

for any function f that takes $\mathcal{F}(\mathbb{X})$, the collection of all finite subsets of $\mathbb{X}$, to the real line. A new set of observations Z_k is received at time k, which is modelled by a *multi-object likelihood function* $g_k(Z_k|X_k)$. Thus the *multi-object posterior* at time k is given by Bayes rule

$$\pi_k(X_k|Z_{1:k}) = \frac{g_k(Z_k|X_k)\,\pi_{k|k-1}(X_k|Z_{1:k-1})}{\int g_k(Z_k|X)\,\pi_{k|k-1}(X|Z_{1:k-1})\,\delta X}. \tag{56}$$

Collectively, (54) and (56) are referred to as the *multi-object Bayes filter.* This is a completely general form, which serves as the basis for deriving multi-target filtering and tracking algorithms under the RFS framework. In general, computing the exact multi-object posterior is numerically intractable, and approximations are required to derive practical algorithms. Specifically, it is necessary to restrict the form of the density π_k to a tractable class of functions that is closed under the operations (54) and (56). Such a class of functions is commonly referred to as a *conjugate prior*.

2.4 MULTI-TARGET FILTERING WITH RANDOM FINITE SETS

2.4.1 *Multi-object Dynamic Model*

In this section we specify an RFS formulation of the standard multi-object dynamic model, which captures the evolution of the individual object states, as well as the appearance and disappearance of objects over time. Let X_{k-1} represent the multi-object state at time $k-1$. Each object $x_{k-1} \in X_{k-1}$ has probability $P_{S,k}(x_{k-1})$ of surviving to time k, and probability $1 - P_{S,k}(x_{k-1})$ of being terminated somewhere between times $k-1$ and k. If an object with state x_{k-1} survives to time k, it takes on a new state x_k with probability density $p_{k|k-1}(x_k|x_{k-1})$, and if terminated it becomes the empty set $\emptyset$. Hence, each object gives rise to a Bernoulli RFS as defined by (52), with parameters $r = P_{S,k}(x_{k-1})$ and $p(\cdot) = p_{k|k-1}(\cdot|x_{k-1})$. The objects are assumed to be conditionally independent, thus the RFS of objects surviving from

$k-1$ to k is a multi-Bernoulli, which we denote by $S_{k|k-1}\left(X_{k|k-1}\right)$. The density of $S_{k|k-1}\left(X_{k-1}\right)$ is denoted by $\pi_{S,k|k-1}\left(\cdot|X_{k-1}\right)$, and has the form (53) with parameter set $\left\{\left(P_{S,k}\left(x_{k-1}\right), p_{k|k-1}\left(\cdot|x_{k-1}\right)\right)\right\}_{x_{k-1}\in X_{k-1}}$.

For the multi-object dynamic model to be useful in problems with an unknown and time-varying number of objects, it also needs to include a model for the appearance of new objects (referred to as target births). The target births at time k are represented by an RFS B_k, which can be modelled by any type of RFS permitted by the particular filtering algorithm being used. The complete RFS of targets at time k is then given by the union of survivals and births

$$X_k = S_{k|k-1}\left(X_{k-1}\right) \cup B_k. \tag{57}$$

Let $\pi_{B,k}\left(\cdot\right)$ denote the density of B_k. Then, the predicted density of X_k is given by the convolution

$$p_{k|k-1}\left(X_k|X_{k-1}\right) = \sum_{W\subseteq X_k} \pi_{S,k|k-1}\left(W|X_{k-1}\right)\pi_{B,k}\left(X_k - W\right). \tag{58}$$

2.4.2 *Multi-object Observation Model*

Let us assume that an observation is received at time k, and let X_k represent the true multi-object state at that time. Each object $x_k \in X_k$ has probability $P_{D,k}\left(x_k\right)$ of being detected, and probability $1 - P_{D,k}\left(x_k\right)$ of being missed. If detected, the object gives rise to a measurement z_k with probability density $g_k\left(z_k|x_k\right)$, and if missed it generates the empty set $\emptyset$. Thus, each object produces an observation modelled by a Bernoulli RFS with parameters $r = P_{D,k}\left(x_k\right)$ and $p\left(\cdot\right) = g_k\left(\cdot|x_k\right)$. The single-object measurements are assumed to be conditionally independent, thus the RFS of detections, which we denote by $D_k\left(X_k\right)$, is a multi-Bernoulli RFS. The density of $D_k\left(X_k\right)$ is denoted by $\pi_{D,k}\left(\cdot|X_k\right)$, and is of the form (53) with parameter set $\left\{\left(P_{D,k}\left(x_k\right), g_k\left(\cdot|x_k\right)\right)\right\}_{x_k\in X_k}$.

In addition to the detections of real objects, sensors typically produce false alarms, which also need to be captured by the model. The set of false alarms produced at time k is modelled by an RFS K_k, which may be of any type depending on the filter in

use, but is most commonly modelled as a Poisson RFS. The overall set of observations is given by the union of target detections and false alarms

$$Z_k = D_k(X_k) \cup K_k. \tag{59}$$

Let $\pi_{K,k}(\cdot)$ denote the density of K_k. The density of Z_k is then given by the convolution

$$g_k(Z_k|X_k) = \sum_{W \subseteq Z_k} \pi_{D,k}(W|X_k)\, \pi_{K,k}(Z_k - W), \tag{60}$$

which is known as the multi-object measurement likelihood function.

2.4.3 *Probability Hypothesis Density Filter*

One of the first algorithms to be derived based on random finite sets was the probability hypothesis density filter [126]. The fundamental idea behind this algorithm is to achieve a tractable approximation to (56) by propagating only the first moment of the multi-object density, known as its probability hypothesis density (PHD). The PHD is a function defined on the single-object state space, and has the unique property that its integral over a given region of the state space yields the expected number of objects in that region.

Suppose that v_{k-1} represents the posterior PHD at time $k-1$, $f_{k|k-1}$ the single object transition kernel from time $k-1$ to time k, $P_{S,k}$ the probability of object survival, and γ_k the PHD of spontaneous target births. The predicted PHD at time k is then given by

$$v_{k|k-1}(x_k) = \gamma_k(x_k) + \int P_{S,k}(x_{k-1})\, f_{k|k-1}(x_k|x_{k-1})\, v_{k-1}(x_{k-1})\, dx_{k-1}. \tag{61}$$

Now, denote by Z_k the set of measurements generated by a sensor at time k. This measurement set is assumed to contain false alarms with intensity κ_k, and let the $P_{D,k}(x_k)$ be the probability that an object with state x_k is detected. If detected, an object generates a measurement with likelihood function g_k. The posterior PHD at time k is given by

$$v_k(x_k) = (1 - P_{D,k}(x_k))\, v_{k|k-1}(x_k) + \sum_{z \in Z_k} \frac{P_{D,k}(x_k)\, g_k(z|x_k)\, v_{k|k-1}(x_k)}{\kappa_k(z) + \int P_{D,k}(x)\, g_k(z|x)\, v_{k|k-1}(x)\, dx}. \tag{62}$$

The first implementation of the PHD filter was based on the sequential Monte Carlo method, and was independently proposed in 2003 by Sidenbladh [174], Zajic and Mahler [207], and Vo, Singh and Doucet [193]. An analysis of the convergence properties of the SMC-PHD filter appeared in [40, 92]. The first analytic implementation of the PHD filter based on Gaussian mixtures was proposed in 2006 by Vo and Ma [192], and a convergence analysis of this filter appeared in [39]. The work in [192] included an implementation for linear-Gaussian models based on Kalman filters, and for nonlinear-Gaussian models based on the EKF or UKF. A version based on the Gaussian particle filter was presented in [39], and another based on the cubature Kalman filter in [125]. The Gaussian mixture PHD filter was extended to jump Markov models in [158], allowing it to handle manoeuvring targets.

Over the last decade, the PHD filter has become a popular topic among tracking researchers, and applications to many different multi-target filtering problems have been proposed. Useful summaries of some of the notable applications are provided in [135] and [136].

2.4.4 *Cardinalised Probability Hypothesis Density Filter*

The cardinalised probability hypothesis density (CPHD) filter, proposed by Mahler in [129], is a generalisation of the PHD filter allowing for more flexible modelling of the target cardinality distribution. Whereas the PHD filter models the multi-object state as a Poisson RFS, the CPHD filter models it as a more general i.i.d cluster RFS. This means that the cardinality distribution is no longer constrained to be a Poisson, and instead may take on any arbitrary form. The drawback of this approach is that the filter equations become more complicated than the PHD filter, and there is an increase in computational complexity. However, the i.i.d cluster model has been shown to improve the filtering performance, with the CPHD filter achieving better state estimates and reduced variance in the cardinality estimates [196].

Let v_{k-1} be the PHD at time $k-1$, ρ_{k-1} the cardinality distribution, $P_{S,k}$ the target survival probability, $f_{k|k-1}$ the single object transition kernel, $\rho_{B,k}$ the cardinality distribution of spontaneous births at time k, and $C_j^l = \frac{l!}{j!(l-j)!}$ the binomial coefficient. The predicted PHD and cardinality distribution at time k are given by

$$v_{k|k-1}(x_k) = \int P_{S,k}(x_{k-1}) f_{k|k-1}(x_k|x_{k-1}) v_{k-1}(x_{k-1}) dx_{k-1} \tag{63}$$

$$\rho_{k|k-1}(n) = \sum_{j=0}^{n} \rho_{B,k}(n-j)\, \Pi_{k|k-1}\left[v_{k-1}, \rho_{k-1}\right](j) \tag{64}$$

where,

$$\Pi_{k|k-1}[v,\rho](j) = \sum_{l=j}^{\infty} C_j^l \frac{\langle P_{S,k}, v\rangle^j \langle 1 - P_{S,k}(x), v(x)\rangle^{l-j}}{\langle 1, v\rangle^l}. \tag{65}$$

The posterior PHD and cardinality distribution are given by

$$\begin{aligned} v_k(x) = {} & (1 - P_{D,k}(x)) \frac{\left\langle \Upsilon_k^1\left[v_{k|k-1}; Z_k\right], \rho_{k|k-1}\right\rangle}{\left\langle \Upsilon_k^0\left[v_{k|k-1}; Z_k\right], \rho_{k|k-1}\right\rangle} v_{k|k-1}(x) \\ & + \sum_{z \in Z_k} \psi_{k,z}(x) \frac{\left\langle \Upsilon_k^1\left[v_{k|k-1}; Z_k - \{z\}\right], \rho_{k|k-1}\right\rangle}{\left\langle \Upsilon_k^0\left[v_{k|k-1}; Z_k\right], \rho_{k|k-1}\right\rangle} v_{k|k-1}(x) \end{aligned} \tag{66}$$

$$\rho_k(n) = \frac{\Upsilon_k^0\left[v_{k|k-1}; Z_k\right](n)\, \rho_{k|k-1}(n)}{\left\langle \Upsilon_k^0\left[v_{k|k-1}; Z_k\right], \rho_{k|k-1}\right\rangle} \tag{67}$$

where,

$$\begin{aligned} \Upsilon_k^u[v, Z](n) = {} & \sum_{j=0}^{\min(|Z|,n)} (|Z| - j)!\, p_{K,k}(|Z| - j)\, P_{j+u}^n \\ & \times \frac{\langle 1 - P_{D,k}, v\rangle^{n-(j+u)}}{\langle 1, v\rangle^n} e_j\left(\Xi_k(v, Z)\right), \end{aligned} \tag{68}$$

$$\psi_{k,z}(x) = \frac{\langle 1, \kappa_k\rangle}{\kappa_k(z)} g_k(z|x)\, P_{D,k}(x), \tag{69}$$

$$\Xi_k(v, Z) = \{\langle v, \psi_{k,z}\rangle : z \in Z\}. \tag{70}$$

Like the PHD filter, the CPHD filter has been implemented using Gaussian mixtures [189, 199, 196], for both linear-Gaussian models based on the Kalman filter, and nonlinear-Gaussian models using the EKF and UKF. The CPHD filter has also been implemented for nonlinear non-Gaussian models using sequential Monte Carlo methods [196]. For the case of manoeuvring targets, a CPHD filter based on a jump Markov model was proposed in [132].

2.4.5 *Cardinality Balanced Multi-object Multi-Bernoulli Filter*

The PHD and CPHD filters approximate the multi-object Bayes recursion by propagating the first moment of the multi-object density, and in the case of the CPHD filter, the cardinality distribution. In contrast, an algorithm called the multi-object multi-Bernoulli (MeMBer) filter, proposed by Mahler in [135], approximates the Bayes re-

cursion by approximating the posterior as a multi-Bernoulli random finite set. The original MeMBer filter was designed to operate on standard point-detection level data, and a version that operates directly on image observations, without the requirement for detections, was proposed in [80]. In [200], Vo, Vo and Cantoni pointed out that a cardinality bias was present in the original formulation of the MeMBer filter, and proposed a solution known as the cardinality balanced MeMBer (CBMeMBer) filter, for the point-detection model. The CBMeMBer recursion is summarised as follows.

Let the multi-object density at time $k-1$ be a multi-Bernoulli of the form $\pi_{k-1} = \left\{\left(r_{k-1}^{(i)}, p_{k-1}^{(i)}\right)\right\}_{i=1}^{M_{k-1}}$, and let the density of spontaneous births be a multi-Bernoulli given by $\pi_{B,k} = \left\{\left(r_{B,k}^{(i)}, p_{B,k}^{(i)}\right)\right\}_{i=1}^{M_{B,k}}$. Then the predicted density at time k is a multi-Bernoulli given by

$$\pi_{k|k-1} = \left\{\left(r_{P,k|k-1}^{(i)}, p_{P,k|k-1}^{(i)}\right)\right\}_{i=1}^{M_{k-1}} \cup \left\{\left(r_{B,k}^{(i)}, p_{B,k}^{(i)}\right)\right\}_{i=1}^{M_{B,k}} \tag{71}$$

where

$$r_{P,k|k-1}^{(i)} = r_{k-1}^{(i)} \left\langle p_{k-1}^{(i)}, P_{S,k} \right\rangle, \tag{72}$$

$$p_{P,k|k-1}^{(i)}(x) = \frac{\left\langle f_{k|k-1}(x|\cdot), p_{k-1}^{(i)} P_{S,k} \right\rangle}{\left\langle p_{k-1}^{(i)}, P_{S,k} \right\rangle}. \tag{73}$$

If we collect the terms of the predicted density into a single set of parameters, which we express as $\pi_{k|k-1} = \left\{\left(r_{k|k-1}^{(i)}, p_{k|k-1}^{(i)}\right)\right\}_{i=1}^{M_{k|k-1}}$, then the CBMeMBer filter approximates the posterior multi-object density as a multi-Bernoulli, given by

$$\pi_k \approx \left\{\left(r_{L,k}^{(i)}, p_{L,k}^{(i)}\right)\right\}_{i=1}^{M_{k|k-1}} \cup \left\{\left(r_{U,k}^{*}(z), p_{U,k}^{*}(\cdot; z)\right)\right\}_{z \in Z_k} \tag{74}$$

where

$$r_{L,k}^{(i)} = r_{k|k-1}^{(i)} \frac{1 - \left\langle p_{k|k-1}^{(i)}, P_{D,k} \right\rangle}{1 - r_{k|k-1}^{(i)} \left\langle p^{(i)}, P_{D,k} \right\rangle}, \tag{75}$$

$$p_{L,k}^{(i)}(x) = p_{k|k-1}^{(i)}(x) \frac{1 - P_{D,k}(x)}{1 - \left\langle p_{k|k-1}^{(i)}, P_{D,k} \right\rangle}, \tag{76}$$

$$r^*_{U,k}(z) = \frac{\sum_{i=1}^{M_{k|k-1}} \frac{r^{(i)}_{k|k-1}\left(1-r^{(i)}_{k|k-1}\right)\left\langle p^{(i)}_{k|k-1},\psi_{k,z}\right\rangle}{\left(1-r^{(i)}_{k|k-1}\left\langle p^{(i)}_{k|k-1},P_{D,k}\right\rangle\right)^2}}{\kappa_k(z) + \sum_{i=1}^{M_{k|k-1}} \frac{r^{(i)}_{k|k-1}\left\langle p^{(i)}_{k|k-1},\psi_{k,z}\right\rangle}{1-r^{(i)}_{k|k-1}\left\langle p^{(i)}_{k|k-1},P_{D,k}\right\rangle}}, \tag{77}$$

$$p^*_{U,k}(x;z) = \frac{\sum_{i=1}^{M_{k|k-1}} \frac{r^{(i)}_{k|k-1}}{1-r^{(i)}_{k|k-1}} p^{(i)}_{k|k-1}(x)\,\psi_{k,z}(x)}{\sum_{i=1}^{M_{k|k-1}} \frac{r^{(i)}_{k|k-1}}{1-r^{(i)}_{k|k-1}} \left\langle p^{(i)}_{k|k-1},\psi_{k,z}\right\rangle}, \tag{78}$$

$$\psi_{k,z}(x) = g_k(z|x)\,P_{D,k}(x). \tag{79}$$

Like the PHD and CPHD filters, the CBMeMBer filter can be implemented using either Gaussian mixtures or sequential Monte Carlo methods, both of which were proposed in [200, 196]. A convergence analysis of the SMC-CBMeMBer filter was presented in [119]. Some interesting applications of the CBMeMBer filter include multi-target tracking with audio-visual measurement data [81], an extension to tracking manoeuvring targets using a jump Markov model [51], and its use in multi-target sensor control [64].

2.4.6 *PHD/CPHD Filtering with Unknown Birth Model*

The original forms of the Gaussian mixture PHD/CPHD filters assume that both the posterior PHD at the previous time, and the prior PHD for new targets entering the scene at the current time, are both Gaussian mixtures [192, 199]. Although it is possible to approximate an arbitrary density using a sum of Gaussians, this approach is potentially inefficient, since a large number of Gaussians may be necessary to obtain a reasonable approximation. This is especially true in the case where the birth density is uniform, which applies in the common situation where targets may appear anywhere with equal probability.

The use of a diffuse target birth model is addressed for the GMPHD filter in [83] and for the SMC-PHD filter in [171]. A more formal derivation of the modified GM-PHD filter, and the derivation of a modified GMCPHD filter was given in the author's conference paper [16], along with some preliminary simulation results for a simple linear-Gaussian scenario. The author's journal article [18] built upon this, by applying the modified GMPHD/CPHD filters to the more challenging problem of bearings-only filtering, where the angular component of the PHD for new targets

is modelled using a uniform distribution on the bearing space. The modified filters were compared to the corresponding traditional versions under various GM approximations to this birth model.

The modified filter equations are based on a principled approximation, which is derived by showing that the birth PHD can be defined using a uniform distribution on the part of the state space which is directly observed through the measurements, and a Gaussian distribution for the unobserved parts. We refer to this as a partially uniform birth (PUB) model, and the filters based on this model as the PUB-GMPHD and PUB-GMCPHD filters. Use of the PUB model leads to a predicted PHD which is no longer a Gaussian mixture. Nonetheless, the updated PHD can be accurately approximated by a GM, provided that the birth PHD is Gaussian in the unobserved state components, and the uniform distribution for the observed state components covers a large enough region such that the effect of truncating the Gaussians is negligible.

The main advantage of this approach is that it reduces the number of parameters that must be selected by the end user. To construct a GM representation of the birth PHD requires selecting a number of parameters, including the mean locations, covariance matrices and weights of the Gaussians. Poor selection can adversely affect performance, so there is a potentially significant benefit in removing this requirement. Furthermore, simulations show that this method provides much better performance than a GM with a small number of components, and similar performance to a GM with a large number of components.

The remainder of this section presents the results of the work from the author's journal article [18] and conference paper [16].

2.4.6.1 *GMPHD Filter with Partially Uniform Birth (PUB-GMPHD)*

The general form of the PHD filter recursion (without target spawning) is given by [126]

$$v_{k|k-1}(x) = \left\langle P_{S,k}(\cdot)\, f_{k|k-1}(x|\cdot), v_{k-1}(\cdot) \right\rangle + \gamma_k(x), \tag{80}$$

$$v_k(x) = (1 - P_{D,k}(x))\, v_{k|k-1}(x) + \sum_{z \in Z} \frac{P_{D,k}(x)\, g_k(z|x)\, v_{k|k-1}(x)}{\kappa_k(z) + \left\langle P_{D,k}(\cdot)\, g_k(z|\cdot), v_{k|k-1}(\cdot) \right\rangle}, \tag{81}$$

where v_{k-1} is the posterior PHD from time $k-1$, $P_{S,k}$ is the target survival probability at time k, $f_{k|k-1}$ is the single-target transition density from time $k-1$ to time k, γ_k

is the prior PHD of spontaneous target births at time k, $P_{D,k}$ is the probability of detection at time k, $g_k(z|x)$ is the likelihood function for a measurement z given target state x, and $\kappa_k(z)$ is the clutter density at time k in the vicinity of measurement z. We now derive approximations of the PHD prediction and update for the case when the PHD of spontaneous target births is partially uniform, which we refer to as the PUB-GMPHD filter.

PREDICTION Let the posterior PHD at time $k-1$ be a Gaussian mixture of the form

$$v_{k-1}(x) = \sum_{i=1}^{J_{k-1}} w_{k-1}^{(i)} \mathcal{N}\left(x; m_{k-1}^{(i)}, P_{k-1}^{(i)}\right). \tag{82}$$

Let θ represent the part of the state x that is directly observed through measurement z, and ϕ be the part of the state which is not observed (i.e. θ is either a scalar or a vector consisting of the terms involved in the measurement likelihood calculation (89), and ϕ is a scalar or vector consisting of the remaining terms). The PHD in (82) may be expressed as

$$v_{k-1}(\theta,\phi) = \sum_{i=1}^{J_{k-1}} w_{k-1}^{(i)} \mathcal{N}\left(\theta,\phi; m_{\theta,k-1}^{(i)}, m_{\phi,k-1}^{(i)}, P_{(\theta,\phi),k-1}^{(i)}\right) \tag{83}$$

where $m_{\theta,k-1}^{(i)}$ is the observed part of $m_{k-1}^{(i)}$, $m_{\phi,k-1}^{(i)}$ is the unobserved part of $m_{k-1}^{(i)}$, and $P_{(\theta,\phi),k-1}^{(i)}$ is the covariance matrix $P_{k-1}^{(i)}$ transformed into the space defined by (θ,ϕ). We now define the PHD of spontaneous target births as

$$\gamma_k(\theta,\phi) = w_{b,k}\mathcal{U}(\theta;\mathbb{B})\,\mathcal{N}\left(\phi;\bar{\phi},P_\phi\right) \tag{84}$$

where $w_{b,k}$ is the expected number of targets appearing at time k, $\mathcal{U}(\theta;\mathbb{B})$ is the uniform distribution in θ over the region $\mathbb{B}$, $\bar{\phi}$ is the prior mean of the unmeasured state component, and P_ϕ is its prior covariance matrix. Now, the predicted PHD at time k is given by

$$v_{k|k-1}(\theta,\phi) = v_{S,k|k-1}(\theta,\phi) + \gamma_k(\theta,\phi), \tag{85}$$

where

$$v_{S,k|k-1}(\theta,\phi) = p_{S,k}\sum_{i=1}^{J_{k-1}} w_{k-1}^{(i)} \mathcal{N}\left(\theta,\phi; m_{\theta,k|k-1}^{(i)}, m_{\phi,k|k-1}^{(i)}, P_{(\theta,\phi),k|k-1}^{(i)}\right), \tag{86}$$

assuming the process noise is additive Gaussian, and the survival probability $P_{S,k}$ is independent of target state.

Let us augment the state with a binary variable β so we may distinguish between the surviving components and the birth component, resulting in the following form for the predicted PHD

$$v_{k|k-1}(\theta,\phi,\beta) = \begin{cases} \sum_{i=1}^{J_{k-1}} w_{k|k-1}^{(i)} \mathcal{N}\left(\theta,\phi; m_{\theta,k|k-1}^{(i)}, m_{\phi,k|k-1}^{(i)}, P_{(\theta,\phi),k|k-1}^{(i)}\right), & \beta = 0 \\ w_{b,k}\mathcal{U}(\theta;\mathbb{B})\,\mathcal{N}\left(\phi;\bar{\phi},P_{\phi}\right), & \beta = 1 \end{cases} \tag{87}$$

where the predicted weights are $w_{k|k-1}^{(i)} = P_{S,k} w_{k-1}^{(i)}$. As noted in [171], separating the surviving and birth components using the β variable is necessary in order to avoid biasing the cardinality estimates.

UPDATE Let us assume that new targets are always detected at their time of birth. For simplicity, we also assume that the detection probability for surviving targets is independent of their state, which leads to the definition

$$P_{D,k}(\theta,\phi,\beta) = \begin{cases} P_{D,k}, & \beta = 0 \\ 1, & \beta = 1 \end{cases}. \tag{88}$$

Furthermore, we assume that the measurement likelihood function is Gaussian and defined by

$$g(z|\theta,\phi,\beta) = \mathcal{N}(z;\theta,P_{\theta}). \tag{89}$$

The posterior intensity is now given by (90) and (91), where, for the sake of brevity, the subscript $k|k-1$ is omitted for $w^{(i)}$, $\theta^{(i)}$, $m_{\theta}^{(i)}$,$m_{\phi}^{(i)}$ and $P_{(\theta,\phi)}^{(i)}$.

$$\begin{aligned} v_k(\theta,\phi,0) = {} & (1-P_{D,k}) \sum_{i=1}^{J_{k-1}} w^{(i)} \mathcal{N}\left(\theta,\phi; m_{\theta}^{(i)}, m_{\phi}^{(i)}, P_{(\theta,\phi)}^{(i)}\right) \\ & + \sum_{z\in Z} \sum_{i=1}^{J_{k-1}} \frac{P_{D,k}\mathcal{N}\left(z; m_{\theta}^{(i)}, P_{\theta}\right) w^{(i)} \mathcal{N}\left(\theta,\phi; m_{\theta}^{(i)}, m_{\phi}^{(i)}, P_{(\theta,\phi)}^{(i)}\right)}{\kappa_k(z) + \left\langle P_{D,k}(\cdot)\, g(z|\cdot), v_{k|k-1}(\cdot)\right\rangle}, \end{aligned} \tag{90}$$

$$v_k(\theta,\phi,1) = \sum_{z\in Z} \frac{\mathcal{N}(\theta;z,P_{\theta})\, w_{b,k}\mathcal{U}(\theta;\mathbb{B})\,\mathcal{N}\left(\phi;\bar{\phi},P_{\phi}\right)}{\kappa_k(z) + \left\langle P_{D,k}(\cdot)\, g(z|\cdot), v_{k|k-1}(\cdot)\right\rangle}. \tag{91}$$

In the denominators in (90) and (91) we have the following,

$$\left\langle P_{D,k}\left(\cdot\right)g\left(z|\cdot\right),v_{k|k-1}\left(\cdot\right)\right\rangle=\sum_{i=1}^{J_{k-1}}P_{D,k}w_{k|k-1}q_k^{(i)}\left(z\right)+w_k^b\int\mathcal{N}\left(\phi;\bar{\phi},P_\phi\right)d\phi\int\mathcal{N}\left(\theta;z,P_\theta\right)\frac{1_{\mathbb{B}}\left(\theta\right)}{V_{\mathbb{B}}}d\theta \tag{92}$$

where $q_k^{(i)}\left(z\right)=\mathcal{N}\left(z;m_{\theta,k|k-1}^{(i)},S_{k|k-1}^{(i)}\right)$ is the likelihood of measurement z against component i (with $S_{k|k-1}^{(i)}$ being the innovation covariance for component i), $1_{\mathbb{B}}\left(\cdot\right)$ is the indicator function for the region $\mathbb{B}$, and $V_{\mathbb{B}}$ is the volume of region $\mathbb{B}$. We now make the approximation

$$\mathcal{N}\left(\theta;z,P_\theta\right)1_{\mathbb{B}}\left(\theta\right)\approx\mathcal{N}\left(\theta;z,P_\theta\right) \tag{93}$$

which can be justified in practice by assuming that the standard deviation of the measurement noise Σ_θ is small compared to the surveillance region $\mathbb{B}$. This leads to the following approximation for the integral

$$\left\langle P_{D,k}\left(\cdot\right)g\left(z|\cdot\right),v_{k|k-1}\left(\cdot\right)\right\rangle\approx\sum_{i=1}^{J_{k-1}}w_{k|k-1}^{(i)}q_k^{(i)}\left(z\right)+\frac{w_{b,k}}{V_{\mathbb{B}}}. \tag{94}$$

Substituting (93) and (94) into (90) and (91) yields the following approximation for the posterior PHD, which is similar to the update equations given in [83]

$$v_k\left(\theta,\phi,0\right)\approx\sum_{i=1}^{J_{k-1}}w_{m,k}^{(i)}\mathcal{N}\left(\theta,\phi;m_{\theta,k|k-1}^{(i)},m_{\phi,k|k-1}^{(i)},P_{(\theta,\phi),k|k-1}^{(i)}\right)+\sum_{z\in Z}\sum_{i=1}^{J_{k-1}}w_{s,k}^{(i)}\mathcal{N}\left(\theta,\phi;m_{\theta,k}^{(i)},m_{\phi,k}^{(i)},P_{(\theta,\phi),k}^{(i)}\right), \tag{95}$$

$$v_k\left(\theta,\phi,1\right)\approx\sum_{z\in Z}w_{b,k}^{(i)}\mathcal{N}\left(\theta;z,P_\theta\right)\mathcal{N}\left(\phi;\bar{\phi},P_\phi\right), \tag{96}$$

where

$$w_{m,k}^{(i)}=\left(1-P_{D,k}\right)w_{k|k-1}^{(i)}, \tag{97}$$

$$w_{s,k}^{(i)}=\frac{P_{D,k}q_k^{(i)}\left(z\right)w_{k|k-1}^{(i)}}{\kappa_k\left(z\right)+\sum_{i=1}^{J_{k-1}}w_{k|k-1}^{(i)}q_k^{(i)}\left(z\right)+w_{b,k}/V_{\mathbb{B}}}, \tag{98}$$

$$w_{b,k}^{(i)}=\frac{w_{b,k}/V_{\mathbb{B}}}{\kappa_k\left(z\right)+\sum_{i=1}^{J_{k-1}}w_{k|k-1}^{(i)}q_k^{(i)}\left(z\right)+w_{b,k}/V_{\mathbb{B}}}. \tag{99}$$

Since newborn targets at time k become surviving targets at time $k+1$, the predicted PHD for survivals at time $k+1$ is obtained by propagating the sum of both the $\beta = 0$ and $\beta = 1$ components of the posterior PHD at time k to the next time step.

2.4.6.2 *GMCPHD Filter with Partially Uniform Birth (PUB-GMCPHD)*

The CPHD filter is a generalisation of the PHD filter that relaxes the assumption that the number of targets follows a Poisson distribution. To achieve this, the filter propagates the discrete probability distribution on the number of targets (or cardinality distribution), thereby allowing it to take on an arbitrary form. In this section, we derive an approximation of the GMCPHD filter using the PUB model, which we refer to as the PUB-GMCPHD filter.

PREDICTION Let the posterior PHD at time $k-1$ be a Gaussian mixture of the same form as in (82). Let ρ_{k-1} denote the posterior cardinality distribution at time $k-1$, and $\rho_{B,k}$ the prior cardinality distribution of spontaneous births at time k. Then, as per the standard GMCPHD filter, the predicted cardinality distribution at time k is given by the convolution [129]

$$\rho_{k|k-1}(n) = \sum_{j=0}^{n} \rho_{B,k}(n-j) \sum_{l=j}^{\infty} C_j^l \rho_{k-1}(l) \left(P_{S,k}\right)^j \left(1-P_{S,k}\right)^{l-1} \tag{100}$$

where $C_j^l = \frac{l!}{j!(l-j)!}$ is the binomial coefficient. The PHD of target births is the same as defined in (84), and the predicted PHD at time $k-1$ is the same as (85). As we did for the PUB-GMPHD filter, we again augment the state with the variable β to distinguish surviving components from the birth component, leading to the definition (87) for the predicted PHD.

UPDATE Using the definition in (88) for the detection probability, and making the following definitions to simplify the notation,

$$\chi = \frac{\left\langle \Upsilon_k^1\left[v_{k|k-1}, Z\right], \rho_{k|k-1}\right\rangle}{\left\langle \Upsilon_k^0\left[v_{k|k-1}, Z\right], \rho_{k|k-1}\right\rangle}, \tag{101}$$

$$\chi(z) = \frac{\left\langle \Upsilon_k^1\left[v_{k|k-1}, Z-\{z\}\right], \rho_{k|k-1}\right\rangle}{\left\langle \Upsilon_k^0\left[v_{k|k-1}, Z\right], \rho_{k|k-1}\right\rangle}, \tag{102}$$

the CPHD update can be expressed as

$$\rho_k(n) = \frac{\Upsilon_k^0\left[v_{k|k-1}, Z\right](n)}{\left\langle \Upsilon_k^0\left[v_{k|k-1}, Z\right], \rho_{k|k-1}\right\rangle}\rho_{k|k-1}(n), \tag{103}$$

$$\begin{aligned} v_k(\theta,\phi,0) &= (1-P_{D,k})\,\chi v_{k|k-1}(\theta,\phi,0) \\ &\quad + \sum_{z\in Z}\psi_{k,z}(\theta,\phi,0)\,\chi(z)\,v_{k|k-1}(\theta,\phi,0), \end{aligned} \tag{104}$$

$$v_k(\theta,\phi,1) = \sum_{z\in Z}\psi_{k,z}(\theta,\phi,1)\,\chi(z)\,v_{k|k-1}(\theta,\phi,1), \tag{105}$$

where

$$\Upsilon_k^u[v,Z](n) = \sum_{j=0}^{\min(|Z|,n)}(|Z|-j)!\rho_{K,k}(|Z|-j)\,P_{j+u}^n \tag{106}$$

$$\times \frac{\langle (1-P_{D,k}), v\rangle^{n-(j+u)}}{\langle 1, v\rangle^n} e_j(\Xi_k(v,Z)), \tag{107}$$

$$\psi_{k,z}(\theta,\phi,\beta) = \frac{\langle 1,\kappa_k\rangle}{\kappa_k(z)} g(z|\theta,\phi)\,P_{D,k}(\theta,\phi,\beta), \tag{108}$$

$$\Xi_k(v,Z) = \{\langle v,\psi_{k,z}\rangle : z\in Z\}, \tag{109}$$

$$P_{j+u}^n = \frac{n!}{(n-(j+u))!}. \tag{110}$$

In the above, $\rho_{K,k}(\cdot)$ is the cardinality distribution of clutter at time k, and $e_j(\cdot)$ is the elementary symmetric function of order j. The inner product between $v_{k|k-1}$ and $\psi_{k,z}$ is given by

$$\left\langle v_{k|k-1}, \psi_{k,z}\right\rangle = \frac{\langle 1,\kappa_k\rangle}{\kappa_k(z)}\left(\frac{w_{b,k}}{V_{\mathbb{B}}} + P_{D,k}\sum_{i=1}^{J_{k|k-1}} w_{k|k-1}^{(i)} q_k^{(i)}(z)\right). \tag{111}$$

Hence, the elementary symmetric functions are calculated over the set

$$\Xi_k\left(v_{k|k-1}, Z\right) = \left\{\frac{\langle 1,\kappa_k\rangle}{\kappa_k(z)}\left(\frac{w_{b,k}}{V_{\mathbb{B}}} + P_{D,k}\sum_{i=1}^{J_{k|k-1}} w_{k|k-1}^{(i)} q_k^{(i)}(z)\right) : z\in Z\right\}. \tag{112}$$

The remaining inner products required in the CPHD calculations are

$$\left\langle 1, v_{k|k-1}\right\rangle = \left\langle 1, w_{k|k-1}\right\rangle + w_k^b, \tag{113}$$

$$\left\langle 1-P_{D,k}, v_{k|k-1}\right\rangle = (1-P_{D,k})\left\langle 1, w_{k|k-1}\right\rangle, \tag{114}$$

which results in the following equation for Υ_k^u,

$$\Upsilon_k^u\left[v_{k|k-1},Z\right](n)=\sum_{j=0}^{\min(|Z|,n)}(|Z|-j)!P_{K,k}(|Z|-j)(1-P_D)^{n-(j+u)}\times\frac{\left\langle 1,w_{k|k-1}\right\rangle^{n-(j+u)}}{\left(\left\langle 1,w_{k|k-1}\right\rangle+w_{b,k}\right)^n}P_{j+u}^n e_j\left(\Xi_k\left(v_{k|k-1},Z\right)\right). \tag{115}$$

Finally, the posterior PHD is approximated by

$$v_k(\theta,\phi,0)\approx\sum_{i=1}^{J_{k|k-1}}w_{m,k}^{(i)}\mathcal{N}\left(\theta,\phi;m_{\theta,k|k-1}^{(i)},m_{\phi,k|k-1}^{(i)},P_{(\theta,\phi),k|k-1}^{(i)}\right)+\sum_{z\in Z}\sum_{i=1}^{J_{k|k-1}}w_{s,k}^{(i)}(z)\mathcal{N}\left(\theta,\phi;m_{\theta,k}^{(i)},m_{\phi,k}^{(i)},P_{(\theta,\phi),k}^{(i)}\right), \tag{116}$$

$$v_k(\theta,\phi,1)\approx\sum_{z\in Z}w_{b,k}(z)\mathcal{N}(\theta;z,P_\theta)\mathcal{N}\left(\phi;\bar{\phi},P_\phi\right), \tag{117}$$

where

$$w_{m,k}^{(i)}=(1-P_{D,k})w_{k|k-1}^i\chi, \tag{118}$$

$$w_{s,k}^{(i)}(z)=\frac{\langle 1,\kappa_k\rangle}{\kappa_k(z)}P_{D,k}w_{k|k-1}^{(i)}q_k^{(i)}(z)\chi(z), \tag{119}$$

$$w_{b,k}(z)=\frac{\langle 1,\kappa_k\rangle}{\kappa_k(z)}\frac{w_{b,k}}{V_{\mathbb{B}}}\chi(z), \tag{120}$$

which is a Gaussian mixture as required. As we did for the PHD filter, the $\beta=0$ and $\beta=1$ components of the posterior PHD at time k are added together in the prediction at time $k+1$.

The calculation of the posterior cardinality distribution remains unchanged from the conventional GMCPHD filter, except for the new definition of $\Upsilon_k^u\left[v_{k|k-1},Z\right](n)$. Hence, the posterior cardinality distribution is given by (103).

2.4.6.3 *Performance of PUB-GM(C)PHD Filters*

In this section, we present simulation results demonstrating the performance of the PUB-GMPHD/CPHD filters applied to single-sensor bearings-only multi-target filtering. This is an important problem, especially in the context of covert surveillance, because bearing measurements often constitute the only reliable output from sensors which operate in passive mode. It is challenging due to the non-linear measurement model, and the potential for poor observability of targets. The effect of observability in bearings-only target motion analysis has been studied in [153, 85, 33], but for

the case of bearings-only multi-target tracking in clutter, the effects are difficult to characterise in a theoretical context.

To analyse the performance of the PUB-GM(C)PHD filters, we compare them with the corresponding conventional filters under Gaussian mixture birth models with various numbers of components. We simulate a scenario consisting of a single bearings-only sensor on board a manoeuvring platform. The position of this platform is known at all times and denoted in Cartesian coordinates as $\left(x_k^{(s)}, y_k^{(s)}\right)$ for time index k. There are up to six targets in the sensor's detection region, each following a nearly constant velocity motion model, defined as follows. At time k, let target t have Cartesian position coordinates $\left(x_k^{(t)}, y_k^{(t)}\right)$ and instantaneous velocity vector $\left(\dot{x}_k^{(t)}, \dot{y}_k^{(t)}\right)$, such that the target's state vector can be defined as

$$\tilde{x}_k^{(t)} = \begin{bmatrix} x_k^{(t)} & y_k^{(t)} & \dot{x}_k^{(t)} & \dot{y}_k^{(t)} \end{bmatrix}.$$

For notational convenience, we also define a similar vector $\tilde{x}_k^{(s)}$ for the known state of the sensor platform. The target state evolves according to the discrete-time transition

$$\tilde{x}_{k+1}^{(t)} = F\tilde{x}_k^{(t)} + \Gamma v_k \tag{121}$$

$$F = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \otimes I_2, \qquad \Gamma = \begin{bmatrix} T^2/2 \\ T \end{bmatrix} \otimes I_2, \tag{122}$$

where T is the time between measurements, and $v_k \sim \mathcal{N}(0, Q)$ is a 2×1 i.i.d Gaussian process noise vector with $Q = \sigma_v^2 I_2$ where $\sigma_v = 0.005$ m/s^2 is the standard deviation of the process noise acceleration.

The sensor measures target t bearing according to the model

$$z_k^t = h\left(\tilde{x}_k^{(t)}, \tilde{x}_k^{(s)}\right) + w_k \tag{123}$$

$$h\left(\tilde{x}_k^{(t)}, \tilde{x}_k^{(s)}\right) = \arctan\left(\frac{x_k^{(t)} - x_k^{(s)}}{y_k^{(t)} - y_k^{(s)}}\right) \tag{124}$$

where $w_k \sim \mathcal{N}\left(0, \sigma_\theta^2\right)$ is zero-mean white Gaussian measurement noise with standard deviation $\sigma_\theta = 1^\circ$. The sensor generates one set of measurements every 10 seconds, and the detection probability is 0.95 for all targets. The measurement sets also contain false alarms distributed uniformly across the bearing space, the number of which follows a Poisson distribution with a known mean of 25 per scan. Three targets are

present at the beginning of the scenario, with two arriving at time 300s, and one more at time 600s. One target leaves at 2200s, and another leaves at 2600s. The scenario geometry and an example of the measurements processed by the filter are shown in Figure 1.

To handle the non-linear measurement model, all filters use local linearisation based on the extended Kalman filter (EKF). Clearly, other types of non-linear filtering algorithms may be used in place of the EKF, such as the Unscented Kalman filter [96], Cubature Kalman filter [5], or Shifted Rayleigh filter [41], but comparing the performance of these algorithms is beyond the scope of this paper. In these simulations, the expected number of new targets per scan is set to 0.05, the Malahanobis distance threshold for merging components in the Gaussian mixture is set to 2, and the weight threshold for elimination of mixture components is 10^{-5}. All performance metrics shown are averaged over 500 Monte Carlo runs, with fixed ground truth and independent measurement realisations.

The results of both PUB-GMPHD and PUB-GMCPHD are compared against the corresponding filters with Gaussian mixture birth models with different numbers of components. To construct the GM birth models, we position the means of the Gaussians evenly across the bearing space, at a nominal range of 12 km from the sensor with a range standard deviation of 4 km. The bearing standard deviation is set such that the gaps between the means are adequately covered. For this analysis, we have tested the filters with various GM birth models, where the number of Gaussians components ranges from 4 up to 1024. We refer to these models as GMX, where X is the number of Gaussians. For the GM4 model, a bearing standard deviation of 40° is used, GM16 uses 10°, GM64 uses 2.5°, and GM256 and GM1024 both use 1°.

Figures 2a and 2b illustrate two of the GM birth models tested, and Figure 2c shows the birth model with uniform bearing distribution. From the figures, the GM16 and uniform birth models appear very similar, and one would expect their performance to be almost the same. However, the results will show that they exhibit very different performance, which is due to the diffuseness of the components in the GM16 representation, and the effect this has on the non-linear measurement update.

Figure 3 shows the optimal sub-pattern assignment (OSPA) distance [173] (with order 2 and cutoff 4 km) for the PUB-GM(C)PHD filters. As expected, we find that CPHD outperforms PHD, which is due to the reduced variability in the cardinality estimates. Figure 4 shows the average increase in the OSPA for each GM birth model

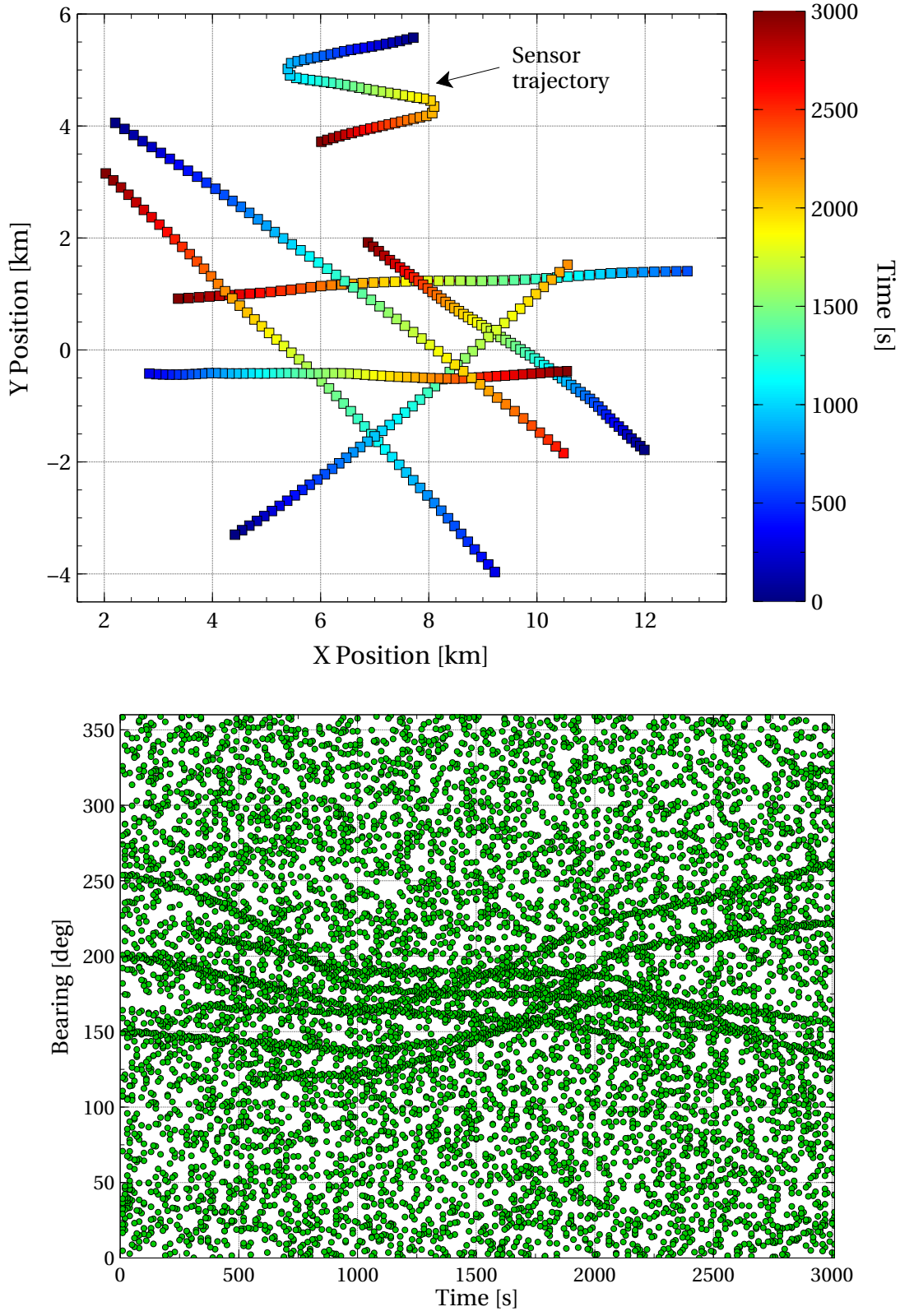
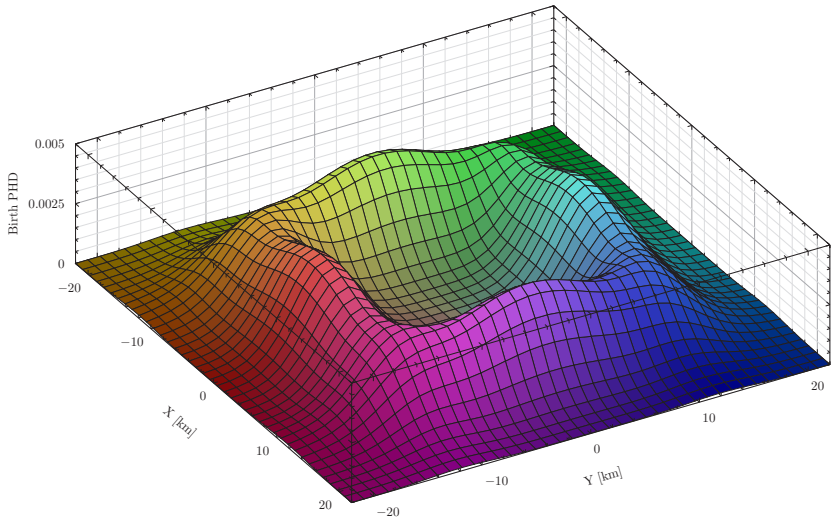
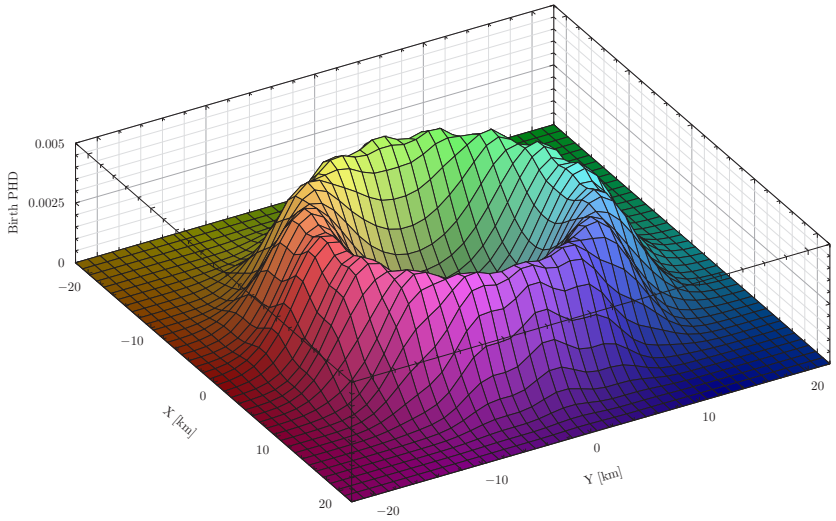


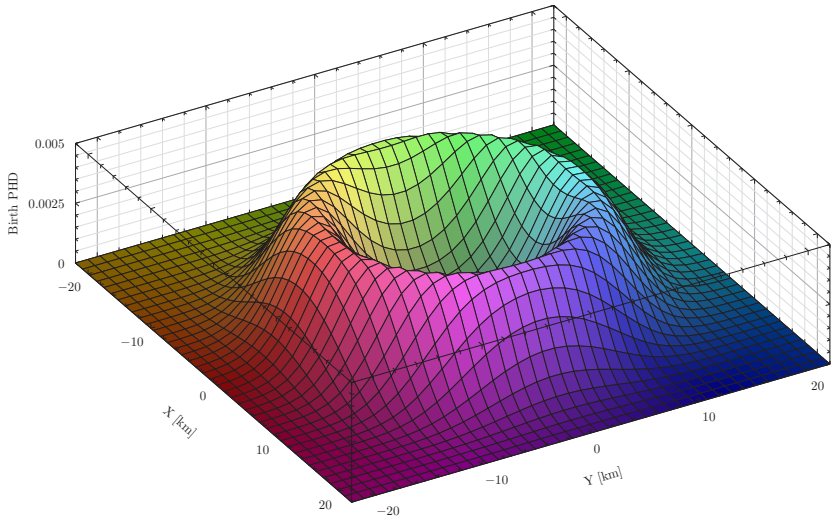
Figure 1: Target-observer geometry and example measurement set for PUB-GM(C)PHD test scenario.



(a) GM4 birth model.



(b) GM16 birth model.



(c) Partially uniform birth model.

Figure 2: Birth models used for testing PUB-GM(C)PHD filters.

compared to the PUB model, expressed as a percentage of the OSPA for the PUB model. This is computed at all times k using the formula

$$\Delta_{OSPA}(k) = \frac{100}{N_{MC}} \sum_{i=1}^{N_{MC}} \frac{O^{(i)}_{\text{GMX}}(k) - O^{(i)}_{PUB}(k)}{O^{(i)}_{PUB}(k)} \tag{125}$$

where N_{MC} is the number of Monte Carlo runs, $O^{(i)}_{\text{GMX}}(k)$ is the OSPA at time k on run i using the GMX birth model, and $O^{(i)}_{PUB}(k)$ is the OSPA at time k on run i using the PUB model.

Table 1 summarises the results by showing the time averaged performance under each birth model. The first column ($\bar{\Delta}_{OSPA}$) contains the time averaged percentage increase in OSPA under the GM birth model compared to the PUB model, calculated by averaging (125) over all k. The second column (Exec Tme) contains the average execution time, expressed as a multiple of the execution time under the PUB model.

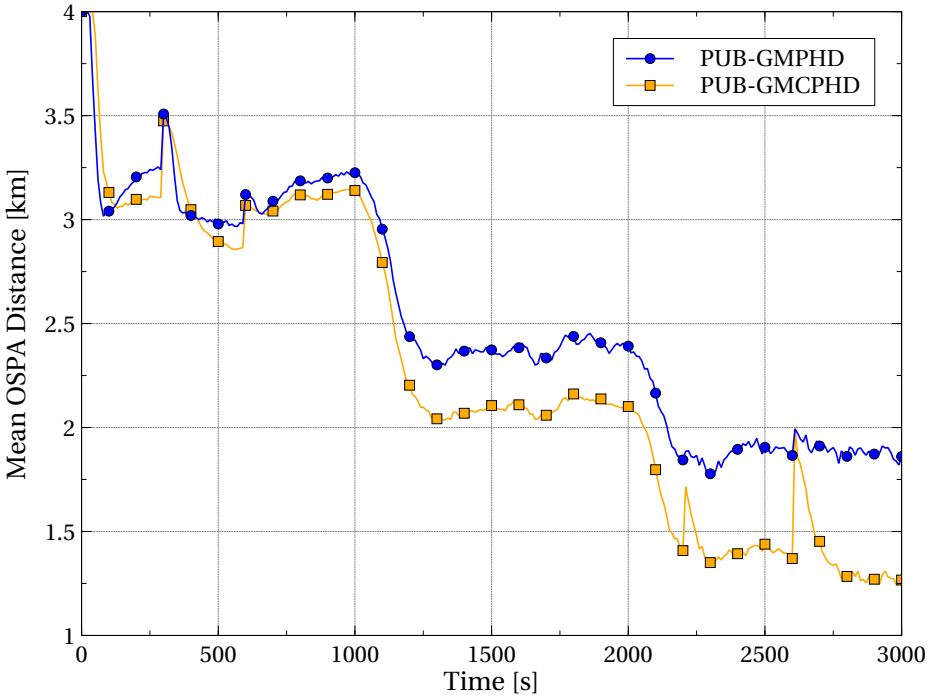


Figure 3: OSPA distance for PUB-GMPHD and PUB-GMCPHD filters.

From Figure 4 and Table 1 it is clear that using the GM4 and GM16 models to represent the PHD of new targets has led to higher estimation error than the filter with uniform birth. This is mainly due to worse localisation performance, which is induced by the poor initialisation afforded by the birth model. This can be explained by the fact that when the birth PHD components are updated using the non-linear measurement model, the more diffuse Gaussians suffer from higher approximation

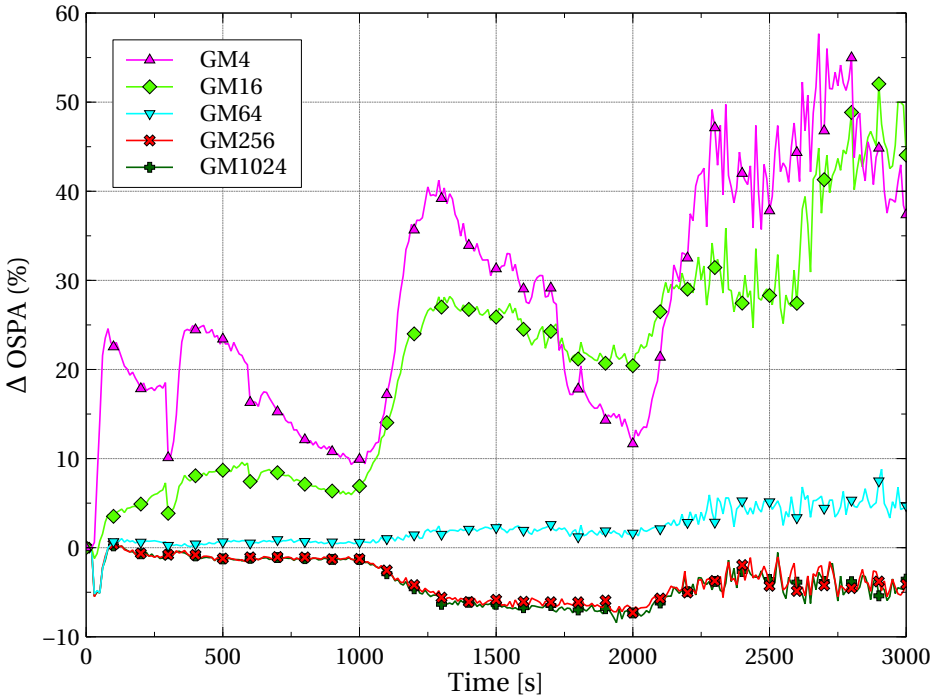


Figure 4: Percentage OSPA increase for GMPHD filter with GM birth models compared to the PUB model.

Table 1: GM(C)PHD filter performance with different birth models.

Birth Model	GMPHD		GMCPHD	
	$\bar{\Delta}_{OSPA}$ (%)	Exec Tme	$\bar{\Delta}_{OSPA}$ (%)	Exec Tme
PUB	0	1	0	1
GM4	27.53	3.41	46.70	2.82
GM16	21.03	1.02	34.14	1.00
GM64	2.09	1.39	4.79	1.26
GM256	-3.51	2.40	-3.07	2.15
GM1024	-3.77	6.20	-4.80	4.20

errors. The Gaussian components in the GM4 and GM16 models are necessarily more diffuse, and as a result, they suffer form these errors to a much greater extent than the GM64, GM256 and GM1024 models.

The higher execution time for the GM4 model is due to the fact that the filter retains the diffuse Gaussians from the birth PHD for longer before they are eliminated from the mixture, thereby increasing the average number of components in the posterior. This effect occurs in bearings-only tracking because when the bearing component of the initial covariance is very large, it takes several update cycles to reduce to a value closer to the variance of the measured bearings. This delay in convergence means that

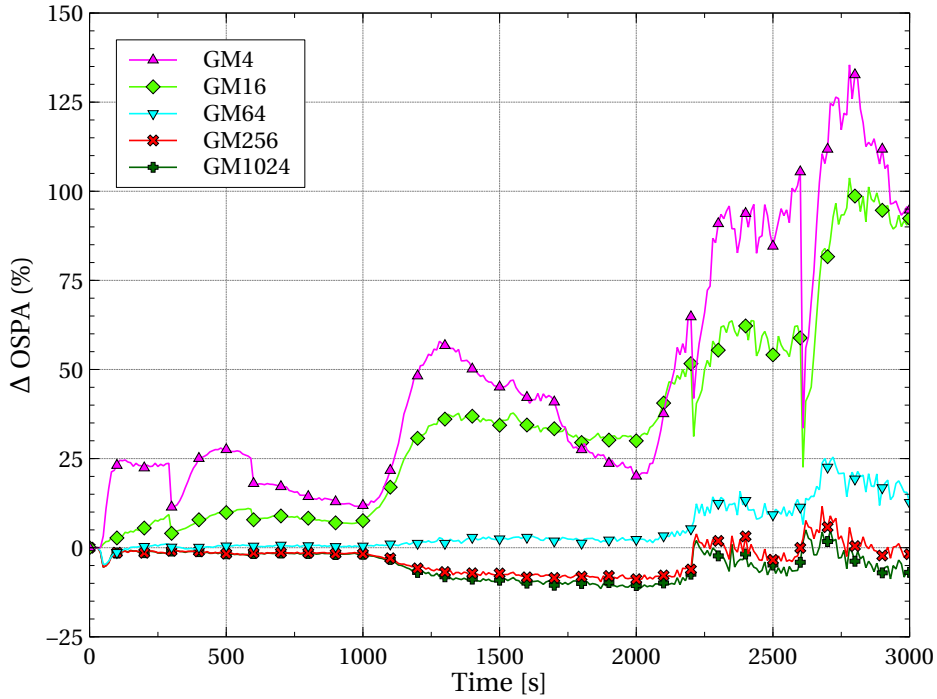


Figure 5: Percentage OSPA increase for GMCPHD filter with GM birth models compared to the PUB model.

there is more opportunity for false alarms to confirm the presence of a target, hence it takes longer for false components to be discarded.

As the number of Gaussians in the birth model increases, the average OSPA improves, as can be seen from Figure 4 and Table 1. The GM64 model performs slightly worse than the uniform model, and GM256 performs slightly better, however, a further increase to 1024 components provides very marginal improvement, with a significant increase in computational cost. Note that the differences in Δ_{OSPA} percentages between the PHD and CPHD filters are due to the difference in absolute error between the filters. For the uniform birth model, the PHD filter has higher error than the CPHD (see Figure 3), which leads to lower percentage differences than for the CPHD filter.

From these results we can conclude that using the uniform target birth model avoids the potential computational overhead that can occur when using a GM birth model with either too many or too few components. The uniform birth model also provides error performance which is similar to using a GM model with many components, with some small penalty incurred by approximating the posterior as a Gaussian mixture.

2.4.7 *GMCPHD Filtering with Unknown Clutter Rate*

A great deal of research effort has been devoted to multi-target filtering in the presence of clutter, however, it is often assumed that the clutter density is a known parameter that can be provided as prior information. This is unrealistic for real-world applications, as it is often impossible to know this accurately in advance, and its value is often dependent on environmental conditions which may vary over time. It is therefore important to investigate trackers which do not require the clutter density to be specified, and can instead estimate it on-line.

While the original CPHD filter [129, 199] requires knowledge of the clutter density, it was demonstrated in [133] that this filter can also be used to jointly estimate the clutter density and target states. This allows filtering in the presence of an unknown and time-varying clutter background, a technique referred to as λ-CPHD filtering. This method uses the concept of 'clutter targets' to explain the presence of clutter measurements. Just like real targets, clutter targets are modelled with a detection probability, survival probability, and birth rate. However, since the cardinality distribution now includes both real and clutter targets, there is no way to obtain the cardinality distribution for only the real targets using this method. For this reason, the performance of the λ-CPHD filter is significantly worse than that of the conventional CPHD filter if the clutter density is accurately known. To improve upon this, we propose a simple bootstrap filtering scheme in which we use the λ-CPHD filter to estimate only the clutter density. This is subsequently passed to a conventional CPHD filter, that estimates the PHD and cardinality distribution for real targets.

Some traditional techniques for tracking with unknown clutter density include non-parametric versions of probabilistic data association (PDA) [11] and integrated PDA (IPDA) [151] for the case of a single target in clutter, and non-parametric joint PDA (JPDA) [10] and joint integrated PDA (JIPDA) [148] for multiple targets in clutter. The drawback of JPDA and JIPDA is that their computational complexity increases exponentially as the target density increases, making them unsuitable in many situations. Some other independent techniques for on-line clutter rate estimation have also been proposed in [118] and [177].

2.4.7.1 *λ-GMCPHD Filter*

The section describes the operation of the λ-GMCPHD filter, which was proposed in [133] as a method of dealing with cases in which the clutter rate is unknown and possibly time-varying.

We begin by defining some notation. Let $\tilde{v}_k(\cdot)$ denote the posterior PHD of a composite multi-target state comprising both real targets and clutter targets, and let $\tilde{\rho}_k(\cdot)$ denote the cardinality distribution of this state. The composite PHD $\tilde{v}_k(\cdot)$ can be decomposed as

$$\tilde{v}_k(\tilde{x}) = \begin{cases} v_k^{(1)}(x), & \tilde{x} = x \\ v_k^{(0)}(c), & \tilde{x} = c \end{cases} \tag{126}$$

where $v_k^{(1)}(\cdot)$ is the PHD for real targets, and $v_k^{(0)}(\cdot)$ is the PHD for clutter. As in the conventional GMCPHD filter, we represent $v_k^{(1)}(\cdot)$ using a Gaussian mixture, and the probabilities of detection and survival for real targets are denoted $P_D^{(1)}$ and $P_S^{(1)}$ respectively. The PHD for clutter targets $v_k^{(0)}(\cdot)$ is completely characterised by the number of clutter targets, which we denote as $N_k^{(0)}$.

Clutter is modelled in terms of the following parameters; the mean number of clutter target births per frame $N_{B,k}^{(0)}$, the probability of clutter target survival $P_{S,k}^{(0)}$, the probability of clutter target detection $P_{D,k}^{(0)}$, and the spatial likelihood $\alpha_k^{(0)}$. The choice of values for these parameters is governed by the desired filter response. Lower values of $P_{D,k}^{(0)}$ lead to reduced variance in the clutter rate estimates, at the expense of a longer response time to changes in the true clutter rate. It can also be shown that if the true clutter rate is constant, the filter will give the most accurate estimates when that rate is $N_{B,k}^{(0)} P_{D,k}^{(0)} / \left(1 - P_{S,k}^{(0)}\right)$.

PREDICTION At time $k-1$, let $\tilde{\rho}_{k-1}$ denote the posterior cardinality distribution, $v_{k-1}^{(1)}$ the posterior PHD of the form

$$v_{k-1}^{(1)}(x) = \sum_{i=1}^{J_{k-1}} w_{k-1}^{(i)} \mathcal{N}\left(x; m_{k-1}^{(i)}, P_{k-1}^{(i)}\right), \tag{127}$$

and $N_{k-1}^{(0)}$ the posterior number of clutter generators. Then the predicted PHD at time k is given by

$$v_{k|k-1}(x) = v_{S,k|k-1}(x) + \gamma_k(x), \tag{128}$$

where

$$v_{S,k|k-1}(x) = P_{S,k}^{(1)} \sum_{i=1}^{J_{k-1}} w_{k-1}^{(i)} \mathcal{N}\left(x; m_{k|k-1}^{(i)}, P_{k|k-1}^{(i)}\right), \tag{129}$$

$$\gamma_k(x) = \sum_{i=1}^{J_{B,k}} w_{B,k}^{(i)} \mathcal{N}\left(x; m_{B,k}^{(i)}, P_{B,k}^{(i)}\right), \tag{130}$$

$$m_{k|k-1}^{(i)} = F m_{k-1}^{(i)}, \tag{131}$$

$$P_{k|k-1}^{(i)} = F P_{k-1}^{(i)} F + Q, \tag{132}$$

in which $J_{B,k}$ is the number of components in the Gaussian mixture birth PHD at time k, and $w_{B,k}^{(i)}$, $m_{B,k}^{(i)}$ and $P_{B,k}^{(i)}$ are the weight, mean and covariance of the i-th component in the birth PHD respectively. The predictions for the number of clutter generators, clutter rate, and cardinality distribution are given by

$$N_{k|k-1}^{(0)} = N_{B,k}^{(0)} + P_{S,k}^{(0)} N_{k-1}^{(0)}, \tag{133}$$

$$\lambda_{k|k-1}^{(0)} = N_{k|k-1}^{(0)} P_{D,k}^{(0)}, \tag{134}$$

$$\tilde{\rho}_{k|k-1}(\tilde{n}) = \sum_{j=0}^{\tilde{n}} \tilde{\rho}_{B,k}(\tilde{n} - j) \sum_{l=j}^{\infty} C_l^j \tilde{\rho}_{k-1}(l) (1-\phi)^{l-j} \phi^j, \tag{135}$$

where

$$C_l^j = \frac{j!}{l!\,(j-l)!}, \tag{136}$$

$$\phi = \frac{P_{S,k}^{(1)} \sum_{i=1}^{J_{k-1}} w_{k-1}^{(i)} + P_{S,k}^{(0)} N_{k-1}^{(0)}}{\sum_{i=1}^{J_{k-1}} w_{k-1}^{(i)} + \sum_{i=1}^{J_{B,k}} w_{B,k}^{(i)} + N_{k-1}^{(0)}}. \tag{137}$$

UPDATE Given the predictions in (129), (133), (134), and (135) we may calculate the posterior PHD, number of clutter generators, and cardinality distribution by

$$\begin{aligned} v_k^{(1)}(x) = & \sum_{i=1}^{J_{k|k-1}} w_{k,m}^{(i)} \mathcal{N}\left(x; m_{k|k-1}^{(i)}, P_{k|k-1}^{(i)}\right) \\ & + \sum_{z \in Z} \sum_{i=1}^{J_{k|k-1}} w_{k,S}^{(i)}(z) \mathcal{N}\left(x; m_k^{(i)}(z), P_k^{(i)}(z)\right) \end{aligned} \tag{138}$$

$$N_k^{(0)} = N_{k|k-1}^{(0)} \left(1 - P_{D,k}^{(0)}\right) \tilde{\chi} + \sum_{z \in Z} \frac{\lambda_{k|k-1}^{(0)} \alpha_k(z)}{D_1(z)} \tag{139}$$

$$\tilde{\rho}_k(\tilde{n}) = \begin{cases} 0, & \tilde{n} < |Z| \\ \frac{\tilde{\rho}_{k|k-1}(\tilde{n}) \tilde{\Upsilon}_k^{(0)}\left[\tilde{v}_{k|k-1}, Z\right](\tilde{n})}{\left\langle \tilde{\rho}_{k|k-1}, \tilde{\Upsilon}_k^{(0)}\left[\tilde{v}_{k|k-1}, Z\right] \right\rangle}, & \tilde{n} \geq |Z| \end{cases} \tag{140}$$

The weights of the Gaussian components in the updated mixture are calculated by

$$w_{m,k}^{(i)} = \left(1 - P_{D,k}^{(1)}\right) \tilde{\chi} w_{k|k-1}^{(i)}, \tag{141}$$

$$w_{S,k}^{(i)}(z) = \frac{P_{D,k}^{(1)} w_{k|k-1}^{(i)} q_k^{(i)}(z)}{D_1(z)}, \tag{142}$$

where,

$$D_1(z) = \lambda_{k|k-1}^{(0)} \alpha_k(z) + P_{D,k}^{(1)} \sum_{i=1}^{J_{k|k-1}} w_{k|k-1}^{(i)} q_k^{(i)}(z), \tag{143}$$

$$\tilde{\chi} = \frac{1}{D_1} \frac{\left\langle \tilde{\Upsilon}_k^{(1)}\left[\tilde{v}_{k|k-1}, Z\right], \tilde{\rho}_{k|k-1} \right\rangle}{\left\langle \tilde{\Upsilon}_k^{(0)}\left[\tilde{v}_{k|k-1}, Z\right], \tilde{\rho}_{k|k-1} \right\rangle}, \tag{144}$$

$$\tilde{\Upsilon}_k^{(u)}\left[\Phi_{k|k-1}, Z\right](\tilde{n}) = \begin{cases} 0, & \tilde{n} < |Z| + u \\ P_{|Z|+u}^{\tilde{n}} \Phi_{k|k-1}^{\tilde{n}-(|Z|+u)}, & \tilde{n} \geq |Z| + u \end{cases}, \tag{145}$$

$$\Phi_{k|k-1} = 1 - \frac{1}{D_2} \left(P_{D,k}^{(1)} \sum_{i=1}^{J_{k|k-1}} w_{k|k-1}^{(i)} + \lambda_{k|k-1}^{(0)} \right), \tag{146}$$

$$D_2 = \sum_{i=1}^{J_{k|k-1}} w_{k|k-1}^{(i)} + N_{k|k-1}^{(0)}, \tag{147}$$

$$P_{|Z|+u}^{\tilde{n}} = \frac{\tilde{n}!}{(\tilde{n} - (|Z| + u))!}. \tag{148}$$

2.4.7.2 *Bootstrap GMCPHD Filter*

The λ-GMCPHD filter does not propagate the cardinality distribution of real targets, but the joint cardinality distribution of real and clutter targets. This leads to increased variability in the estimated number of real targets compared to the conventional GM-CPHD filter, which propagates the cardinality distribution of real targets only. By running a λ-GMCPHD and conventional GMCPHD filter in parallel, we can utilise the strengths of both filters to improve the overall performance. A simple way to do this is to use λ-GMCPHD to estimate the clutter density only, and the conventional GMCPHD to estimate the intensity and cardinality distribution of real targets.

On each scan, the λ-GMCPHD filter is updated first, and its estimate of the current clutter density calculated as

$$\lambda_{C,k} = N_k^{(0)} P_{D,k}^{(0)}. \tag{149}$$

This is then used in the update of the conventional GMCPHD filter, as illustrated in Figure 6.

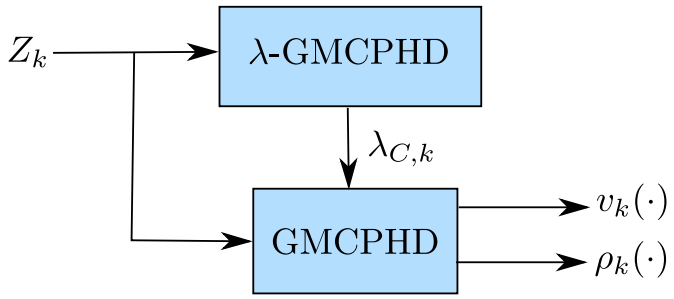


Figure 6: Block diagram of the bootstrap GMCPHD filter.

2.4.7.3 *Performance of the Bootstrap GMCPHD Filter*

In this section we examine the performance of the three versions of the GMCPHD filter applied to a bearings-only multi-target tracking problem. The bootstrap GMCPHD filter is compared to the λ-GMCPHD and the conventional GMCPHD filter with various fixed clutter rate parameters. The performance of each filter is analysed by carrying out 500 Monte Carlo runs and plotting the average OSPA distance [173] between the estimated and true states, with order 1 and cutoff 4km.

The scenario consists of a single bearings-only sensor on board a manoeuvring platform, and up to five targets with nearly constant velocity ($\sigma_v = 0.005\text{m/s}^2$). Two targets are present at the beginning, with another three appearing shortly thereafter, and two disappearing late in the scenario. The detection probability and measurement noise are both known constants, with values of 0.95 and 1° respectively. The number of clutter measurements is Poisson, and the spatial distribution is uniform across all possible bearings. Two different clutter backgrounds are tested, one with a constant rate of 30, and another with a variable rate starting at 20 and increasing to 40 during the scenario. The scenario geometry and an example of the measurements for the variable clutter rate case are shown in Figure 7.

For the case of constant clutter rate, the performance of the bootstrap and λ-GMCPHD filters are compared with the conventional GMCPHD filter. The latter is tested under three different clutter parameter settings; 30 (matched), 40 (too high), and 20 (too low). For the λ-GMCPHD filter, parameter values of $P_{D,k}^{(0)} = 0.5$, $P_{S,k}^{(0)} =$

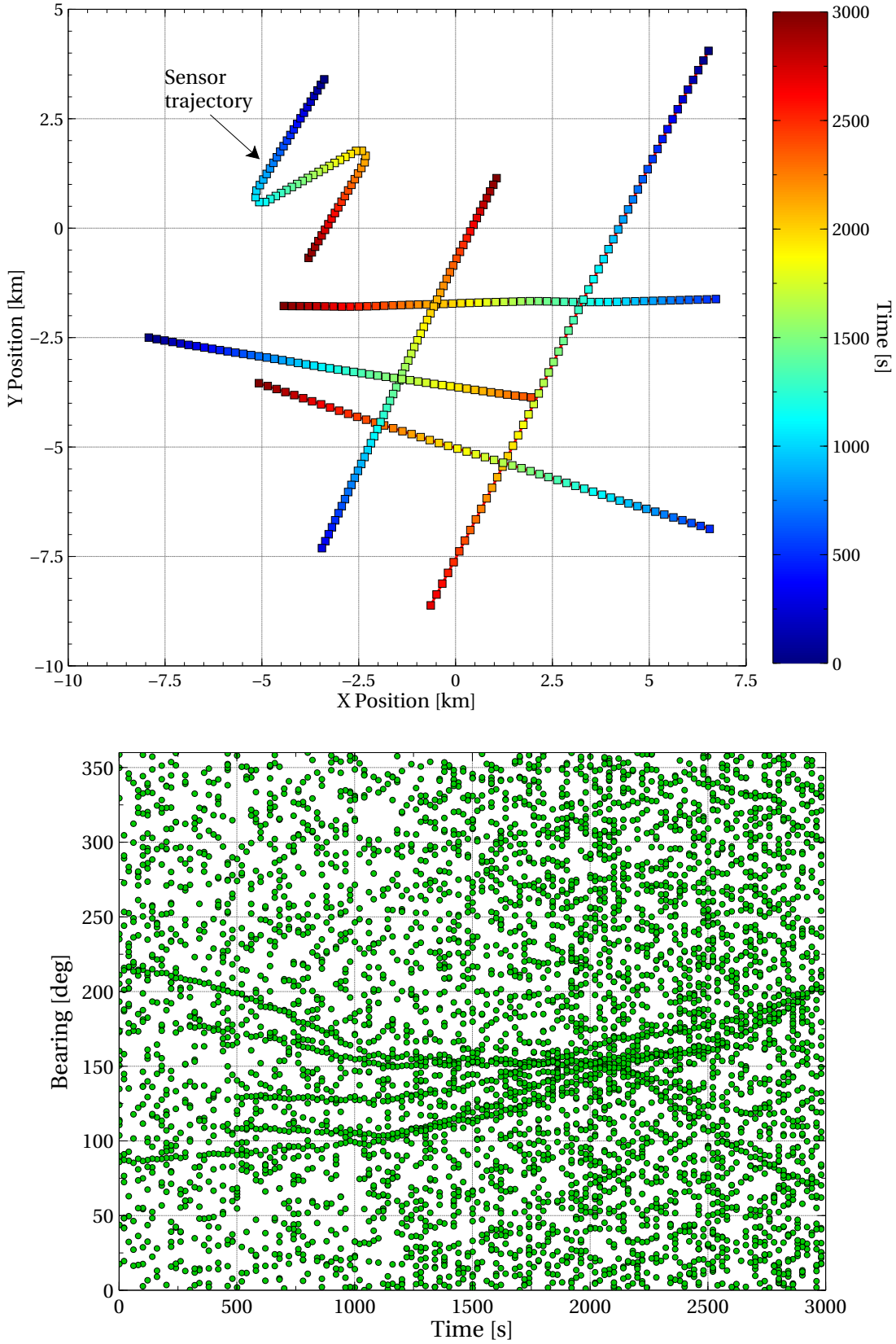


Figure 7: Target-observer geometry and example measurement set for bootstrap GMCPHD filter test scenario. The measurements shown are for the case of time-varying clutter rate.

0.98, and $N_{B,k}^{(0)} = 1$ were used. This is tuned to achieve best performance for a clutter rate of 25, which is slightly offset from the true rate of 30.

The OSPA and clutter rate estimates/parameters for the constant case are shown in Figure 8. As expected, the matched filter performs best, and the unmatched filters perform worse due to the bias introduced by the incorrect clutter rate setting. When the clutter rate is set too low, the filter responds more quickly to new target appearance, but more slowly to target disappearance, because the filter is more likely to explain measurements as originating from targets instead of clutter. The opposite effect occurs when the clutter rate parameter is set too high. This can be seen in Figure 9, which shows the cardinality estimation performance for the conventional GMCPHD filter under different clutter parameter settings. As expected, the cardinality is biased low when the clutter rate is set too high, biased high when the clutter rate is set too low, and unbiased when the clutter rate is matched to the truth.

Interestingly, the λ-GMCPHD alone performs no better on average than the unmatched filters, due to the lack of cardinality information for real targets. From Figure 10, it is clear that the λ-GMCPHD filter exhibits higher cardinality variance than the conventional GMCPHD filter. The bootstrap GMCPHD filter resolves this problem, with much lower cardinality variance, and OSPA performance very similar to the matched GMCPHD filter.

For the variable clutter rate scenario, the two adaptive methods are compared with two parameterisations of the conventional GMCPHD filter, one of which assumes a fixed clutter rate of 30, and the other is the ideal but unrealistic case in which the clutter rate is matched to the exact true value on each scan. For the λ-GMCPHD filter, the clutter model parameters are the same as the constant clutter rate case.

Figure 11 shows the OSPA for the four filters and the clutter rate estimation performance, and Figure 12 shows the cardinality estimation performance. The fixed-parameter GMCPHD filter performs poorly, as its cardinality estimates are biased low at the beginning and high at the end, due to the change in the underlying clutter rate. As before, due to the lack of target cardinality information, the λ-GMCPHD alone performs no better on average than the unmatched conventional GMCPHD, and exhibits increased cardinality variance. The bootstrap GMCPHD filter gives a significant improvement over the unmatched filter, with performance very similar to the ideal matched GMCPHD filter. The reason for this is that the λ-GMCPHD fil-

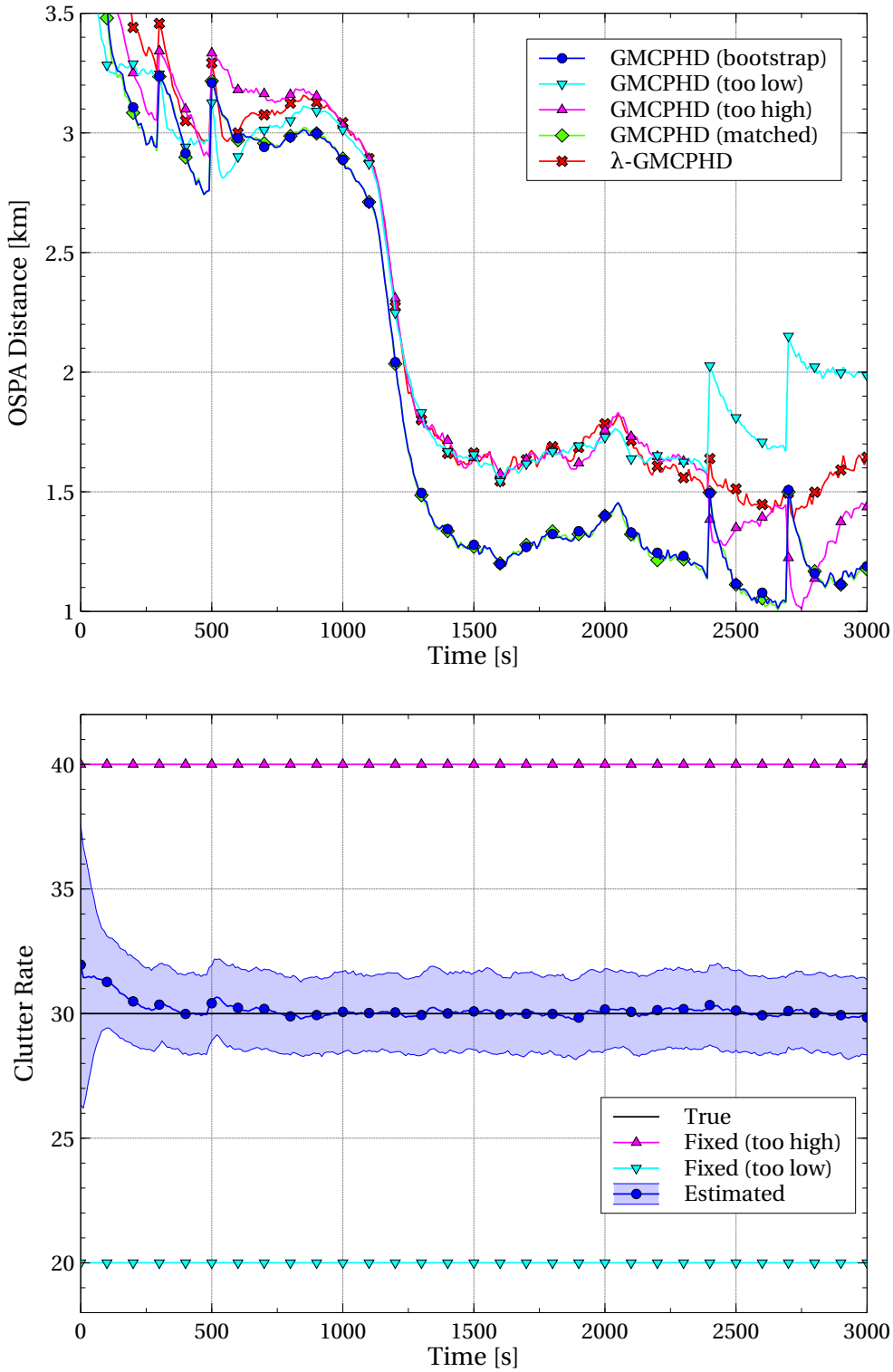


Figure 8: Constant clutter rate scenario, OSPA and clutter rate results for GMCPHD filters. In the clutter rate plot the "fixed" lines show the parameter settings for the corresponding GMCPHD filter, and the "estimated" line shows the estimates generated by the λ-GMCPHD filter, which is also used by the bootstrap GMCPHD filter.

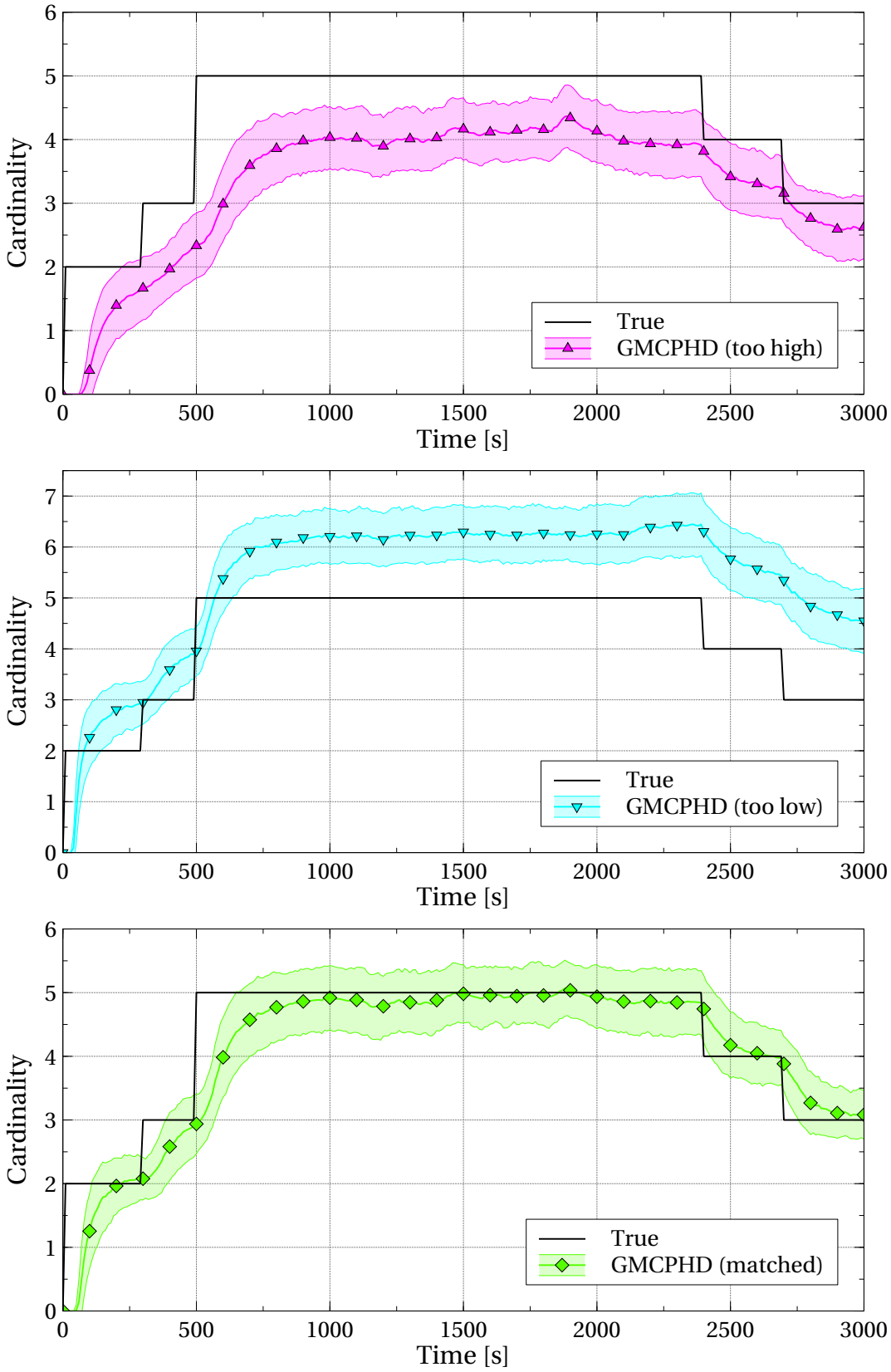


Figure 9: Constant clutter rate scenario, cardinality results for GMCPHD filters with clutter rate set too high, too low, and matched to truth.

ter component is capable of automatically responding to changes in clutter rate over time, as illustrated by the lower plot in Figure 11.

Some bias in the estimated clutter rate is evident from Figure 11, which is due to the fact that the parameters of the clutter model are tuned to give best performance for true clutter rates of around 25. Hence, when the true rate is lower, the estimate will be biased high, and when the true rate is higher, the estimate will be biased low. However, these small biases were not detrimental to the performance of the filter, and the average OSPA is almost identical to the unrealistic case in which the clutter rate is matched exactly to the truth. Clearly, a fixed clutter rate filter could not provide this level of performance, unless its parameters were being manually updated as the true clutter rate changes.

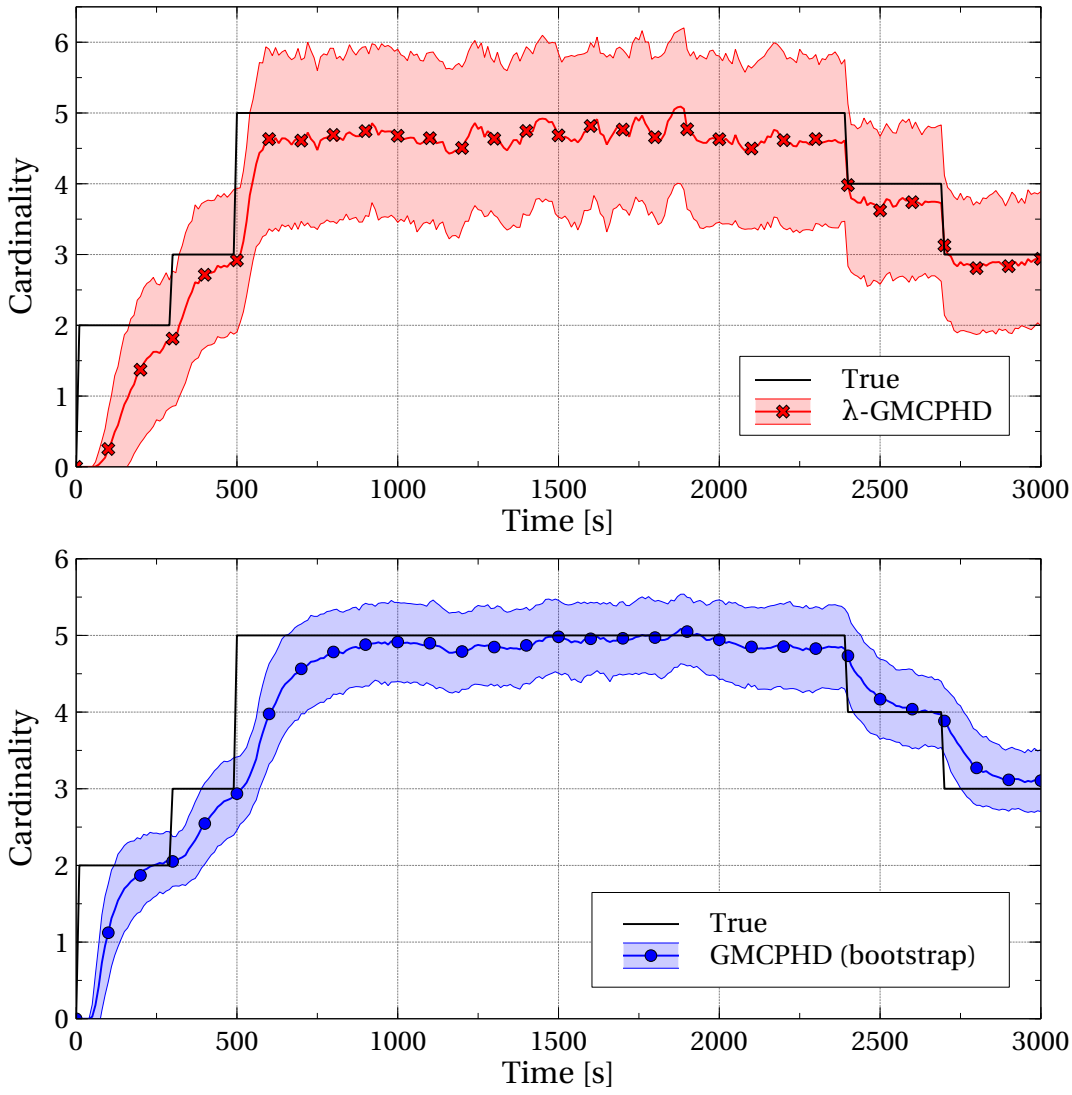


Figure 10: Constant clutter rate scenario, cardinality results for λ-GMCPHD and bootstrap GMCPHD filters.

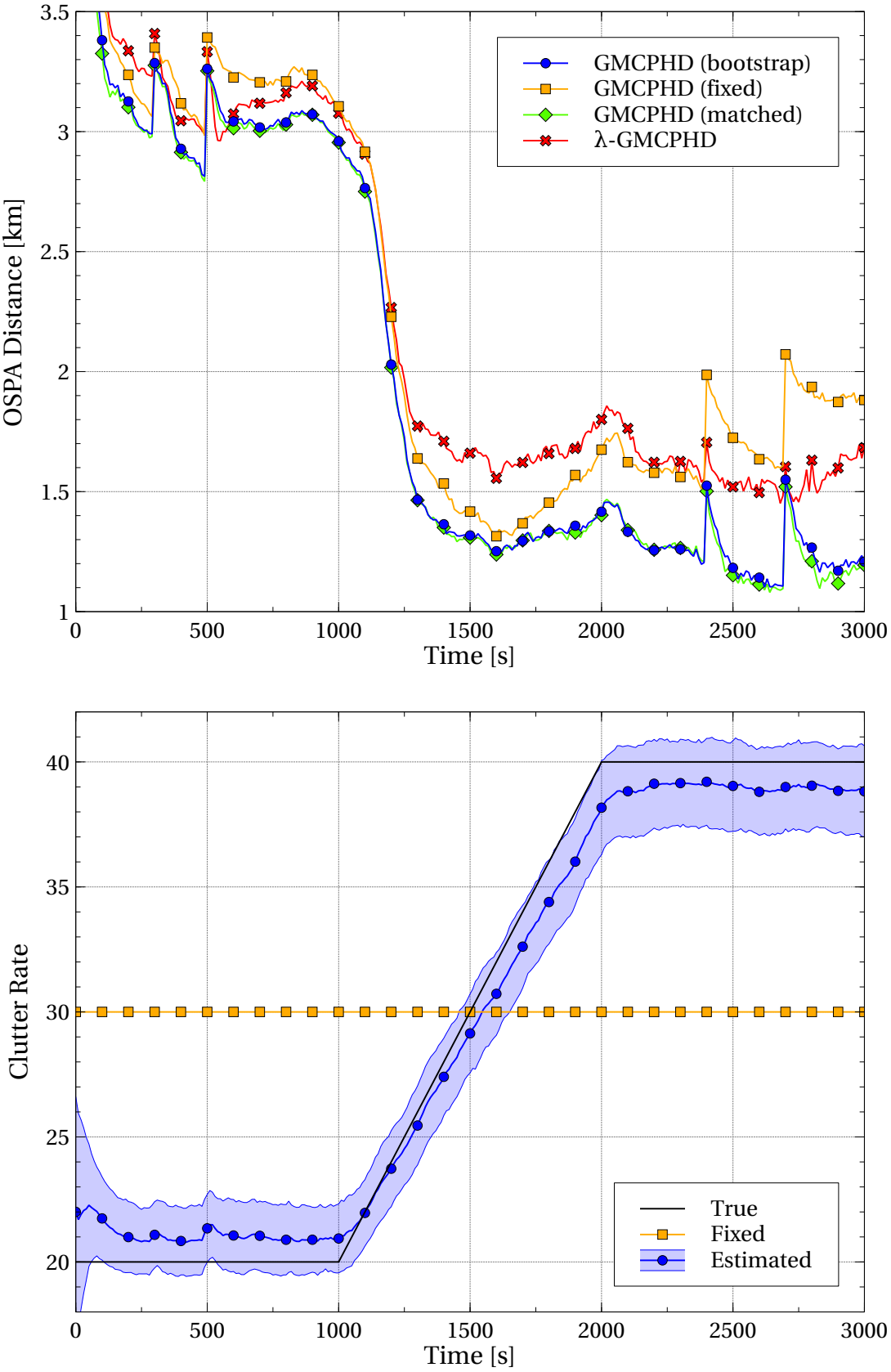


Figure 11: Variable clutter rate scenario, OSPA and clutter rate results fro GMCPHD filters. In the clutter rate plot, the "fixed" line shows the parameter setting for the corresponding GMCPHD filter, which is mid-way between the true lowest and highest clutter rates. The "estimated" line shows the estimates generated by the λ-GMCPHD filter, which is also used by the bootstrap GMCPHD filter.

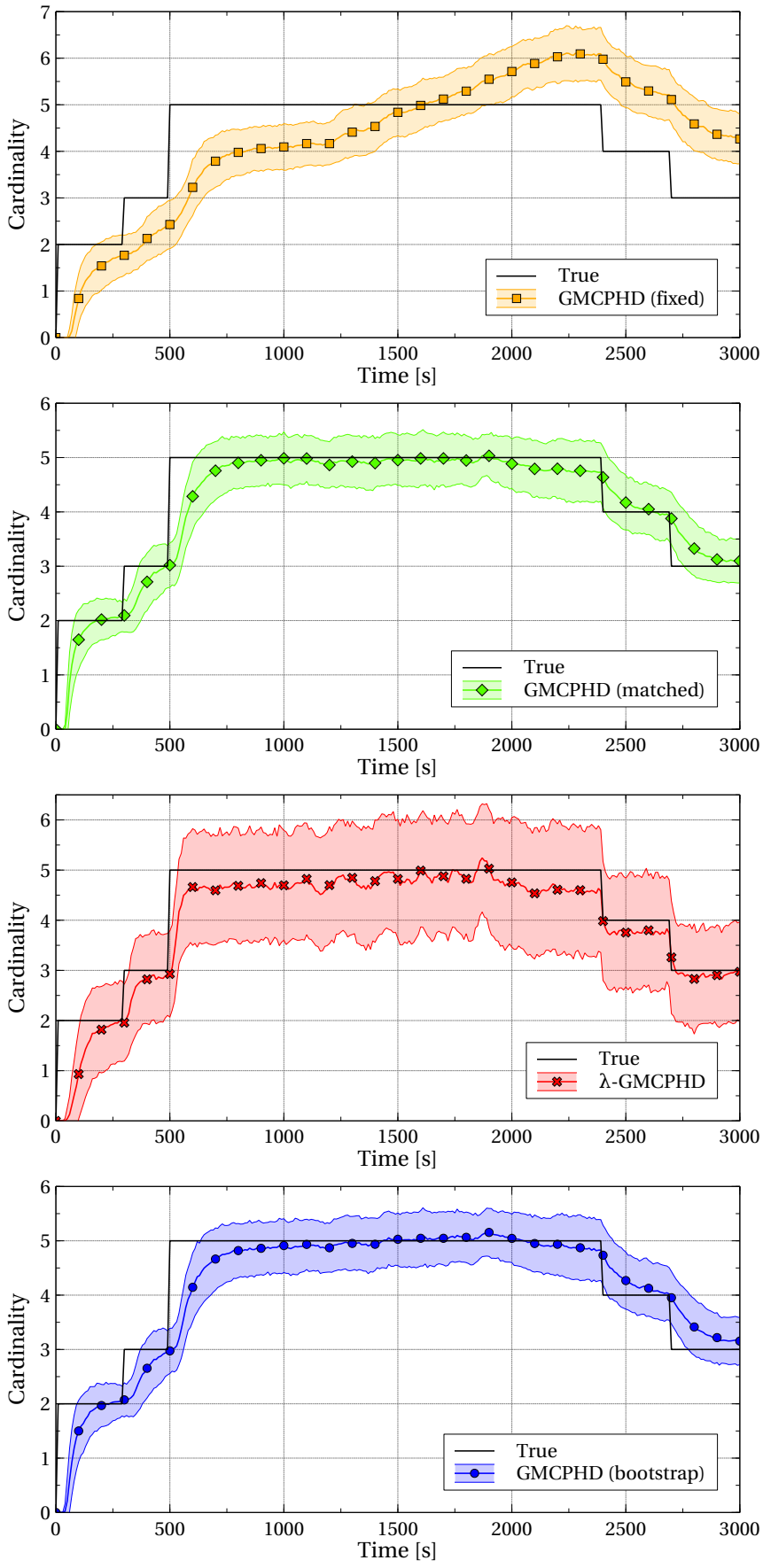


Figure 12: Variable clutter rate scenario, cardinality results for GMCPHD filters.

2.5 MULTI-TARGET TRACKING WITH LABELLED RANDOM FINITE SETS

Until recently, methods based on the RFS framework have only considered the case of unlabelled random finite sets. This has been a major point of criticism for the RFS framework, since it leads to algorithms which are inherently unable to estimate target trajectories over time. All of the algorithms mentioned in the previous section (PHD, CPHD, and CBMeMBer) have this limitation, generating sets of disconnected point estimates at each time instant, instead of continuous trajectories. This limitation was first addressed in [198], in which the idea of *labelled* random finite sets was introduced. This section provides a summary of existing multi-target tracking techniques based on this concept.

2.5.1 *Notation and Definitions*

We begin by defining some notation which will be used when working with labelled random finite sets. The multi-object exponential of a real valued function h raised to a set X is defined as

$$[h]^X \triangleq \prod_{x \in X} h(x), \tag{150}$$

where $[h]^{\emptyset} = 1$, and the elements of X may be of any type such as scalars, vectors, or sets, provided that the function h takes an argument of that type. The generalised Kronecker delta function is defined as

$$\delta_Y(X) \triangleq \begin{cases} 1, & \text{if } X = Y \\ 0, & \text{otherwise} \end{cases} \tag{151}$$

where again, X and Y may be of any type, such as scalars, vectors, or sets. The set inclusion function is given by

$$1_Y(X) \triangleq \begin{cases} 1, & \text{if } X \subseteq Y \\ 0, & \text{otherwise} \end{cases}. \tag{152}$$

A labelled set is simply a set in which every element has been augmented with a label, drawn from a discrete label space. Consider an unlabelled set $X = \{x_1, \ldots, x_n\}$. A labelled version of X is obtained by augmenting each x_i with a label ℓ_i, that is

$$\boldsymbol{X} = \{(x_1, \ell_1), \ldots (x_n, \ell_n)\}. \tag{153}$$

Each element of a labelled set is alternatively denoted by a bold lower case symbol $\boldsymbol{x}_i \equiv (x_i, \ell_i)$, and thus $\boldsymbol{X}$ can also be expressed as

$$\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\} \tag{154}$$

Note that in general, we adopt the notational convention that labelled sets are expressed in bold upper case ($\boldsymbol{X}$), unlabelled sets in regular upper case (X), labelled vectors in bold lower case ($\boldsymbol{x}$), and unlabelled vectors or scalars in regular lower case (x). For a labelled set $\boldsymbol{X}$ given by (153), the function which extracts the labels is defined as

$$\mathcal{L}(\boldsymbol{X}) = \{\ell_1, \ldots, \ell_n\}. \tag{155}$$

Note that if the $\boldsymbol{X}$ contains multiple elements with the same label, then the cardinality of $\mathcal{L}(\boldsymbol{X})$ will not be the same as the cardinality of $\boldsymbol{X}$, since by definition, a set cannot contain repeated elements. To determine whether the the set $\boldsymbol{X}$ contains repeated labels, we define the following distinct label indicator function

$$\Delta(\boldsymbol{X}) = \delta_{|\boldsymbol{X}|}(|\mathcal{L}(\boldsymbol{X})|). \tag{156}$$

A labelled RFS is a labelled set-valued random variable, which only takes on values with distinct labels. More formally, we define it as follows.

Definition 1. A *labelled RFS* with state space $\mathbb{X}$ and discrete label space $\mathbb{L}$, is an RFS on $\mathbb{X} \times \mathbb{L}$, such that the labels within each realisation are always distinct. That is, a realisation $\boldsymbol{X}$ of a labelled RFS always satisfies $\Delta(\boldsymbol{X}) = 1$.

The reason for the distinct label indicator is to ensure uniqueness of the labels within any given realisation of the set. Although by definition, a set can only contain unique elements, this criterion would still be satisfied if a labelled RFS were to contain two elements with the same label but different states. Since an object with a given label cannot be in two states simultaneously, this would not be physically meaningful.

We now define two specific classes of labelled RFS, which are useful in deriving tractable implementations of the multi-object Bayes recursion. A relatively simple class is called the labelled multi-Bernoulli [198], which as its name implies, is a multi-Bernoulli RFS in which each independent Bernoulli element has an associated label which is distinct from all others in the set.

Definition 2. A *labelled multi-Bernoulli* (LMB) RFS is a labelled RFS with state space $\mathbb{X}$ and discrete label space $\mathbb{L}$, which is distributed according to

$$\boldsymbol{\pi}(\boldsymbol{X}) = \Delta(\boldsymbol{X})\, w(\mathcal{L}(\boldsymbol{X}))\, [p(\cdot)]^{\boldsymbol{X}}, \tag{157}$$

where

$$\begin{aligned} w(L) &= \prod_{\ell \in \mathbb{L}} \left(1 - r^{(\ell)}\right) \prod_{\ell \in L} \left(1_L(\ell) \frac{r^{(\ell)}}{1 - r^{(\ell)}}\right), \\ p(x, \ell) &= p^{(\ell)}(x), \end{aligned} \tag{158}$$

in which $r^{(\ell)}$ and $p^{(\ell)}(\cdot)$ are the existence probability and probability density corresponding to label $\ell \in \mathbb{L}$. An LMB distribution is abbreviated using the notation $\boldsymbol{\pi}(\boldsymbol{X}) = \left\{\left(r^{(\ell)}, p^{(\ell)}\right)\right\}_{\ell \in \mathbb{L}}$.

A more sophisticated and flexible class of labelled RFS is called the generalised labelled multi-Bernoulli [198], which is defined as follows.

Definition 3. A *generalised labelled multi-Bernoulli* (GLMB) RFS is a labelled RFS with state space $\mathbb{X}$ and discrete label space $\mathbb{L}$, which satisfies the probability distribution

$$\pi(\boldsymbol{X}) = \Delta(\boldsymbol{X}) \sum_{c \in \mathbb{C}} w^{(c)}(\mathcal{L}(\boldsymbol{X})) \left[p^{(c)}(\cdot)\right]^{\boldsymbol{X}} \tag{159}$$

where $\mathbb{C}$ is an arbitrary index set, and

$$\sum_{L \subseteq \mathbb{L}} \sum_{c \in \mathbb{C}} w^{(c)}(L) = 1, \tag{160}$$

$$\int_{x \in \mathbb{X}} p^{(c)}(x, \ell)\, dx = 1. \tag{161}$$

The GLMB, has been shown to be closed under the Champan-Kolmogorov equation with the standard multi-object transition function, and it is also a *conjugate prior* with respect to the standard multi-object observation function [198], i.e. the form of the density is retained through both prediction and update.

The GLMB is a very general and flexible model for a labelled RFS, but its application to multi-target tracking problems is not immediately obvious. However, a more specific type of GLMB RFS has been proposed, as defined below, which presents a more conspicuous path towards the implementation of multi-target tracking algorithms [198].

Definition 4. A *δ-generalised labelled multi-Bernoulli* (δ-GLMB) RFS with state space $\mathbb{X}$ and label space $\mathbb{L}$ is a generalised labelled multi-Bernoulli RFS in which

$$\mathbb{C} = \mathcal{F}(\mathbb{L}) \times \Xi \tag{162}$$

$$w^{(c)}(L) = w^{(I,\xi)}(L) = \omega^{(I,\xi)}\delta_I(L) \tag{163}$$

$$p^{(c)} = p^{(I,\xi)} = p^{(\xi)} \tag{164}$$

where Ξ is a discrete space. Thus, a δ-GLMB RFS is distributed according to

$$\boldsymbol{\pi}(\boldsymbol{X}) = \Delta(\boldsymbol{X}) \sum_{(I,\xi)\in\mathcal{F}(\mathbb{L})\times\Xi} \omega^{(I,\xi)}\delta_I(\mathcal{L}(\boldsymbol{X}))\left[p^{(\xi)}\right]^{\boldsymbol{X}}. \tag{165}$$

In a δ-GLMB, the sum is taken over the Cartesian product between the space of finite subsets of $\mathbb{L}$, and the discrete space Ξ. In tracking applications, each subset of $\mathbb{L}$ is used to represent a set of target labels, and Ξ is used to represent the space of measurement-to-label association histories. Hence, a particular element $(I,\xi) \in \mathcal{F}(\mathbb{L}) \times \Xi$ can be considered to be the hypothesis that the set of currently existing targets are those with labels I and association history ξ. The weight $\omega^{(I,\xi)}$ is the probability of this hypothesis, and $p^{(\xi)}(\cdot,\ell)$ is the pdf of the target with label ℓ under association history ξ.

Like the GLMB, the δ-GLMB has also been shown to be closed under the standard multi-object prediction, and a conjugate prior with respect to the standard multi-object measurement likelihood [198]. This makes the δ-GLMB a highly useful tool for deriving analytical Bayes recursions for multi-object estimation and tracking problems.

2.5.2 *Standard Labelled Multi-target Dynamic Model*

Here we define the standard multi-target dynamic model with the inclusion of target labels. Let $\boldsymbol{X}$ be the labelled RFS of objects at the current time with label space $\mathbb{L}$.

A particular object $(x, \ell) \in \boldsymbol{X}$ has probability $P_S(x, \ell)$ of surviving to the next time with state (x_+, ℓ_+) and probability density $f(x_+|x, \ell)\, \delta_\ell(\ell_+)$ (where $f(x_+|x, \ell)$ is the single target transition kernel), and probability, $1 - P_S(x, \ell)$ of being terminated. Thus, the set $\boldsymbol{S}$ of surviving objects at the next time is distributed as

$$f_S(\boldsymbol{S}|\boldsymbol{X}) = \Delta(\boldsymbol{S})\, \Delta(\boldsymbol{X})\, 1_{\mathcal{L}(\boldsymbol{X})}(\mathcal{L}(\boldsymbol{S}))\, [\Phi(\boldsymbol{S}; \cdot)]^{\boldsymbol{X}} \tag{166}$$

where

$$\Phi(\boldsymbol{S}; x, \ell) = \sum_{(x_+, \ell_+) \in \boldsymbol{S}} \delta_\ell(\ell_+)\, P_S(x, \ell)\, f(x_+|x, \ell) + \left(1 - 1_{\mathcal{L}(\boldsymbol{S})}(\ell)\right)(1 - P_S(x, \ell)). \tag{167}$$

Now let $\boldsymbol{B}$ be the labeled RFS of newborn objects with label space $\mathbb{B}$, where $\mathbb{L} \cap \mathbb{B} = \emptyset$. To ensure that $\mathbb{L}$ and $\mathbb{B}$ are disjoint, we adopt a labelling scheme such that each new birth is labelled with a pair (k_b, l_b), where k_b is the current scan index and l_b is a unique index. Since the time changes from scan to scan, the space of birth labels is always disjoint from the space of surviving labels. Since the births have distinct labels, and assuming that their states are independent, we model $\boldsymbol{B}$ as a labelled multi-Bernoulli RFS with distribution

$$f_B(\boldsymbol{B}) = \Delta(\boldsymbol{B})\, w_B(\mathcal{L}(\boldsymbol{B}))\, [p_B(\cdot)]^{\boldsymbol{B}}. \tag{168}$$

where $p_B(\cdot, \ell)$ is the single target birth density corresponding to label ℓ, and $w_B(\cdot)$ is the birth weight defined by (158) in which $r^{(\ell)} = r_B^{(\ell)}$ are the birth target existence probabilities (see Section IV-D of [198] for more details). We present the prediction based on an LMB birth model, however, this can easily be extended to the case of a GLMB birth model. The overall multi-object state at the next time step is the union of survivals and new births, i.e. $\boldsymbol{X}_+ = \boldsymbol{S} \cup \boldsymbol{B}$. The label spaces $\mathbb{L}$ and $\mathbb{B}$ are disjoint, and the states of newborn objects are independent of surviving objects, hence $\boldsymbol{S}$ and $\boldsymbol{B}$ are independent. It was shown in [198] that the multi-object transition can be expressed as

$$f(\boldsymbol{X}_+|\boldsymbol{X}) = f_S(\boldsymbol{X}_+ \cap (\mathbb{X} \times \mathbb{L})\,|\boldsymbol{X})\, f_B(\boldsymbol{X}_+ - (\mathbb{X} \times \mathbb{L})). \tag{169}$$

Although the tracking algorithms proposed in this dissertation do not require sampling from the multi-object transition density, this will be necessary in Chapter 5 where we propose a multi-target sensor control scheme. Algorithm 1 demonstrates

how to draw a sample from the standard multi-target survival density, given a current set of target states and labels. A sample from the full transition can be obtained by taking the union of this with a sample drawn from the multi-target birth density.

Algorithm 1: Sample from the standard multi-target survival density.

Input: States at time k (X_k), labels at time k (L_k), survival probability (P_S)
Output: Sampled states at time $k+1$ (X_{k+1}), sampled labels at time $k+1$ (L_{k+1})

Function $[X_{k+1}, L_{k+1}] = \texttt{SampleSurvivals}\left(X_k, L_k, P_S\right)$ **is**
  $j = 0$;
  for $i = 1 : |X_k|$ **do**
    $r \sim \texttt{Uniform}\left([0,1]\right)$;
    if $r < P_S\left(X_k^{(i)}\right)$ **then**
      $j = j + 1$;
      $X_{k+1}^{(j)} \sim f\left(\cdot | X_k^{(i)}\right)$;
      $L_{k+1}^{(j)} = L_k^{(i)}$;

2.5.3 *Standard Multi-target Observation Model*

Let $\boldsymbol{X}$ be the labelled RFS of objects that exist at the observation time. A particular object $\boldsymbol{x} \in \boldsymbol{X}$ has probability $P_D\left(\boldsymbol{x}\right)$ of generating a detection z with likelihood $g\left(z|\boldsymbol{x}\right)$, and probability $1 - P_D\left(\boldsymbol{x}\right)$ of being misdetected. Let W be the set of target detections. Assuming the elements of W are conditionally independent, W is a multi-Bernoulli RFS with parameter set $\left\{\left(P_D\left(\boldsymbol{x}\right), g\left(\cdot|\boldsymbol{x}\right)\right) : \boldsymbol{x} \in \boldsymbol{X}\right\}$, and we write its probability density as

$$\pi_D\left(W|\boldsymbol{X}\right) = \left\{\left(P_D\left(\boldsymbol{x}\right), g\left(\cdot|\boldsymbol{x}\right)\right) : \boldsymbol{x} \in \boldsymbol{X}\right\}\left(W\right). \tag{170}$$

Let K be the set of clutter observations, which are independent of the target detections. We model K as a Poisson RFS with intensity $\kappa(\cdot)$, hence K is distributed according to

$$\pi_K\left(K\right) = e^{-\langle \kappa, 1\rangle} \kappa^K. \tag{171}$$

The overall multi-object observation is the union of target detections and clutter observations, i.e. $Z = W \cup K$. Since W and K are independent, the multi-object likelihood is

$$g\left(Z|\boldsymbol{X}\right) = \sum_{D \subseteq Z} \pi_D\left(W|\boldsymbol{X}\right) \pi_K\left(Z - W\right). \tag{172}$$

As demonstrated in [135], this can be equivalently expressed as

$$g\left(Z|\boldsymbol{X}\right) = e^{-\langle \kappa,1\rangle}\kappa^{K}\sum_{\theta\in\Theta}\left[\psi_{Z}\left(\cdot;\theta\right)\right]^{\boldsymbol{X}} \tag{173}$$

where Θ is the set of all mappings of labels to measurement indices, $\theta : \mathbb{L} \rightarrow \{0,1,...,|Z|\}$, such that $[\theta(i) = \theta(j) > 0] \Rightarrow [i = j]$ (i.e. a given θ can map multiple labels to 0, but only one label can be mapped to each non-zero measurement index). Finally, the function $\psi_{Z}\left(\cdot;\theta\right)$ in (173) is given by

$$\psi_{Z}\left(x,\ell;\theta\right) = \begin{cases} \frac{P_{D}(x,\ell)g\left(z_{\theta(\ell)}|x,\ell\right)}{\kappa\left(z_{\theta(\ell)}\right)}, & \theta\left(\ell\right) > 0 \\ 1 - P_{D}\left(x,\ell\right), & \theta\left(\ell\right) = 0 \end{cases}. \tag{174}$$

The standard observation model as defined in (173) and (174) is based on the fundamental assumption that each target generates at most one measurement, and that each measurement originates from at most one target. Although this is useful in the derivation of computationally efficient filtering algorithms, these assumptions are too restrictive to model real-world phenomena such as measurement merging or extended objects. Chapters 3 and 4 of this dissertation present methods of dealing with these cases, providing a higher-fidelity modelling of the observations using the RFS formulation.

Algorithm 2 shows how to draw a sample from the standard multi-target observation model, given a current set of target states.

2.5.4 *Generalised Labelled Multi-Bernoulli Filter*

The GLMB filter [198] models the multi-object state as a GLMB random finite set, as specified by Definition 4. The parameters of the GLMB density are propagated forward in time via a prediction, and new observation data is incorporated via a measurement update. These two procedures are summarised in the following subsections, and pseudo-code detailing their implementation can be found in Section 2.6.1.

Algorithm 2: Sample from the standard multi-target measurement model.

Input: Target states (X), sensor state (u), detection probability (P_D), clutter intensity (λ), clutter region (S)
Output: Measurement sample (Z)

Function $Z = \texttt{SampleMeasurements}(X, u, P_D, \lambda, S)$ **is**
  $Z = \emptyset$;
  for $i = 1 : |X|$ **do**
    $r \sim \texttt{Uniform}([0,1])$;
    if $r < P_D\left(X^{(i)}\right)$ **then**
      $z \sim g\left(\cdot|X^{(i)}, u\right)$;
      $Z = Z \cup \{z\}$;
  $n \sim \texttt{Poisson}(\cdot;\lambda)$;
  for $i = 1 : n$ **do**
    $z \sim \texttt{Uniform}(S)$;
    $Z = Z \cup \{z\}$;

2.5.4.1 *Prediction*

If the posterior multi-object density at the current time is a GLMB given by (165), then under the standard multi-object dynamic model defined in Section 2.5.2, the predicted multi-object density at the next time step is also a GLMB given by [198]

$$\boldsymbol{\pi}_+(\boldsymbol{X}) = \Delta(\boldsymbol{X}) \sum_{c\in\mathbb{C}} w_+^{(c)}(\mathcal{L}(\boldsymbol{X})) \left[p_+^{(c)}(\cdot)\right]^{\boldsymbol{X}}, \tag{175}$$

where

$$w_+^{(c)}(L) = w_B(L\cap\mathbb{B})\, w_S^{(c)}(L\cap\mathbb{L}), \tag{176}$$

$$p_+^{(c)}(x,\ell) = 1_{\mathbb{L}}(\ell)\, p_S^{(c)}(x,\ell) + 1_{\mathbb{B}}(\ell)\, p_B(x,\ell), \tag{177}$$

$$p_S^{(c)}(x,\ell) = \frac{\left\langle P_S(x,\ell) f(x|\cdot,\ell), p^{(c)}(\cdot,\ell)\right\rangle}{\eta_S^{(c)}(\ell)}, \tag{178}$$

$$\eta_S^{(c)}(\ell) = \int \left\langle P_S(x,\ell) f(x|\cdot,\ell), p^{(c)}(\cdot,\ell)\right\rangle dx, \tag{179}$$

$$w_S^{(c)}(J) = \left[\eta_S^{(c)}\right]^J \sum_{I\subseteq\mathbb{L}} 1_I(J) \left[q_S^{(c)}\right]^{I-J} w^{(c)}(I), \tag{180}$$

$$q_S^{(c)}(\ell) = \left\langle 1 - P_S(\cdot,\ell), p^{(c)}(\cdot,\ell)\right\rangle. \tag{181}$$

In principle, this involves generating a predicted GLMB component for all combinations of target birth, death and survival, for each component in the current density. In practice however, doing so is usually infeasible because the number of predicted

components becomes too large. To improve the efficiency of the prediction, approximations can be used to avoid generating unlikely components. In practice, the prediction for target survivals and target births are carried out separately, and the resulting GLMBs are combined to arrive at the overall prediction.

2.5.4.2 *Survival Prediction*

One approach to computing the predicted GLMB of surviving targets is to use a k-shortest paths algorithm, which identifies the most likely combinations of surviving targets in decreasing order of weight. To achieve this, we first generate a graph structure with one row per target and two columns as shown in Figure 13, in which the nodes in the first column contain the costs of target survival (from equation (179)), and the nodes in the second column contain the costs of target death (from equation (181)). A k-shortest paths algorithm is used to generate the cheapest paths from the start node to the end node. Each path is then used to compute a new predicted component, where the surviving targets are those that correspond to rows in which the the first column was visited.

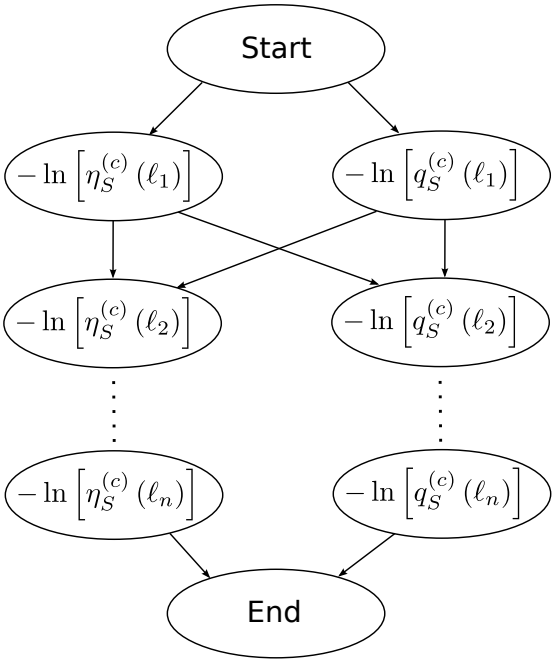


Figure 13: Graph structure used in the GLMB survival prediction.

Another method to reduce the computation is to impose restrictions on the target death process. For example, if it reasonable to assume that on each scan, a maximum of one existing target can die, then the number of predicted components can be greatly reduced, resulting in a significant reduction in the computational load.

2.5.4.3 *Measurement Driven Birth*

In many multi-target tracking scenarios, it is possible for new targets to appear anywhere within the sensor's detection region. In such cases, the target birth model must not restrict the locations where new tracks can be initiated, or else the tracker might fail to find the new targets. The GLMB filter assumes that the birth density is a GLMB, and the simplest way to implement this is to construct a "static" birth density off-line. The problem with this method is that it restricts the states of new targets to be in the areas covered by the static birth model. To ensure that all possible birth states are covered by the model may be computationally prohibitive, since the number of required components could be very large.

A solution to this is to construct the birth model on-line using the measurement data. In the most general case, a measurement driven birth model would allow any number of measurements on any scan to initiate new tentative tracks. Since each term in the GLMB needs to capture a complete set of hypothesised births, this approach may also be computationally prohibitive, since it requires enumerating all possible subsets of the measurements on each scan. Nevertheless, the computational load can be reduced by making a few reasonable assumptions as follows;

1. a maximum of one new target can appear on each scan,
2. measurements that are likely to have originated from existing targets do not give rise to new birth terms, and
3. new targets are always detected on the first scan after they appear.

Applying these assumptions reduces the number of terms in the birth density, which in turn significantly reduces the number of components in the predicted GLMB.

Let us denote the measurement set from the previous scan as Z_{k-1}. A measurement driven birth GLMB is constructed according to

$$\pi_{B,k}(\boldsymbol{X}) = \Delta(\boldsymbol{X}) \sum_{i \in |Z_{k-1}|} w_B(\mathcal{L}(\boldsymbol{X})) \left[p_B(\cdot|Z_{k-1}, i)\right]^{\boldsymbol{X}} \tag{182}$$

where

$$w_B(L) = \delta_1(|L|)\, 1_{\mathbb{B}_{k-1}}(L)\, P_B(L), \tag{183}$$

$$p_B(x, (k-1, j)\,|Z_{k-1}, i) = \left\langle f(x|\cdot, (k-1, j)), p\left(\cdot|z_{k-1}^{(i)}\right)\right\rangle \delta_i(j) \tag{184}$$

in which $\mathbb{B}_{k-1} = \{(k-1,i)\,|i \in 1 \ldots |Z_{k-1}|\}$ is the space of birth labels at time $k-1$, $P_B(L)$ (for $L = \{(k-1,i)\}$) is the prior probability that the i-th measurement in scan $k-1$ was generated by a new target, and $p\left(\cdot|z_{k-1}^{(i)}\right)$ is the prior pdf of that target. Note that the delta function in (183) restricts the birth density to have non-zero value only when $\boldsymbol{X}$ is a singleton, i.e. only one birth target is permitted at a time. However, different choices for the label and pdf are possible, depending on which measurement is associated with the new target. In practice it is also possible to further reduce the number of terms by making use of consecutive measurement scans to construct the birth density. If we assume that a birth target must be detected on the first two scans after its appearance, it is often possible to exclude many of the terms in (182), since a term is included only when a measurement in the next scan validates the possible presence of a target.

2.5.4.4 *Measurement Update*

Suppose the prior multi-object density at the current time is a GLMB given by (165), and we wish to compute the posterior multi-object density conditioned on a set of received measurements Z. Let Θ be the set of mappings between target labels and measurement indices, as specified in Section 2.5.3. Under the standard multi-object observation likelihood function (173), the posterior multi-object density is also a GLMB given by [198]

$$\boldsymbol{\pi}(\boldsymbol{X}|Z) = \Delta(\boldsymbol{X}) \sum_{c\in\mathbb{C}} \sum_{\theta\in\Theta(\mathcal{L}(\boldsymbol{X}))} w_Z^{(c,\theta)}(\mathcal{L}(\boldsymbol{X})) \left[p^{(c,\theta)}(\cdot|Z)\right]^{\boldsymbol{X}}, \tag{185}$$

where

$$w_Z^{(c,\theta)}(L) = \frac{w^{(c)}(L)\left[\eta_Z^{(c,\theta)}\right]^I}{\sum_{J\subseteq\mathbb{L}}\sum_{c\in\mathbb{C}}\sum_{\theta\in\Theta(J)} w^{(c)}(J)\left[\eta_Z^{(c,\theta)}\right]^J}, \tag{186}$$

$$p^{(c,\theta)}(x,\ell|Z) = \frac{p^{(c)}(x,\ell)\,\psi_Z(x,\ell;\theta)}{\eta_Z^{(c,\theta)}(\ell)}, \tag{187}$$

$$\eta_Z^{(c,\theta)}(\ell) = \left\langle p^{(c)}(\cdot,\ell), \psi_Z(\cdot,\ell;\theta)\right\rangle, \tag{188}$$

$$\psi_Z(x,\ell;\theta) = \begin{cases} \frac{P_D(x,\ell)g\left(z_{\theta(\ell)}|x,\ell\right)}{\kappa\left(z_{\theta(\ell)}\right)}, & \theta(\ell) > 0 \\ 1 - P_D(x,\ell), & \theta(\ell) = 0 \end{cases}. \tag{189}$$

In principle, the update involves generating an updated GLMB component for all possible associations of measurements to targets. Usually this is not feasible in practice, since the number of associations increases combinatorially as the number of targets and measurements increases. Several techniques can be employed to reduce the number of posterior GLMB components that need to be stored.

The first of these is measurement gating [29], which involves disallowing unlikely measurement-to-target mappings based on a predefined probability threshold. This is a well established method, and it is used extensively in other multi-target tracking algorithms to reduce the computational load. Removing the unlikely associations up-front results in a drastic reduction in the number of feasible measurement-to-target mappings.

The next is the use of a ranked assignment algorithm to generate a requested number of components with highest weights. For a set of targets with labels $\{\ell_1, \ldots, \ell_n\}$, and a set of measurements Z (with $|Z| = m$), this is done by constructing the following cost matrix,

$$-\ln \begin{pmatrix} \phi^{(c)}(\ell_1, z_1) & \cdots & \phi^{(c)}(\ell_1, z_m) & \phi^{(c)}(\ell_1, \varnothing) & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \phi^{(c)}(\ell_n, z_n) & \cdots & \phi^{(c)}(\ell_n, z_m) & 0 & \cdots & \phi^{(c)}(\ell_n, \varnothing) \end{pmatrix}. \tag{190}$$

where

$$\phi^{(c)}(\ell, z) = \left\langle p^{(c)}(\cdot, \ell), \frac{P_D(\cdot, \ell)\, g(z|\cdot, \ell)}{\kappa(z)} \right\rangle, \tag{191}$$

$$\phi^{(c)}(\ell, \varnothing) = \left\langle p^{(c)}(\cdot, \ell), 1 - P_D(\cdot, \ell) \right\rangle. \tag{192}$$

In (190), each row represents a target, the first m columns represent detections, and the last n columns represent misdetections. Note that if measurement gating is carried out beforehand, many of the detection terms in the first m columns will be zero, indicating that the corresponding measurement-to-target association is infeasible. The matrix (190) is processed using a ranked assignment algorithm (for example Murty's algorithm [147]), which yields a sorted list of the cheapest one-to-one assignments of rows to columns, in increasing order of cost. Note that using this formulation, each solution must assign every row to a column, but there may be columns that are not assigned to a row. In generating the list of ranked assignments, whenever a row is

assigned to a column index greater than m by Murty's algorithm, the assignment is reported as 0, to indicate that the target was misdetected. Each element in the list is then used to construct a component in the posterior GLMB density.

The final method for reducing the number of posterior components is to carry out a pruning procedure after the update, which removes components with low weights from the GLMB representation. Several different criteria can be used to remove components, for example, deleting those with weights falling below a fixed threshold, or those with weights lower than some proportion of the highest weight. Furthermore, an upper bound on the total number of components can be enforced, for example, by retaining only the N highest weighted components in the posterior density.

Section 2.6.1 in this chapter contains detailed pseudo-code for an implementation of the GLMB filter, based on the standard multi-object dynamic and measurement models.

2.5.5 *Labelled Multi-Bernoulli Filter*

The aim of the LMB filter is to reduce the computational complexity of the GLMB filter by applying an approximation that avoids the exponential growth in the number of components in the posterior multi-object density.

2.5.5.1 *Prediction*

Instead of being represented as a GLMB, the posterior at time $k-1$ is represented as an LMB random finite set with a density of the form (157) and parameter set $\boldsymbol{\pi}_{k-1} = \left\{\left(r_{k-1}^{(\ell)}, p_{k-1}^{(\ell)}\right)\right\}_{\ell\in\mathbb{L}}$. The target birth model is also an LMB with parameter set $\boldsymbol{\pi}_{B,k} = \left\{\left(r_{B,k}^{(\ell)}, p_{B,k}^{(\ell)}\right)\right\}_{\ell\in\mathbb{B}}$, where $\mathbb{B}$ is the space of labels for new targets appearing at time k. In this case, the predicted LMB density is given by

$$\boldsymbol{\pi}_{k|k-1} = \left\{\left(r_{k|k-1}^{(\ell)}, p_{k|k-1}^{(\ell)}\right)\right\}_{\ell\in\mathbb{L}} \cup \left\{\left(r_{B,k}^{(\ell)}, p_{k|k-1}^{(\ell)}\right)\right\}_{\ell\in\mathbb{B}}, \tag{193}$$

where

$$r_{k|k-1}^{(\ell)} = r_{k-1}^{(\ell)} \eta_S\left(\ell\right), \tag{194}$$

$$p_{k|k-1}^{(\ell)}\left(x\right) = \frac{\left\langle P_S\left(\cdot,\ell\right) f\left(x|\cdot,\ell\right), p_k\left(\cdot,\ell\right)\right\rangle}{\eta_S\left(\ell\right)}, \tag{195}$$

$$\eta_S\left(\ell\right) = \left\langle P_S\left(\cdot,\ell\right), p_k\left(\cdot,\ell\right)\right\rangle. \tag{196}$$

That is, to obtain the predicted LMB, we simply take the union of the set of predicted surviving tracks and the set of birth tracks. This is much cheaper to compute than the GLMB prediction, since it does not involve the sum over subsets of $\mathbb{L}$ which appears in (180). The reader is referred to [164] for more details.

2.5.5.2 *Measurement Update*

Firstly, the predicted LMB density is divided into disjoint clusters, each of which is approximately independent of all others. This is done based on the current measurement set, by observing that a pair of tracks can be considered as members of different independent clusters if no single measurement falls inside the gating region of both tracks simultaneously. More details on how this is performed can be found in [164].

The predicted LMB for each cluster is then converted into the GLMB representation. If the predicted LMB for a cluster is given by $\boldsymbol{\pi}_{k|k-1} = \left\{\left(r_{k|k-1}^{(\ell)}, p_{k|k-1}^{(\ell)}\right)\right\}_{\ell\in\mathbb{L}_+}$, this can be converted to a single component GLMB, given by (157). In principle, this involves calculating the GLMB weight for all subsets of $\mathbb{L}_+$, however, in practice, approximations can be used to reduce the number of components and improve the efficiency of the conversion. These include methods such as target grouping, truncation with k-shortest paths, or sampling procedures. For more details see [164], which expresses the converted density in δ-GLMB form, essentially enumerating all possible subsets of the tracks appearing in the predicted LMB. After computing the GLMB density of each cluster, each one is updated according to the standard GLMB update procedure (185)-(188).

After the updates, the posterior GLMB for each target cluster can be approximated by an LMB with matching probability hypothesis density, with parameters

$$r^{(\ell)} = \sum_{c\in\mathbb{C}} \sum_{L\subseteq\mathbb{L}_+} \sum_{\theta\in\Theta} w^{(c,\theta)}(L)\, 1_I(\ell)\,, \tag{197}$$

$$p^{(\ell)}(x) = \frac{1}{r^{(\ell)}} \sum_{c\in\mathbb{C}} \sum_{L\subseteq\mathbb{L}_+} \sum_{\theta\in\Theta} w^{(c,\theta)}(L)\, 1_I(\ell)\, p^{(\theta)}(x,\ell)\,. \tag{198}$$

The existence probability corresponding to each label is the sum of the weights of the GLMB components that include that label, and its pdf becomes the weighted sum of the corresponding pdfs from the GLMB. Thus, the pdf of each track in the LMB becomes a mixture of single-object densities. To avoid the number of components growing too large, it is necessary to reduce these mixtures by a process of pruning

and merging. Finally, the overall LMB approximation to the posterior at scan k is given by the union of the approximate posterior LMBs across all clusters.

Note that the computational saving of the LMB filter depends on the assumption that the number of targets (and hence the number of GLMB components) in each group will be relatively small. In this case, the total number of GLMB components that need to be processed across all target groups will be significantly lower compared to the full GLMB filter.

While the LMB approximation matches the PHD (and therefore the mean cardinality) of the full GLMB, it cannot match its complete cardinality distribution. The reason for this is that the LMB model has fewer degrees of freedom than the GLMB, and thus the form of its cardinality distribution is more restricted. Since the LMB filter makes this approximation after every update, it incurs a cumulative error in the cardinality distribution, which can lead to increased estimation error compared to the GLMB filter.

Detailed pseudo-code for the LMB filter recursion for the standard multi-object dynamic and measurement models is given in Section 2.6.2 at the end of this chapter.

2.6 IMPLEMENTATION OF LABELLED RFS FILTERS

This section contains detailed pseudo-code for both the GLMB and LMB filter, under the standard multi-target dynamic and measurement models. The code is divided into functions, many of which are used in later parts of this dissertation.

2.6.1 *Pseudo-code for the Standard GLMB Filter*

We begin by defining a data structure to represent a GLMB density, and the notation used within the pseudo-code to refer to the components of the GLMB. In the code, the symbols Φ, Ψ and Ω denote GLMB densities, and the parameters of a GLMB density Φ are denoted by the items in Table 2. Figure 14 contains a graphical representation of this data structure, which shows how a GLMB density can be represented on a computer by storing the single-target densities, target labels, component weights, and cardinality distribution. Note that in practice, a more efficient representation of a GLMB can be obtained by storing only the unique single-target densities in a "track table", and representing the components as references to the entries in this table. This

avoids a great deal of redundancy, as a single-target density with a given association history will usually appear in many GLMB components. However, for the sake of simplicity, the pseudo-code is presented as though each component directly stores the densities of its constituent targets. Note that for the superscripted parameters in Table 2, when an index is replaced with a colon, for example $\Phi.\rho^{(:)}$ or $\Phi.p^{(n,i,:)}$, this refers to the collection of items taken over all indices (similar to Matlab syntax). The symbol k is used to denote a discrete time index and t represents continuous time, thus t_k is used to denote the continuous time corresponding to time index k.

Table 2: GLMB data structure components.

Symbol	Description
$\Phi.\rho^{(n)}$	Cardinality distribution (probability of cardinality n)
$\Phi.w^{(n,i)}$	Weight of i-th component of cardinality n
$\Phi.p^{(n,i,j)}$	Probability density of the j-th target in the i-th component of cardinality n
$\Phi.\ell^{(n,i,j)}$	Label of the j-th target in the i-th component of cardinality n
$\Phi.C_{max}$	Maximum cardinality represented
$\Phi.N_n$	Number of components of cardinality n
$\Phi.t$	Continuous time at which this density is valid

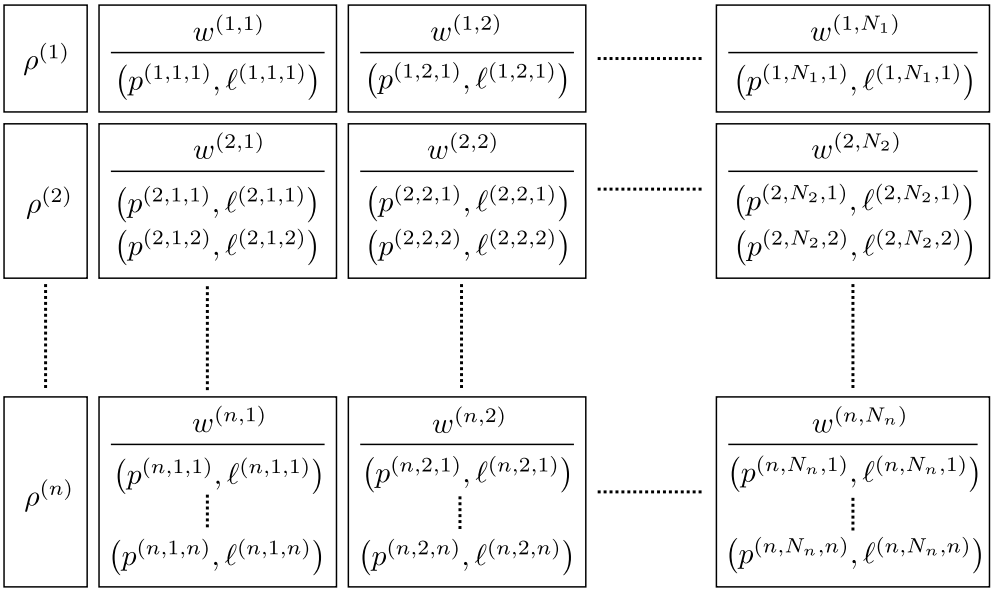


Figure 14: Graphical representation of the GLMB data structure.

Pseudo-code for the main recursion of the GLMB filter is presented in Algorithm 3. This algorithm takes a collection of measurement sets and the associated time stamps, and returns a list of labelled target trajectory estimates. The main loop consists of calculating the birth and survival densities, multiplying them together to obtain the predicted density, performing the measurement update, extracting the target estimates from the posterior, and finally, pruning the posterior density. Pseudo-code for all of these sub-functions will be defined in the sequel. For simplicity of expression, the argument list for each function contains only the input data and any items generated internally by the GLMB recursion. Any user-defined parameters required by the filter are not shown explicitly as function arguments, and are instead listed at the top of each block of pseudo-code.

Algorithm 3: GLMB filter recursion.

Input: Measurements $(Z_{1:N})$, measurement times $(t_{1:N})$
Output: Track estimates (τ_x), track labels (τ_ℓ)

Function $[\tau_x, \tau_\ell] = \texttt{GLMB\_Filter}\,(Z_{1:N}, t_{1:N})$ **is**
  $\ell_{next} = 1$;
  $\tau_x = \emptyset$;
  $\tau_\ell = \emptyset$;
  $P_{A,1} = \texttt{Zeros}\,(1, |Z_1|)$;
  for $k = 2 : N$ **do**
    $[\Omega_{B,k}, \ell_{next}] = \texttt{BirthGLMB}\,(Z_{k-1}, P_{A,k-1}, t_{k-1}, Z_k, t_k, \ell_{next})$;
    if $k = 2$ **then**
      $\Omega_+ = \Omega_{B,k}$;
    else
      $\Omega_{S,k} = \texttt{PredictSurvivalGLMB}\,(\Omega_{k-1}, t_k)$;
      $\Omega_+ = \texttt{MultiplyGLMB}\,(\Omega_{S,k}, \Omega_{B,k})$;
    $[\Omega_k, P_{A,k}] = \texttt{UpdateGLMB}\,(\Omega_+, Z_k)$;
    $[\tau_x, \tau_\ell] = \texttt{ExtractEstimatesGLMB}\,(\Omega_k, \tau_x, \tau_\ell)$;
    $\Omega_k = \texttt{PruneGLMB}\,(\Omega_k)$;

The six main functions used in the GLMB recursion of Algorithm 3 are defined in Algorithms 4-9 as follows. The additional functions used therein are defined later in Table 3.

- `BirthGLMB` (Algorithm 4): Compute a birth GLMB density using the measurements from the current and previous scans, and the probabilities that each previous measurement was generated by a target (these probabilities are computed as part of the measurement update function in Algorithm 7).

- `PredictSurvivalGLMB` (Algorithm 5): Takes the GLMB density at the current time, and computes the predicted GLMB density of surviving targets at the next time step.

- `MultiplyGLMB` (Algorithm 6): Takes two GLMB densities, and computes the density corresponding to the union between the underlying random finite sets. This is used to compute the overall predicted GLMB density, given the densities of surviving and birth targets.

- `UpdateGLMB` (Algorithm 7): Takes a predicted GLMB, and updates it using a new measurement set to return the posterior GLMB density.

- `ExtractEstimatesGLMB` (Algorithm 8): Extracts a labelled set of estimates from a GLMB density, which is used to update a list of trajectory estimates. If the label of a given estimate is already in the list, that estimate is appended to the corresponding trajectory. Estimates whose labels are not already in the list are used to initiate new estimated trajectories.

- `PruneGLMB` (Algorithm 9): Takes a GLMB and approximates it as another GLMB with fewer components. This is done by removing components with low weights according to a threshold. If the number remaining still exceeds a predefined maximum, further components are discarded.

Algorithm 4: GLMB birth density generation.

Input: Previous measurement set (Z_{k-1}), probabilities that each previous measurement is from a target ($P_{A,k-1}$), previous measurement time (t_{k-1}), current measurement set (Z_k), current time (t_k), next target label (ℓ_{next})

Parameters: Maximum association probability to generate birth (P_{thresh}), probability that a previously unassigned measurement is from a new target (P_{birth})

Output: GLMB birth density (Ω), next label (ℓ_{next})

Function $[\Omega, \ell_{next}] = \texttt{BirthGLMB}\left(Z_{k-1}, P_{A,k-1}, t_{k-1}, Z_k, t_k, \ell_{next}\right)$ **is**
- $c = 1$;
- **for** $i = 1 : |Z_{k-1}|$ **do**
  - **if** $P_{A,k-1}^{(i)} < P_{thresh}$ **then**
    - $p = \texttt{TargetPrior}\left(Z_{k-1}^{(i)}\right)$;
    - $p_+ = \texttt{TargetPredict}\left(p, t_k - t_{k-1}\right)$;
    - $V = \texttt{ValidationGate}\left(p_+\right)$;
    - `keep` $=$ `false`;
    - **for** $j = 1 : |Z_k|$ **do**
      - **if** $Z_k^{(j)} \in V$ **then**
        - `keep` $=$ `true`;
    - **if** `keep` **then**
      - $\Omega.p^{(1,c,1)} = p_+$;
      - $\Omega.\ell^{(1,c,1)} = \ell_{next}$;
      - $\Omega.w^{(1,c)} = P_{birth}$;
      - $c = c + 1$;
      - $\ell_{next} = \ell_{next} + 1$;
- $\Omega = \texttt{NormaliseGLMB}\left(\Omega\right)$;
- $\Omega.t = t_k$;

Algorithm 5: GLMB survival prediction.

Input: Previous GLMB density (Φ), prediction time (t)
Parameters: Target survival probability (P_S), number of predicted components to generate (N_p)
Output: Predicted GLMB density (Ω)

Function $\Omega = \texttt{PredictSurvivalGLMB}(\Phi, t)$ **is**
  $N_{cdn} = \texttt{Allocate}\left(N_p, \texttt{Poisson}\left(1 : \Phi.M, \texttt{Mean}\left(\Phi.\rho^{(:)}\right)\right)\right)$;
  $k = \texttt{Ones}(1, \Phi.M)$;
  for $n = 1 : \Phi.C_{max}$ **do**
    $N_{cmp} = \texttt{Allocate}\left(N_{cdn}(n), \Phi.w^{(n,:)}\right)$;
    for $i = 1 : \Phi.N_n$ **do**
      $G = \texttt{SurvivalGraph}\left(\Phi.p^{(n,i,:)}, P_s\right)$;
      $P = \texttt{ShortestPaths}\left(G, N_{cmp}(i)\right)$;
      for $(p, c) \in P$ **do**
        $s = \{j; p(j) = 1\}$;
        $m = |s|$;
        for $j = 1 : m$ **do**
          $\Omega.p^{(m,k(m),j)} = \texttt{TargetPredict}\left(\Phi.p^{(n,i,s(j))}, t - \Phi.t\right)$;
          $\Omega.\ell^{(m,k(m),j)} = \Phi.\ell^{(n,i,s(j))}$;
        $\Omega.w^{(m,k(m))} = \Phi.w^{(n,i)} \times \Phi.\rho^{(n)} \times e^{-c}$;
        $k(m) = k(m) + 1$;
  $\Omega = \texttt{NormaliseGLMB}(\Omega)$;
  $\Phi.t = t$;

Algorithm 6: Multiply two GLMB densities.

Input: First GLMB density (Φ), second GLMB density (Ψ)
Output: Multiplied GLMB density (Ω)

Function $\Omega = \texttt{MultiplyGLMB}(\Phi, \Psi)$ **is**
  $c = \texttt{Ones}(1, \Phi.M + \Psi.M)$;
  for $m = 1 : \Phi.C_{max}$ **do**
    for $n = 1 : \Psi.M$ **do**
      $k = m + n$;
      for $i = 1 : \Phi.N_m$ **do**
        for $j = 1 : \Psi.N_n$ **do**
          $\Omega.w^{(k,c(k))} = \Phi.w^{(m,i)} \times \Psi.w^{(n,j)}$;
          $\Omega.p^{(k,c(k),:)} = \left[\Phi.p^{(m,i,:)}; \Psi.p^{(n,j,:)}\right]$;
          $\Omega.\ell^{(k,c(k),:)} = \left[\Phi.\ell^{(m,i,:)}; \Psi.\ell^{(n,j,:)}\right]$;
          $c(k) = c(k) + 1$;
  $\Omega = \texttt{NormaliseGLMB}(\Omega)$;

Algorithm 7: GLMB measurement update.

Input: Predicted GLMB (Φ), measurements (Z)
Parameters: Number of updated components to generate (N_u), detection probability (P_D), clutter intensity (λ), surveillance volume (V)
Output: Updated GLMB (Ω), probability that each measurement is from a target (P_A)

Function $[\Omega, P_A] = \texttt{UpdateGLMB}(\Phi, Z)$ **is**
  $\theta_i = \emptyset$;
  $N_{cdn} = \texttt{Allocate}\left(N_u, \texttt{Poisson}\left(0 : \Phi.M, \texttt{Mean}\left(\Phi.\rho^{(:)}\right)\right)\right)$;
  for $n = 1 : \Phi.C_{max}$ **do**
    $j = 1$;
    $N_{cmp} = \texttt{Allocate}\left(N_{cdn}(n), \Phi.w^{(n,:)}\right)$;
    for $i = 1 : \Phi.N_n$ **do**
      $C = \texttt{CostMatrixAssign}\left(\Phi.p^{(n,i,:)}, \Phi.\ell^{(n,i,:)}, Z, P_D\right)$;
      $A = \texttt{Murty}\left(C, N_{cmp}(i)\right)$;
      for $(a, c) \in A$ **do**
        $\Omega.w^{(n,j)} = \Phi.\rho^{(n)} \times \Phi.w^{(n,i)} \times \lambda^{|Z|} \times V^{-1} \times e^{-c}$;
        for $k = 1 : n$ **do**
          $\ell = \Phi.\ell^{(n,i,k)}$;
          if $a(\ell) > 0$ **then**
            $\Omega.p^{(n,j,k)} = \texttt{TargetUpdate}\left(\Phi.p^{(n,i,k)}, z_{a(\ell)}\right)$;
            $\theta_{a(\ell)} = \left[\theta_{a(\ell)}; (n, j)\right]$;
          else
            $\Omega.p^{(n,j,k)} = \Phi.p^{(n,i,k)}$;
          $\Omega.\ell^{(n,j,k)} = \ell$;
        $j = j + 1$;
  $\Omega = \texttt{NormaliseGLMB}(\Omega)$;
  for $i = 1 : |Z|$ **do**
    $P_A(i) = \texttt{Sum}\left(\Omega.w^{(\theta_i)}\right)$;

Algorithm 8: GLMB estimate extraction.

Input: GLMB density (Φ), current track estimates (τ_x), current track labels (τ_ℓ)
Output: Updated track estimates (τ_x), updated track labels (τ_ℓ)

Function $[\tau_x, \tau_\ell] = \texttt{ExtractEstimatesGLMB}\left(\Phi, \tau_x, \tau_\ell\right)$ **is**
  $n = \text{argmax}_n \left(\Phi.\rho^{(n)}\right)$;
  $i = \text{argmax}_i \left(\Phi.w^{(n,i)}\right)$;
  for $j = 1 : n$ **do**
    $\ell = \Phi.\ell^{(n,i,j)}$;
    $\hat{x} = \texttt{TargetEstimateMAP}\left(\Phi.p^{(n,i,j)}\right)$;
    if $\ell \in \tau_\ell$ **then**
      $\tau_x\left(\ell\right) = \left[\tau_x\left(\ell\right); \left(\Phi.t, \hat{x}\right)\right]$;
    else
      $\tau_\ell = \left[\tau_\ell; \ell\right]$;
      $\tau_x\left(\ell\right) = \left[\left(\Phi.t, \hat{x}\right)\right]$;

Algorithm 9: GLMB pruning.

Input: GLMB density (Φ)
Parameters: Maximim number of components to retain (N_r), minimum number of components to retain for any cardinality (N_c), maximum ratio of highest to lowest component weight (R_m)
Output: Pruned GLMB density (Ω)

Function $\Omega = \texttt{PruneGLMB}\left(\Phi\right)$ **is**
  $[C, I] = \texttt{Sort}\left(\Phi.w, \Phi.\rho\right)$;
  $w_{min} = \Omega.w^{(C(1),I(1))} / R_m$;
  $M_{tot} = 0$;
  $M_{cdn} = \texttt{Zeros}\left(1, \Phi.M\right)$;
  for $i = 1 : |C|$ **do**
    $c = C\left(i\right)$;
    $s = I\left(i\right)$;
    $m = M_{cdn}\left(c\right)$;
    if $\left[\left(M_{tot} < N_r\right)$ ***and*** $\left(\Phi.w^{(c,s)} > w_{min}\right)\right]$ ***or*** $\left(m < N_c\right)$ **then**
      $\Omega.p^{(c,m,:)} = \Phi.p^{(c,s,:)}$;
      $\Omega.\ell^{(c,m,:)} = \Phi.\ell^{(c,s,:)}$;
      $\Omega.w^{(c,m)} = \Phi.w^{(c,s)}$;
      $M_{tot} = M_{tot} + 1$;
      $M_{cdn}\left(c\right) = m + 1$;
  $\Omega = \texttt{NormaliseGLMB}\left(\Omega\right)$;

Most of the routines specified above make use of the function `NormaliseGLMB`, which normalises the component weights of an unnormalised GLMB, as well as computing its cardinality distribution. Pseudo-code for this function is defined in Algorithm 10.

Algorithm 10: GLMB weight normalisation.

Input: Unnormalised GLMB density (Φ)
Output: Normalised GLMB density (Ω)

Function $\Omega = \texttt{NormaliseGLMB}(\Phi)$ **is**
  for $i = 1 : \Phi.C_{max}$ **do**
    $\Omega.\rho^{(i)} = \texttt{Sum}\left(\Phi.w^{(i,:)}\right)$;
    $\Omega.w^{(i,:)} = \Phi.w^{(i,:)} / \Omega.\rho^{(i)}$;
  $\Omega.\rho^{(:)} = \Omega.\rho^{(:)} / \texttt{Sum}\left(\Omega.\rho^{(:)}\right)$;

Algorithms 4-9 also make use of some additional base functions, which are described in Table 3. Finally, in Algorithm 11 we show how to draw a labelled multi-target sample from a GLMB distribution. Although this is not required for the implementation of the GLMB filter, this sampling function is presented here, as it will be used later in this dissertation.

Algorithm 11: Draw samples from a GLMB density.

Input: GLMB density (Φ)
Output: Sampled states (X), sampled labels (L)

Function $[X, L] = \texttt{DrawSampleGLMB}(\Phi)$ **is**
  $n \sim \texttt{DiscreteDistribution}\left(\Phi.\rho^{(:)}\right)$;
  $j \sim \texttt{DiscreteDistribution}\left(\Phi.w^{(n,:)}\right)$;
  for $k = 1 : n$ **do**
    $X^{(k)} \sim \Phi.p^{(n,j,k)}$;
    $L^{(k)} = \Phi.\ell^{(n,j,k)}$;

Table 3: Base functions used in GLMB filter pseudo-code.

Function Name	Description
`TargetPrior` (z)	Generate a prior pdf for a single target, given that it produced the measurement z
`TargetPredict` (p, δ_t)	Propagate the target pdf p forward by time δ_t
`TargetUpdate` (p, z)	Update the prior target pdf p with measurement z
`TargetEstimateMAP` (p)	Maximum a posteriori estimate for a target with pdf p
`ValidationGate` (p)	Returns the measurement validation region for a single target pdf p
`Poisson` (y, λ)	Poisson pdf with mean λ computed at each element of the array y
`Uniform` (S)	Uniform pdf on the set defined by S
`Allocate` (n, w)	Randomised proportional allocation of a scalar number of objects n, into a fixed number of bins with weights given by the array w
`CostMatrixAssign` (X, L, Z, P_D)	Compute a $\|X\| \times (\|Z\| + \|X\|)$ measurement assignment cost matrix for a set of targets with pdfs X and labels L, a set of measurements Z, and detection probability function $P_D(x, \ell)$ (using equation (190))
`SurvivalGraph` (X, L, P_s)	Generate survival graph of Figure (13), for a set of targets with pdfs X and labels L, and survival probability function $P_s(x, \ell)$
`ShortestPaths` (G, k)	List the k cheapest paths and their costs, in increasing order of cost, between the start node and end node in the graph G
`Murty` (C, n)	Use Murty's algorithm to list the cheapest n row-to-column assignments in cost matrix C, in increasing order of cost

2.6.2 *Pseudo-code for the Standard LMB Filter*

First we define a data structure to represent an LMB density, the parameters of which are given in Table 4. The structure is much simpler than that of the GLMB, since the LMB only needs to store a label, a single-target density, and an existence probability for each target. To distinguish LMB densities from GLMB densities, we denote an LMB with a tilde over the symbol (i.e.$\tilde{\Phi}$ and $\tilde{\Omega}$).

Table 4: LMB data structure components.

Symbol	Description
$\tilde{\Phi}.r^{(i)}$	Existence probability of the i-th target
$\tilde{\Phi}.p^{(i)}$	Probability density of the i-th target
$\tilde{\Phi}.\ell^{(i)}$	Label of the i-th target
$\tilde{\Phi}.M$	Number of targets
$\tilde{\Phi}.t$	Continuous time at which the density is valid

The main LMB recursion is given in Algorithm 12, which takes a collection of measurement sets, and returns a list of labelled target trajectory estimates. The functions used in this algorithm are defined in the sequel.

Algorithm 12: LMB filter recursion.

Input: Measurements $(Z_{1:N})$, measurement times $(t_{1:N})$
Output: Track estimates (τ_x), track labels (τ_ℓ)

Function $[\tau_x, \tau_\ell] = \texttt{LMB\_Filter}(Z_{1:N}, t_{1:N})$ **is**
  $\ell_{next} = 1$;
  $\tau_x = \emptyset$;
  $\tau_\ell = \emptyset$;
  $P_{A,1} = \texttt{Zeros}(1, |Z_1|)$;
  for $k = 2 : N$ **do**
    $\left[\tilde{\Omega}_{B,k}, \ell_{next}\right] = \texttt{BirthLMB}(Z_{k-1}, P_{A,k-1}, t_{k-1}, Z_k, t_k, \ell_{next})$;
    if $k = 2$ **then**
      $\tilde{\Omega}_+ = \tilde{\Omega}_{B,k}$;
    else
      $\tilde{\Omega}_{S,k} = \texttt{PredictSurvivalLMB}(\tilde{\Omega}_{k-1}, t_k - t_{k-1})$;
      $\tilde{\Omega}_+ = \texttt{JoinLMB}(\tilde{\Omega}_{S,k}, \tilde{\Omega}_{B,k})$;
    $\left[\tilde{\Omega}_k, P_{A,k}\right] = \texttt{UpdateLMB}(\tilde{\Omega}_+, Z_k)$;
    $[\tau_x, \tau_\ell] = \texttt{ExtractEstimatesLMB}(\tilde{\Omega}_k, \tau_x, \tau_\ell)$;

The four main functions used in Algorithm 12 are defined in Algorithms 13-16 as follows:

- `PredictSurvivalLMB` (Algorithm 13): Takes the LMB density at the current time, and computes the predicted LMB density of surviving targets at the next time.
- `BirthLMB` (Algorithm 14): Compute a birth LMB density using the measurements from the current and previous scans, and the probabilities that each previous measurement was generated by a target (these probabilities are computed as part of the measurement update function in Algorithm 15).
- `UpdateLMB` (Algorithm 15): Takes a predicted LMB, and updates it using a new measurement set to return the posterior LMB density.
- `ExtractEstimatesLMB` (Algorithm 16): Extracts a labelled set of estimates from an LMB density, which is used to update a list of estimated target trajectories.

In Algorithm 15, two additional functions are used to convert between the LMB and GLMB representation. The function `LMBtoGLMB` converts an LMB density to GLMB representation, and is defined in Algorithm 17. The function `ApproximateLMB`, defined by Algorithm 18, approximates a GLMB density in LMB form. In doing so, each single-target density in the approximate LMB becomes a mixture, so mixture reduction is carried out to manage the number of components. The LMB filter makes use of the base functions described in Table 3, and the additional base functions described in Table 5.

Algorithm 13: LMB survival prediction.

Input: LMB density ($\tilde{\Phi}$), prediction time (t),
Parameters: Target survival probability (P_S)
Output: Predicted LMB density ($\tilde{\Omega}$)

Function $\tilde{\Omega} =$ `PredictSurvivalLMB` $\left(\tilde{\Phi}, t\right)$ **is**
  for $i = 1 : \tilde{\Phi}.C_{max}$ **do**
    $\tilde{\Omega}.\ell^{(i)} = \tilde{\Phi}.\ell^{(i)}$;
    $\tilde{\Omega}.p^{(i)} =$ `TargetPredict` $\left(\tilde{\Phi}.p^{(i)}, t - \tilde{\Phi}.t\right)$;
    $\eta_S = \left\langle P_S, \tilde{\Phi}.p^{(i)} \right\rangle$;
    $\tilde{\Omega}.r^{(i)} = \eta_S \times \tilde{\Phi}.r^{(i)}$;
  $\tilde{\Omega}.t = t$;

Algorithm 14: Measurement driven LMB birth density generation.

Input: Previous measurements (Z_{k-1}), probabilities that each previous measurement is from a target ($P_{A,k-1}$), time of previous measurements (t_{k-1}), current measurements (Z_k), current time (t_k), next target label (ℓ_{next})

Parameters: Maximum association probability to generate birth (P_{thresh}), probability that a previously unassigned measurement is from a new target (P_{birth})

Output: LMB birth density ($\tilde{\Omega}$), next label (ℓ_{next})

Function $\left[\tilde{\Omega}, \ell_{next}\right] = \texttt{BirthLMB}\left(Z_{k-1}, P_{A,k-1}, t_{k-1}, Z_k, t_k, \ell_{next}\right)$ **is**
- $c = 1$;
- **for** $i = 1 : |Z_{k-1}|$ **do**
  - **if** $P_{A,k-1}^{(i)} < P_{thresh}$ **then**
    - $p = \texttt{TargetPrior}\left(Z_{k-1}^{(i)}\right)$;
    - $p_+ = \texttt{TargetPredict}\left(p, t_k - t_{k-1}\right)$;
    - $V = \texttt{ValidationGate}\left(p_+\right)$;
    - `keep` $=$ `false`;
    - **for** $j = 1 : |Z_k|$ **do**
      - **if** $z_{k,j} \in V$ **then**
        - `keep` $=$ `true`;
    - **if** `keep` **then**
      - $\tilde{\Omega}.p^{(c)} = p_+$;
      - $\tilde{\Omega}.\ell^{(c)} = \ell_{next}$;
      - $\tilde{\Omega}.r^{(c)} = P_{birth}$;
      - $c = c + 1$;
      - $\ell_{next} = \ell_{next} + 1$;
- $\tilde{\Omega}.t = t_k$;

Algorithm 15: LMB measurement update.

Input: LMB density ($\tilde{\Phi}$), measurements (Z)

Output: Posterior LMB density ($\tilde{\Omega}$)

Function $\tilde{\Omega} = \texttt{UpdateLMB}\left(\tilde{\Phi}, Z\right)$ **is**
- $C = \texttt{ClusterTracks}\left(\tilde{\Phi}, Z\right)$;
- $\tilde{\Omega} = \emptyset$;
- $P_A = \texttt{Zeros}\left(1, |Z|\right)$;
- **for** $W \in C$ **do**
  - $\Phi_W = \texttt{LMBtoGLMB}\left(\tilde{\Phi}, W\right)$;
  - $\left[\Omega_W, P_{A,W}\right] = \texttt{UpdateGLMB}\left(\Phi_W, Z\right)$;
  - $\tilde{\Omega}_W = \texttt{ApproximateLMB}\left(\Omega_W\right)$;
  - $\tilde{\Omega} = \texttt{JoinLMB}\left(\tilde{\Omega}, \tilde{\Omega}_W\right)$;
  - $P_A = P_A + P_{A,W}$;

Algorithm 16: LMB estimate extraction.

Input: Track estimates (τ_x), track labels (τ_ℓ), LMB density ($\tilde{\Phi}$)
Parameters: Existence probability threshold (r_{thresh})
Output: Updated track estimates (τ_x), updated track labels (τ_ℓ)

Function $[\tau_x, \tau_\ell] = \texttt{ExtractEstimatesLMB}\left(\tau_x, \tau_\ell, \tilde{\Phi}, T\right)$ **is**
  for $i = 1 : \tilde{\Phi}.C_{max}$ **do**
    if $\tilde{\Phi}.r^{(i)} > r_{thresh}$ **then**
      $\ell = \tilde{\Phi}.\ell^{(i)}$;
      $\hat{x} = \texttt{TargetEstimateMAP}\left(\tilde{\Phi}.p^{(i)}\right)$;
      if $\ell \in \tau_\ell$ **then**
        $\tau_x(\ell) = \left[\tau_x(\ell); \left(\tilde{\Phi}.t, \hat{x}\right)\right]$;
      else
        $\tau_\ell = [\tau_\ell; \ell]$;
        $\tau_x(\ell) = \left[\left(\tilde{\Phi}.t, \hat{x}\right)\right]$;

Algorithm 17: LMB to GLMB conversion.

Input: LMB density ($\tilde{\Phi}$)
Output: GLMB density (Ω)

Function $\Omega = \texttt{LMBtoGLMB}\left(\tilde{\Phi}\right)$ **is**
  for $n = 1 : \tilde{\Phi}.C_{max}$ **do**
    $c = 1$;
    for $I \in \mathcal{F}_n\left(\left\{1, \ldots, \tilde{\Phi}.M\right\}\right)$ **do**
      $\Omega.w^{(n,c)} = \texttt{Product}\left(1 - \tilde{\Phi}.r^{(:)}\right)$;
      for $j = 1 : n$ **do**
        $\Omega.p^{(n,c,j)} = \tilde{\Phi}.p^{(I_j)}$;
        $\Omega.\ell^{(n,c,j)} = \tilde{\Phi}.\ell^{(I_j)}$;
        $\Omega.w^{(n,c)} = \Omega.w^{(n,c)} \times \tilde{\Phi}.r^{(I_j)} \div \left(1 - \tilde{\Phi}.r^{(I_j)}\right)$;
      $c = c + 1$;

Algorithm 18: Approximate a GLMB as an LMB.

Input: GLMB density (Φ)
Output: LMB density ($\tilde{\Omega}$)

Function $\tilde{\Omega} = \texttt{ApproximateLMB}(\Phi)$ **is**
  for $n = 1 : \Phi.C_{max}$ **do**
    for $i = 1 : \Phi.N_n$ **do**
      for $j = 1 : n$ **do**
        $k = \texttt{Find}\left(\tilde{\Omega}.\ell^{(:)}, \Phi.\ell^{(n,i,j)}\right)$;
        if $k > 0$ **then**
          $\tilde{\Omega}.r^{(k)} = \tilde{\Omega}.r^{(k)} + \left(\Phi.\rho^{(n)} \times \Phi.w^{(n,i,j)}\right)$;
          $p^{(k)} = \left[p^{(k)}; \Phi.p^{(n,i,j)}\right]$;
          $w^{(k)} = \left[w^{(k)}; \Phi.\rho^{(n)} \times \Phi.w^{(n,i,j)}\right]$;
        else
          $k = \tilde{\Omega}.M + 1$;
          $\tilde{\Omega}.\ell^{(k)} = \Phi.\ell^{(n,i,j)}$;
          $\tilde{\Omega}.r^{(k)} = \Phi.\rho^{(n)} \times \Phi.w^{(n,i,j)}$;
          $p^{(k)} = \left[\Phi.p^{(n,i,j)}\right]$;
          $w^{(k)} = \left[\Phi.\rho^{(n)} \times \Phi.w^{(n,i,j)}\right]$;
  for $i = 1 : \tilde{\Omega}.C_{max}$ **do**
    $\tilde{\Omega}.p^{(i)} = \texttt{MixtureReduce}\left(p^{(i)}, w^{(i)}\right)$;

Table 5: Base functions used in the LMB filter pseudo-code.

Function Name	Description
$\texttt{JoinLMB}\left(\tilde{\Phi}, \tilde{\Psi}\right)$	Join two LMB structures together, by concatenating the arrays of single-target densities, target labels and existence probabilities.
$\texttt{MixtureReduce}\left(p, w\right)$	Reduce the number of components in a pdf mixture with densities p and weights w
$\texttt{Find}\left(X, x\right)$	Find the index of the element x in the array X, or return 0 if it is not present
$\texttt{ClusterTracks}\left(X, Z\right)$	Construct disjoint clusters from the tracks in X, such that no single measurement in Z falls in the validation gate of two tracks from different clusters

3 MULTI-TARGET TRACKING IN THE PRESENCE OF MERGED MEASUREMENTS

Most tracking algorithms in the literature have been developed with the assumption that the measurements follow the standard multi-object observation model. Fundamentally, this model assumes that each target generates at most one measurement, and that each measurement is generated by at most one target. The main reasons for the popularity of this model are its simplicity, and that it facilitates computationally efficient tracking algorithms. However, there are important cases where this model is too restrictive, and leads to poor performance. For example, when the measurements from two or more targets become merged, the assumption that each measurement comes from at most one target is violated. The contribution of this chapter is the development of the first multi-target tracker for a merged measurement model, based on labelled random finite sets. The proposed tracker yields trajectory estimates for an unknown and time-varying number of targets, in the presence of measurement merging. The main results of this chapter have previously appeared in the author's conference paper [19], journal article [21], and partially in the co-authored journal article [157].

Section 3.1 provides an introduction to tracking with merged measurements, and describes some existing techniques. An RFS-based likelihood model is proposed in Section 3.2 to handle merging in a principled manner. An exact but intractable multi-target tracker based on this likelihood is derived in Section 3.3. Section 3.4 presents a method to approximate an arbitrary labelled RFS density as a GLMB, which is used in Section 3.5 to derive two techniques for implementing the algorithm in practice. Further details of the implementation are given in Section 3.6, some simulation results are presented in Section 3.7, and finally, pseudo-code is provided in Section 3.8.

3.1 INTRODUCTION

In the majority of multi-target tracking algorithms, the measurement model is based on the premise that all targets generate measurements independently of each other. This assumption is made because it leads to computationally cheaper algorithms, however, most real-world sensors violate this assumption when the targets are close together in the measurement space. In such cases, a group of closely spaced targets may only produce a single detection, referred to as a *merged measurement*. When this happens, if the tracker assumes that the targets generate measurements independently of each other, it will usually infer that some targets have disappeared, when in fact their measurements have become merged.

A common situation involving merged measurements is when the sensor has finite resolution [44]. For example, most radar and sonar systems effectively operate by dividing the measurement space into discrete detection cells, and when multiple targets fall within the same cell, they produce a merged measurement. Another example is in computer vision, where detection algorithms may produce merged measurements for groups of objects appearing close together or occluding one-another in an image. Since merged measurements are unavoidable when processing real world sensor data, and methods that assume conditionally independent measurements perform poorly when this occurs, it is fundamentally important to investigate algorithms that can deal with this phenomenon in a principled manner.

A number of Bayesian multi-target tracking algorithms have been proposed to handle merged measurements. One such algorithm is Joint Probabilistic Data Association (JPDA), which was first applied to a finite resolution sensor model in [36]. The proposed algorithm (called JPDAM) used a grid-based model for computing merging probabilities for two targets, which were then incorporated into the JPDA update. This technique was limited to cases where at most two targets are merged at a time. A useful extension to this was proposed in [183, 184] which can handle merged measurements from an arbitrary number of targets, by defining a more general resolution model based on combining pairwise merging probabilities. This provides a tractable approximation to the probabilities of different resolution events, which are again incorporated into the JPDA update. The inclusion of the resolution model in JPDA has been shown to improve its performance in the presence of merged measurements,

however, the fundamental limitation of these methods is that they implicitly assume a fixed and known number of targets.

Several other methods have also been applied to tracking with merged measurements. An approach based on Multiple Hypothesis Tracking (MHT) was proposed in [108], which used a two-target resolution model to maintain tracks when merging occurs. A similar resolution model was applied to multiple model JPDA in [30, 90], which is again limited to a maximum of two targets per measurement. Various numerical data association techniques have been developed for merged measurements in [104, 37] (probabilistic data association), and [102] (Markov chain Monte Carlo). In addition, probabilistic MHT (PMHT) [45], and existence based methods such as linear multi-target integrated PDA (LM-IPDA) [150], and integrated track splitting (ITS) [149] have also been applied to this problem. These are all useful techniques, however, with the exception of LM-IPDA (which trades off performance for reduced complexity), they only handle a small number of targets. Moreover, due to the nature of their formulation, it is difficult to establish any results regarding their Bayes optimality, in the sense of minimising the posterior Bayes risk [135].

The random finite set framework provides a principled framework for multi-object estimation for an unknown and time-varying number of objects [135]. One of the first methods to be derived using this framework was the probability hypothesis density (PHD) filter [126], and Mahler proposed a technique in [130] for applying this to unresolved targets by modelling the multi-target state as an RFS of point clusters. This is a principled approximation, which can deal with a variable number of targets. However, it does not provide labelled track estimates, and the model is limited to cases where merging only arises when the targets are close together in the state space. The latter is too restrictive in cases where the measurement space is a lower dimensional projection of the state space (e.g. bearings-only sensors), because points which are well separated in the state space may become close together, and therefore unresolvable, when projected onto the measurement space.

One criticism of the RFS framework has been that it resulted in algorithms that do not maintain target labels over time, i.e. performing multi-object filtering instead of true multi-object tracking. It was shown in [198, 195] that the RFS framework does allow estimation of target tracks, and a computationally feasible multi-target tracker for the standard sensor model was derived, known as the generalised labelled multi-Bernoulli (GLMB) filter. This chapter further generalises the result in [198] to a sensor

model that includes merged measurements, thereby making it suitable for application to a wider variety of real world problems. To achieve this, a multi-target likelihood function based on random finite sets is introduced, that takes into account the possible merging of target generated measurements. The key feature of this likelihood function is that it considers the likely partitions of the target set, and the feasible assignments of measurements to groups of targets within these partitions. To our knowledge, this is the first provably Bayes optimal multi-object tracker that accommodates merged measurements, with unknown and time-varying number of targets. An additional advantage of this approach is that it can be easily parallelised to facilitate potential implementation in real-time tracking systems.

3.2 MULTI-OBJECT LIKELIHOOD MODEL FOR MERGED MEASUREMENTS

In [135], Mahler proposed an RFS-based technique for filtering with unresolved targets, which modelled the multi-target state as a set of point clusters. A point cluster was defined as a group of unresolved targets, and was represented as a location in the single-target state space, augmented with a positive integer to represent the number of targets that are effectively co-located at that point (relative to the sensor resolution). The likelihood function for this model involves a sum over partitions of the measurement set, making a direct implementation computationally prohibitive. A PHD filter based on this model was proposed in [130], however, it has not been implemented in practice.

The point cluster model proposed in [135] is intuitively appealing in cases where the measurements can only become merged when the targets are close together in the state space. However, this implicitly assumes that the merging only depends on the target states. There are important cases where this assumption is too restrictive, a prime example of which is bearings-only tracking. In bearings-only tracking, targets that cannot be resolved along the line of sight may be separated by a considerable distance in the state space. The location of the sensor has a significant impact on whether measurements become merged or not, but this effect cannot be accommodated using the point cluster model. This occurs in general when the measurement space is a lower dimensional projection of the state space, such that well separated target states may give rise to closely spaced measurements.

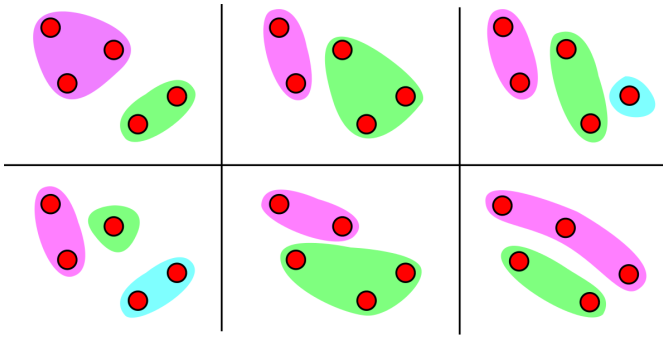


Figure 15: Example of some of the partitions of a set of five targets. Each shaded group of targets can give rise to at most one measurement.

Necessary to address this problem is a likelihood function that accommodates measurement merging, not only when targets are nearby in the state space, but more generally when they are nearby in the measurement space. Achieving this requires considering the partitions of the target set, where each group within a partition produces at most one (merged) measurement.

Definition 5. A *partition* of a set A is a disjoint collection of subsets of A, whose union is equal to A.

Figure 15 depicts an example of some of the partitions of a set of five targets. In the diagram, only six of the total of 15 partitions are shown. In general, the number of partitions of a set of N targets is given by the N-th Bell number, which becomes extremely large as N increases. However, in the context of multi-target tracking, it is usually possible to eliminate many partitions from consideration, since they are physically impossible or extremely unlikely to occur.

A model for the multi-object measurement likelihood is defined as a sum over partitions of the target set as follows,

$$g\left(Z|\boldsymbol{X}\right) = \sum_{\mathcal{U}(\boldsymbol{X})\in\mathcal{P}(\boldsymbol{X})} \mathring{g}\left(Z|\mathcal{U}\left(\boldsymbol{X}\right)\right) \tag{199}$$

where $\mathcal{P}\left(\boldsymbol{X}\right)$ is the set of all partitions of $\boldsymbol{X}$, and $\mathring{g}\left(Z|\mathcal{U}\left(\boldsymbol{X}\right)\right)$ is the measurement likelihood conditioned on the targets being observed according to a given partition $\mathcal{U}\left(\boldsymbol{X}\right)$. This bears some similarity to the approach taken in [184], insofar as they both consider partitions of the targets to form resolution events. However, the key difference is that the likelihood (199) is defined as a function of a set of target states, instead of a vector with fixed dimension. This allows us to utilise FISST principles to arrive at an expression for the posterior multi-object density.

To obtain an expression for $\mathring{g}(Z|\mathcal{U}(\boldsymbol{X}))$, the standard measurement likelihood in (173) is generalised to groups of targets as follows[1]

$$\mathring{g}(Z|\mathcal{U}(\boldsymbol{X})) = e^{-\langle \kappa,1\rangle}\kappa^{Z}\sum_{\theta\in\Theta(\mathcal{U}(\boldsymbol{X}))}\left[\check{\psi}_{Z}(\cdot;\theta)\right]^{\mathcal{U}(\boldsymbol{X})} \tag{200}$$

where $\Theta(\mathcal{U}(\boldsymbol{X}))$ is defined as the set of all mappings of groups of target labels in $\mathcal{U}(\mathcal{L}(\boldsymbol{X}))$ to measurement indices in Z (i.e. $\theta : \mathcal{U}(\mathcal{L}(\boldsymbol{X})) \to \{0,1,...,|Z|\}$, where $[\theta(L) = \theta(J) > 0] \Rightarrow [L = J]$), and $\check{\psi}_{Z}(\boldsymbol{Y};\theta)$ is a "group likelihood" defined by

$$\check{\psi}_{Z}(\boldsymbol{Y};\theta) = \begin{cases} \frac{\check{P}_{D}(\boldsymbol{Y})\check{g}\left(z_{\theta(\mathcal{L}(\boldsymbol{Y}))}|\boldsymbol{Y}\right)}{\kappa\left(z_{\theta(\mathcal{L}(\boldsymbol{Y}))}\right)}, & \theta(\mathcal{L}(\boldsymbol{Y})) > 0 \\ 1-\check{P}_{D}(\boldsymbol{Y}), & \theta(\mathcal{L}(\boldsymbol{Y})) = 0 \end{cases}. \tag{201}$$

In (201), $\check{P}_{D}(\boldsymbol{Y})$ denotes the detection probability for target group $\boldsymbol{Y}$, and $\check{g}\left(z_{\theta(\mathcal{L}(\boldsymbol{Y}))}|\boldsymbol{Y}\right)$ is the likelihood that a target group $\boldsymbol{Y}$ produces a measurement $z_{\theta(\mathcal{L}(\boldsymbol{Y}))}$. For our simulations in Section 3.7, $\check{g}(\cdot|\boldsymbol{Y})$ is modelled as a Gaussian centered on the mean of the group, with a standard deviation that increases as the number of targets in the group increases. However, the form of this function in general will depend on the sensor characteristics, and there are no restrictions placed on its modelling, provided that it can be tractably computed. Note that in (200), the exponent $\mathcal{U}(\boldsymbol{X})$ is a set of target sets, and the base is a real valued function whose argument is a target set. Substituting (200) into (199) yields

$$g(Z|\boldsymbol{X}) = e^{-\langle \kappa,1\rangle}\kappa^{Z}\sum_{\substack{\mathcal{U}(\boldsymbol{X})\in\mathcal{P}(\boldsymbol{X})\\ \theta\in\Theta(\mathcal{U}(\boldsymbol{X}))}}\left[\check{\psi}_{Z}(\cdot;\theta)\right]^{\mathcal{U}(\boldsymbol{X})}, \tag{202}$$

which is referred to as the merged measurement likelihood function.

3.3 A GENERAL FORM FOR A MERGED MEASUREMENT TRACKER

This section contains a derivation of an exact but computationally intractable form for a labelled RFS Bayes recursion, based on the merged measurement likelihood defined by (202). This requires the following definition, which can be used to represent the density of any labelled RFS [21, 157].

1 For notational convenience, $\mathcal{U}(\boldsymbol{X})$ is used to denote a partition of the label-state pairs in the labelled set $\boldsymbol{X}$, and $\mathcal{U}(\mathcal{L}(\boldsymbol{X}))$ to denote the equivalent partition of the labels only.

Definition 6. Given a labeled multi-object density $\boldsymbol{\pi}$ on $\mathcal{F}(\mathbb{X}) \times \mathbb{L}$, and any positive integer n, we define the *joint existence probability* of the label set $\{\ell_1, \ldots \ell_n\}$ by

$$w(\{\ell_1, \ldots, \ell_n\}) \triangleq \int \boldsymbol{\pi}(\{(x_1, \ell_1), \ldots, (x_n, \ell_n)\}) \, d(x_1, \ldots, x_n)$$

and the joint probability density (on $\mathbb{X}^n$) of $x_1, \ldots, x_n$, conditional on their corresponding labels $\ell_1, \ldots, \ell_n$ by

$$p(\{(x_1, \ell_1), \ldots, (x_n, \ell_n)\}) \triangleq \frac{\boldsymbol{\pi}(\{(x_1, \ell_1), \ldots, (x_n, \ell_n)\})}{w(\{\ell_1, \ldots, \ell_n\})}.$$

For $n = 0$, we define $w(\emptyset) \triangleq \boldsymbol{\pi}(\emptyset)$ and $p(\emptyset) \triangleq 1$. It is implicit that $p(\boldsymbol{X})$ is defined to be zero whenever $w(\mathcal{L}(\boldsymbol{X}))$ is zero. Consequently the labelled RFS density can be expressed as

$$\boldsymbol{\pi}(\boldsymbol{X}) = w(\mathcal{L}(\boldsymbol{X})) \, p(\boldsymbol{X}). \tag{203}$$

Note that $\sum_{L \in \mathcal{F}(\mathbb{L})} w(L) = 1$, and since $\boldsymbol{\pi}$ is symmetric in its arguments, it follows that w is also symmetric in $\ell_1, \ldots, \ell_n$. Hence, $w(\cdot)$ is a probability distribution on $\mathcal{F}(\mathbb{L})$.

For the work in this chapter, we often encounter labelled RFS densities that are mixtures of (203), which we formally define as follows.

Definition 7. A *labelled RFS mixture* density on state space $\mathbb{X}$ and discrete label space $\mathbb{L}$, is a density of the form

$$\boldsymbol{\pi}(\boldsymbol{X}) = \Delta(\boldsymbol{X}) \sum_{c \in \mathbb{C}} w^{(c)}(\mathcal{L}(\boldsymbol{X})) \, p^{(c)}(\boldsymbol{X}) \tag{204}$$

where

$$\sum_{L \subseteq \mathbb{L}} \sum_{c \in \mathbb{C}} w^{(c)}(L) = 1, \tag{205}$$

$$\int p^{(c)}(\{(x_1, \ell_1), ..., (x_n, \ell_n)\}) \, d(x_1, \ldots, x_n) = 1, \tag{206}$$

and $p^{(c)}(\boldsymbol{X})$ is symmetric in the elements of $\boldsymbol{X}$.

Note that Definition 7 can be regarded as a generalisation of Definition 6, however, (204) can also be expressed in the form of (203).

A labelled RFS mixture density differs from the GLMB density of Definition 3 in that each $p^{(c)}(\boldsymbol{X})$ is a joint density of all the targets in $\boldsymbol{X}$, whereas in Definition 3, it is restricted to a product of single-target densities. Note that any labeled RFS density can be written in the mixture form (204), and this form is used because it is convenient for the update step of the proposed tracker. The following lemma is required to derive the filter prediction and update.

Lemma 8. *Let $\mathbb{X}$ be a state space and $\mathbb{L}$ a discrete label space. Then, for two functions $h : \mathcal{F}(\mathbb{L}) \to \mathbb{R}$ and $g : \mathcal{F}(\mathbb{X} \times \mathbb{L}) \to \mathbb{R}$, where g is integrable on $\mathcal{F}(\mathbb{X} \times \mathbb{L})$,*

$$\int \Delta(\boldsymbol{X})\, h(\mathcal{L}(\boldsymbol{X}))\, g(\boldsymbol{X})\, \delta\boldsymbol{X} = \sum_{L \subseteq \mathbb{L}} h(L)\, g_L \tag{207}$$

where

$$g_L = \int g\left(\left\{(x_1, \ell_1), ..., (x_{|L|}, \ell_{|L|})\right\}\right) d(x_1, \ldots, x_{|L|}). \tag{208}$$

Proof. This is a straightforward generalisation of [198, Lemma 3], to the case where g is non-separable.

$$\begin{aligned}
&\int \delta_{|\boldsymbol{X}|}(|\mathcal{L}(\boldsymbol{X})|)\, h(\mathcal{L}(\boldsymbol{X}))\, g(\boldsymbol{X})\, \delta\boldsymbol{X} \\
&\quad = \sum_{n=0}^{\infty} \frac{1}{n!} \sum_{(\ell_1,\ldots,\ell_n) \in \mathbb{L}^n} \delta_n(|\{\ell_1, \ldots, \ell_n\}|)\, h(\{\ell_1, \ldots, \ell_n\}) \\
&\qquad\qquad \times \int (g(\{(x_1, \ell_1), \ldots, (x_n, \ell_n)\}))\, d(x_1, \ldots, x_n) \\
&\quad = \sum_{n=0}^{\infty} \sum_{\{\ell_1,\ldots,\ell_n\} \in \mathcal{F}_n(\mathbb{L})} h(\{\ell_1, \ldots, \ell_n\})\, g_{\{\ell_1,\ldots,\ell_n\}} \\
&\quad = \sum_{L \subseteq \mathbb{L}} h(L)\, g_L
\end{aligned}$$

□

The multi-object prediction and merged measurement update can now be derived for the general labelled RFS mixture model, as stated by the following two propositions. Note that for any function $f : \mathcal{F}(\mathbb{X} \times \mathbb{L}) \to \mathbb{R}$, we have used the convention $f_{\{\ell_1,\ldots,\ell_n\}}(x_1, \ldots, x_n) = f(\{(x_1, \ell_1), \ldots, (x_n, \ell_n)\})$ to denote it as a function on $\mathbb{X}^n$, given the set of labels $\{\ell_1, \ldots, \ell_n\}$.

Proposition 9. *If the previous multi-object posterior is a labelled RFS mixture density with probability distribution of the form (204), then the predicted multi-object density under the transition kernel (169) is given by*

$$\pi_+(\boldsymbol{X}_+) = \Delta(\boldsymbol{X}_+) \sum_{c \in \mathbb{C}} \sum_{L \subseteq \mathbb{L}} w_{+,L}^{(c)}(\mathcal{L}(\boldsymbol{X}_+))\, p_{+,L}^{(c)}(\boldsymbol{X}_+) \tag{209}$$

where

$$w_{+,L}^{(c)}(J) = 1_L(J \cap \mathbb{L})\, w_B(J - \mathbb{L})\, w^{(c)}(L)\, \eta_S^{(c)}(L) \tag{210}$$

$$p_{+,L}^{(c)}(\boldsymbol{X}_+) = [p_B(\cdot)]^{\boldsymbol{X}_+ - \mathbb{X}\times\mathbb{L}}\, p_{S,L}^{(c)}(\boldsymbol{X}_+ \cap \mathbb{X}\times\mathbb{L}) \tag{211}$$

$$p_{S,L}^{(c)}(\boldsymbol{S}) = \frac{\int p_L^{(c)}(x_1,\ldots x_{|L|}) \prod_{i=1}^{|L|} \Phi(\boldsymbol{S}; x_i, \ell_i)\, d(x_1,\ldots,x_{|L|})}{\eta_S^{(c)}(L)} \tag{212}$$

$$\eta_S^{(c)}(L) = \int\int p_L^{(c)}(x_1,\ldots,x_{|L|}) \prod_{i=1}^{|L|} \Phi(\boldsymbol{S}; x_i, \ell_i)\, d(x_1,\ldots,x_{|L|})\, \delta\boldsymbol{S} \tag{213}$$

$$p_L^{(c)}(x_1,\ldots,x_{|L|}) \triangleq p^{(c)}(\{(x_1,\ell_1),\ldots(x_{|L|},\ell_{|L|})\}) \tag{214}$$

Proof. The density of surviving targets is

$$\begin{aligned}
\pi_S(\boldsymbol{W}) &= \int f_S(\boldsymbol{W}|\boldsymbol{X})\, \pi(\boldsymbol{X})\, \delta\boldsymbol{X} \\
&= \int \Delta(\boldsymbol{W})\, \Delta(\boldsymbol{X})\, 1_{\mathcal{L}(\boldsymbol{X})}(\mathcal{L}(\boldsymbol{W}))\, [\Phi(\boldsymbol{W};\cdot)]^{\boldsymbol{X}} \\
&\qquad \times \Delta(\boldsymbol{X}) \sum_{c\in\mathbb{C}} w^{(c)}(\mathcal{L}(\boldsymbol{X}))\, p^{(c)}(\boldsymbol{X})\, \delta\boldsymbol{X} \\
&= \Delta(\boldsymbol{W}) \sum_{c\in\mathbb{C}} \sum_{L\subseteq\mathbb{L}} 1_L(\mathcal{L}(\boldsymbol{W}))\, w^{(c)}(L)\, \eta_S^{(c)}(L)\, p_{S,L}^{(c)}(\boldsymbol{W})
\end{aligned} \tag{215}$$

where the last line is obtained by swapping the order of summation and integration, and applying Lemma 8 with $h(L) = 1_L(\mathcal{L}_W)\, w^{(c)}(L)$ and $g(\boldsymbol{X}) = [\Phi(\boldsymbol{W};\cdot)]^{\boldsymbol{X}}\, p^{(c)}(\boldsymbol{X})$. Hence the predicted density, which is the product of the birth and survival densities, is given by

$$\begin{aligned}
\pi_+(\boldsymbol{X}_+) &= f_B(\boldsymbol{X}_B)\, \pi_S(\boldsymbol{X}_S) \\
&= \Delta(\boldsymbol{X}_B)\, \Delta(\boldsymbol{X}_S) \sum_{c\in\mathbb{C}} \sum_{L\subseteq\mathbb{L}} w_B(\mathcal{L}(\boldsymbol{X}_B))\, 1_L(\mathcal{L}(\boldsymbol{X}_S)) \\
&\qquad \times w^{(c)}(L)\, \eta_S^{(c)}(L)\, [p_B(\cdot)]^{\boldsymbol{X}_B}\, p_{S,L}^{(c)}(\boldsymbol{X}_S) \\
&= \Delta(\boldsymbol{X}_+) \sum_{c\in\mathbb{C}} \sum_{L\subseteq\mathbb{L}} w_B(\mathcal{L}(\boldsymbol{X}_+) - \mathbb{L})\, 1_L(\mathcal{L}(\boldsymbol{X}_+) \cap \mathbb{L}) \\
&\qquad \times w^{(c)}(L)\, \eta_S^{(c)}(L)\, [p_B(\cdot)]^{\boldsymbol{X}_+ - \mathbb{X}\times\mathbb{L}}\, p_{S,L}^{(c)}(\boldsymbol{X}_+ \cap \mathbb{X}\times\mathbb{L}) \\
&= \Delta(\boldsymbol{X}_+) \sum_{c\in\mathbb{C}} \sum_{L\subseteq\mathbb{L}} w_{+,L}^{(c)}(\mathcal{L}(\boldsymbol{X}_+))\, p_{+,L}^{(c)}(\boldsymbol{X}_+).
\end{aligned}$$

□

Proposition 10. *If the prior is a labelled RFS mixture density with probability distribution of the form* (204)*, then the posterior multi-object density under the merged measurement likelihood function* (202) *is given by*

$$\pi\left(\boldsymbol{X}|Z\right)=\Delta\left(\boldsymbol{X}\right)\sum_{c\in\mathbb{C}}\sum_{\substack{\mathcal{U}(\boldsymbol{X})\in\mathcal{P}(\boldsymbol{X})\\ \theta\in\Theta(\mathcal{U}(\boldsymbol{X}))}}w_{Z}^{(c,\theta)}\left(\mathcal{L}\left(\boldsymbol{X}\right);\mathcal{U}\left(\mathcal{L}\left(\boldsymbol{X}\right)\right)\right)p^{(c,\theta)}\left(\boldsymbol{X}|Z,\mathcal{U}\right) \tag{216}$$

where

$$w_{Z}^{(c,\theta)}\left(L;\mathcal{U}\left(L\right)\right)=\frac{w^{(c)}\left(L\right)\eta_{Z}^{(c,\theta)}\left(\mathcal{U}\left(L\right)\right)}{\sum_{c\in\mathbb{C}}\sum_{J\subseteq\mathbb{L}}\sum_{\substack{\mathcal{U}(J)\in\mathcal{P}(J)\\ \theta\in\Theta(\mathcal{U}(J))}}w^{(c)}\left(J\right)\eta_{Z}^{(c,\theta)}\left(J,\mathcal{U}\right)} \tag{217}$$

$$p^{(c,\theta)}\left(\boldsymbol{X}|Z,\mathcal{U}\right)=\frac{\left[\breve{\psi}_{Z}\left(\cdot;\theta\right)\right]^{\mathcal{U}(\boldsymbol{X})}p^{(c)}\left(\boldsymbol{X}\right)}{\eta_{Z}^{(c,\theta)}\left(\mathcal{L}\left(\boldsymbol{X}\right),\mathcal{U}\right)} \tag{218}$$

$$\eta_{Z}^{(c,\theta)}\left(L,\mathcal{U}\left(L\right)\right)=\int\left[\breve{\psi}_{Z}\left(\cdot;\theta\right)\right]^{\mathcal{U}(L)}p_{L}^{(c)}\left(x_{1},\ldots,x_{|L|}\right)d\left(x_{1},\ldots,x_{|L|}\right) \tag{219}$$

Proof. The product of the prior distribution and the likelihood is

$$\begin{aligned}
&\pi\left(\boldsymbol{X}\right)g\left(Z|\boldsymbol{X}\right)\\
&\quad=\Delta\left(\boldsymbol{X}\right)\lambda\sum_{c\in\mathbb{C}}\sum_{\substack{\mathcal{U}(\boldsymbol{X})\in\mathcal{P}(\boldsymbol{X})\\ \theta\in\Theta(\mathcal{U}(\boldsymbol{X}))}}\left(\mathcal{L}\left(\boldsymbol{X}\right)\right)w^{(c)}\left(\mathcal{L}\left(\boldsymbol{X}\right)\right)\left[\breve{\psi}_{Z}\left(\cdot;\theta\right)\right]^{\mathcal{U}(\boldsymbol{X})}p^{(c)}\left(\boldsymbol{X}\right)\\
&\quad=\Delta\left(\boldsymbol{X}\right)\lambda\sum_{c\in\mathbb{C}}\sum_{\substack{\mathcal{U}(\boldsymbol{X})\in\mathcal{P}(\boldsymbol{X})\\ \theta\in\Theta(\mathcal{U}(\boldsymbol{X}))}}\left(\mathcal{L}\left(\boldsymbol{X}\right)\right)w^{(c)}\left(\mathcal{L}\left(\boldsymbol{X}\right)\right)\\
&\qquad\qquad\times\eta_{Z}^{(c,\theta)}\left(\mathcal{L}\left(\boldsymbol{X}\right),\mathcal{U}\left(\mathcal{L}\left(\boldsymbol{X}\right)\right)\right)p^{(c,\theta)}\left(\boldsymbol{X}|Z,\mathcal{U}\right),
\end{aligned} \tag{220}$$

and its integral with respect to the multi-object state is

$$\begin{aligned}
&\int\pi\left(\boldsymbol{X}\right)g\left(Z|\boldsymbol{X}\right)\delta\boldsymbol{X}\\
&\quad=\lambda\sum_{c\in\mathbb{C}}\sum_{n=0}^{\infty}\frac{1}{n!}\sum_{(\ell_{1},\ldots,\ell_{n})\in\mathbb{L}^{n}}\sum_{\substack{\mathcal{U}(\{\ell_{1},\ldots,\ell_{n}\})\in\mathcal{P}(\{\ell_{1},\ldots\ell_{n}\})\\ \theta\in\Theta(\mathcal{U}(\{\ell_{1},\ldots,\ell_{n}\}))}}w^{(c)}\left(\{\ell_{1},\ldots,\ell_{n}\}\right)\\
&\qquad\times\eta_{Z}^{(c,\theta)}\left(\{\ell_{1},\ldots,\ell_{n}\},\mathcal{U}\right)\\
&\qquad\times\int p^{(c,\theta)}\left(\{(x_{1},\ell_{1}),\ldots,(x_{n},\ell_{n})\}|Z,\mathcal{U}\right)d\left(x_{1},\ldots x_{n}\right)\\
&\quad=\lambda\sum_{c\in\mathbb{C}}\sum_{n=0}^{\infty}\sum_{L\in\mathcal{F}_{n}(\mathbb{L})}\sum_{\substack{\mathcal{U}(L)\in\mathcal{P}(L)\\ \theta\in\Theta(\mathcal{U}(L))}}\left(L\right)w^{(c)}\left(L\right)\eta_{Z}^{(c,\theta)}\left(L,\mathcal{U}\right)
\end{aligned}$$

$$= \lambda \sum_{c\in\mathbb{C}} \sum_{L\subseteq\mathbb{L}} \sum_{\substack{\mathcal{U}(L)\in\mathcal{P}(L)\\ \theta\in\Theta(\mathcal{U}(L))}} (L)\, w^{(c)}(L)\, \eta_Z^{(c,\theta)}(L,\mathcal{U}) \tag{221}$$

The result (216) follows by substituting (220) and (221) into the Bayes update equation (56). □

Propositions 9 and 10 define the exact form for the multi-object Bayes filter under the merged measurement likelihood model. However, it is not a tractable recursion, and therefore it cannot be implemented in practice. To arrive at a tractable algorithm requires an approximation, in order to limit the filtering density to a form that can be easily represented and processed using the Bayes recursion. The following sections propose such an approximation, and two alternative methods for implementing a tractable GLMB recursion for the merged measurement likelihood model.

3.4 GLMB APPROXIMATION

Consider a labeled RFS density of the following form

$$\boldsymbol{\pi}(\boldsymbol{X}) = \Delta(\boldsymbol{X}) \sum_{c\in\mathbb{C}} w^{(c)}(\mathcal{L}(\boldsymbol{X}))\, p^{(c)}(\boldsymbol{X}), \tag{222}$$

and suppose that we wish to approximate this by a GLMB of the form

$$\hat{\boldsymbol{\pi}}(\boldsymbol{X}) = \Delta(\boldsymbol{X}) \sum_{c\in\mathbb{C}} w^{(c)}(\mathcal{L}(\boldsymbol{X})) \left[p^{(c)}\right]^{\boldsymbol{X}}. \tag{223}$$

The following proposition demonstrates that there exists a simple closed form for $\hat{\boldsymbol{\pi}}$ which matches the cardinality distribution and probability hypothesis density of $\boldsymbol{\pi}$. This was established in the co-authored journal article [157][2].

Proposition 11. *For any labelled RFS density $\boldsymbol{\pi}$ of the form (222), the GLMB density which preserves the cardinality distribution and PHD of $\boldsymbol{\pi}$ is given by*

$$\hat{\boldsymbol{\pi}}(\boldsymbol{X}) = \Delta(\boldsymbol{X}) \sum_{c\in\mathbb{C}} \sum_{I\in\mathcal{F}(\mathbb{L})} \hat{w}^{(c,I)}(\mathcal{L}(\boldsymbol{X})) \left[\hat{p}^{(c,I)}\right]^{\boldsymbol{X}} \tag{224}$$

2 The work in [157] was motivated by the merged measurement GLMB, originally proposed by the author of this dissertation in [21]. However, the key result in [157] was developed in collaboration with the other co-authors. Since this new result is pertinent to the merged measurement GLMB filter, it is revisited in this section.

where

$$\hat{w}^{(c,I)}(L) = w^{(c)}(I)\,\delta_I(L), \tag{225}$$

$$\hat{p}^{(c,I)}(x,\ell) = 1_I(\ell)\,p^{(c)}_{I-\{\ell\}}(x,\ell), \tag{226}$$

$$\hat{p}^{(c)}_{\{\ell_1,\ldots,\ell_n\}}(x,\ell) = \int p^{(c)}(\{(x,\ell),(x_1,\ell_1),\ldots,(x_n,\ell_n)\})\,d(x_1,\ldots,x_n). \tag{227}$$

Proof. The cardinality distribution of $\boldsymbol{\pi}$ is given by

$$\begin{aligned}
\rho(n) &= \frac{1}{n!}\int \boldsymbol{\pi}(\{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n\})\,d(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n) \\
&= \frac{1}{n!}\sum_{(\ell_1,\ldots,\ell_n)\in\mathbb{L}^n}\int\sum_{c\in\mathbb{C}} w^{(c)}(\{\ell_1,\ldots,\ell_n\})\,p^{(c)}(\{(x_1,\ell_1),\ldots,(x_n,\ell_n)\})\,d(x_1,\ldots,x_n) \\
&= \frac{1}{n!}\sum_{(\ell_1,\ldots,\ell_n)\in\mathbb{L}^n}\sum_{c\in\mathbb{C}} w^{(c)}(\{\ell_1,\ldots,\ell_n\}) \\
&= \sum_{L\in\mathcal{F}_n(\mathbb{L})}\sum_{c\in\mathbb{C}} w^{(c)}(L).
\end{aligned} \tag{228}$$

By [198, Proposition 5], the cardinality distribution of $\hat{\boldsymbol{\pi}}$ is given by

$$\begin{aligned}
\hat{\rho}(n) &= \sum_{L\in\mathcal{F}_n(\mathbb{L})}\sum_{c\in\mathbb{C}}\sum_{I\in\mathcal{F}(\mathbb{L})}\hat{w}^{(c,I)}(L) \\
&= \sum_{L\in\mathcal{F}_n(\mathbb{L})}\sum_{c\in\mathbb{C}}\sum_{I\in\mathcal{F}(\mathbb{L})} w^{(c)}(I)\,\delta_I(L) \\
&= \sum_{L\in\mathcal{F}_n(\mathbb{L})}\sum_{c\in\mathbb{C}} w^{(c)}(L) \\
&= \rho(n)
\end{aligned} \tag{229}$$

thus the cardinality distributions of $\boldsymbol{\pi}$ and $\hat{\boldsymbol{\pi}}$ are equal. The PHD of $\boldsymbol{\pi}$ is given by

$$\begin{aligned}
v(x,\ell) &= \int \boldsymbol{\pi}(\{(x,\ell)\cup\boldsymbol{X}\})\,\delta\boldsymbol{X} \\
&= \sum_{n=0}^{\infty}\frac{1}{n!}\sum_{(\ell_1,\ldots,\ell_n)\in\mathbb{L}^n}\int\sum_{c\in\mathbb{C}} w^{(c)}(\{\ell,\ell_1,\ldots,\ell_n\}) \\
&\qquad\qquad \times p^{(c)}(\{(x,\ell),(x_1,\ell_1),\ldots,(x_n,\ell_n)\})\,d(x_1,\ldots,x_n) \\
&= \sum_{n=0}^{\infty}\frac{1}{n!}\sum_{(\ell_1,\ldots,\ell_n)\in\mathbb{L}^n}\sum_{c\in\mathbb{C}} w^{(c)}(\{\ell,\ell_1,\ldots,\ell_n\}) \\
&\qquad\qquad \times \int p^{(c)}(\{(x,\ell),(x_1,\ell_1),\ldots,(x_n,\ell_n)\})\,d(x_1,\ldots,x_n) \\
&= \sum_{n=0}^{\infty}\frac{1}{n!}\sum_{(\ell_1,\ldots,\ell_n)\in\mathbb{L}^n}\sum_{c\in\mathbb{C}} w^{(c)}(\{\ell,\ell_1,\ldots,\ell_n\})\,p^{(c)}_{\{\ell_1,\ldots,\ell_n\}}(x,\ell) \\
&= \sum_{L\subseteq\mathbb{L}}\sum_{c\in\mathbb{C}} w^{(c)}(\{\ell\}\cup L)\,p^{(c)}_L(x,\ell)
\end{aligned} \tag{230}$$

The PHD of $\hat{\boldsymbol{\pi}}$ is given by

$$\begin{aligned}
\hat{v}(x,\ell) &= \int \Delta(\{(x,\ell)\} \cup \boldsymbol{X}) \sum_{c\in\mathbb{C}} \sum_{I\in\mathcal{F}(\mathbb{L})} \hat{w}^{(c,I)}(\{\ell\} \cup \mathcal{L}(\boldsymbol{X})) \left[\hat{p}^{(c,I)}\right]^{\{(x,\ell)\}\cup\boldsymbol{X}} \delta\boldsymbol{X}. \\
&= \sum_{n=0}^{\infty} \frac{1}{n!} \sum_{(\ell_1,\ldots,\ell_n)\in\mathbb{L}^n} \sum_{c\in\mathbb{C}} \sum_{I\in\mathcal{F}(\mathbb{L})} \int \Delta(\{(x,\ell),(x_1,\ell_1),\ldots,(x_n,\ell_n)\}) \\
&\qquad \times \hat{w}^{(c,I)}(\{\ell,\ell_1,\ldots,\ell_n\}) \left[\hat{p}^{(c,I)}\right]^{\{(x,\ell),(x_1,\ell_1),\ldots,(x_n,\ell_n)\}} d(x_1,\ldots,x_n) \\
&= \sum_{n=0}^{\infty} \frac{1}{n!} \sum_{(\ell_1,\ldots,\ell_n)\in\mathbb{L}^n} \delta_{n+1}(|\{\ell,\ell_1,\ldots,\ell_n\}|) \sum_{c\in\mathbb{C}} \sum_{I\in\mathcal{F}(\mathbb{L})} \hat{w}^{(c,I)} \delta_I(\{\ell,\ell_1,\ldots,\ell_n\}) \\
&\qquad \times \int \left[\hat{p}^{(c,I)}\right]^{\{(x,\ell),(x_1,\ell_1),\ldots,(x_n,\ell_n)\}} d(x_1,\ldots,x_n) \\
&= \sum_{n=0}^{\infty} \frac{1}{n!} \sum_{(\ell_1,\ldots,\ell_n)\in\mathbb{L}^n} \delta_{n+1}(|\{\ell,\ell_1,\ldots,\ell_n\}|) \sum_{c\in\mathbb{C}} \sum_{I\in\mathcal{F}(\mathbb{L})} \hat{w}^{(c,I)} \delta_I(\{\ell,\ell_1,\ldots,\ell_n\}) \\
&\qquad \times \hat{p}^{(c,I)}(x,\ell) \int \left[\hat{p}^{(c,I)}\right]^{\{(x_1,\ell_1),\ldots,(x_n,\ell_n)\}} d(x_1,\ldots,x_n) \\
&= \sum_{n=0}^{\infty} \frac{1}{n!} \sum_{(\ell_1,\ldots,\ell_n)\in\mathbb{L}^n} \delta_{n+1}(|\{\ell,\ell_1,\ldots,\ell_n\}|) \sum_{c\in\mathbb{C}} \sum_{I\in\mathcal{F}(\mathbb{L})} \hat{w}^{(c,I)} \delta_I(\{\ell,\ell_1,\ldots,\ell_n\}) \\
&\qquad \times \hat{p}^{(c,I)}(x,\ell) \prod_{i=1}^{n} \int \hat{p}^{(c,I)}(x_i,\ell_i)\, dx_i. \qquad (231)
\end{aligned}$$

Since $\hat{p}^{(c,I)}(x_i,\ell_i)$ is a density in x_i, it integrates to unity and the product disappears. Due to the presence of $\delta_{n+1}(|\{\ell,\ell_1,\ldots,\ell_n\}|)$, and the division by $n!$, the first two sums in (231) can be simplified to a sum over subsets of $\mathbb{L}$. Thus (231) reduces to

$$\hat{v}(x,\ell) = \sum_{L\subseteq\mathbb{L}} \sum_{c\in\mathbb{C}} \sum_{I\in\mathcal{F}(\mathbb{L})} \hat{w}^{(c,I)} \delta_I(\{\ell\} \cup L)\, \hat{p}^{(c,I)}(x,\ell). \qquad (232)$$

Substituting (225) and (226) into (232) yields the result

$$\begin{aligned}
\hat{v}(x,\ell) &= \sum_{L\subseteq\mathbb{L}} \sum_{c\in\mathbb{C}} \sum_{I\in\mathcal{F}(\mathbb{L})} \hat{w}^{(c,I)}(\{\ell\} \cup L)\, \hat{p}^{(c,I)}(x,\ell) \\
&= \sum_{L\subseteq\mathbb{L}} \sum_{c\in\mathbb{C}} \sum_{I\in\mathcal{F}(\mathbb{L})} w(I)\, \delta_I(\{\ell\} \cup L)\, 1_I(\ell)\, p^{(c)}_{I-\{\ell\}}(x,\ell) \\
&= \sum_{c\in\mathbb{C}} \sum_{L\subseteq\mathbb{L}} w^{(c)}(L\cup\{\ell\})\, 1_{L\cup\{\ell\}}(\ell)\, p^{(c)}_{(L\cup\{\ell\})-\{\ell\}}(x,\ell) \\
&= \sum_{c\in\mathbb{C}} \sum_{L\subseteq\mathbb{L}} w^{(c)}(L\cup\{\ell\})\, p_L(x,\ell) \\
&= v(x,\ell), \qquad (233)
\end{aligned}$$

thus the PHDs of $\boldsymbol{\pi}$ and $\hat{\boldsymbol{\pi}}$ are equal. □

The significance of Proposition 11 lies in the fact that it facilitates the development of recursive tracking algorithms using GLMB filtering densities, and measurement likelihood models under which the GLMB is not necessarily a conjugate prior. Although the GLMB is conjugate for the standard likelihood model [198], its usefulness is limited by the fact that the standard model is unsuitable for many applications. Many higher fidelity likelihood models do not posess the separable structure necessary to maintain the GLMB form when Bayes rule is applied, hence, even if the prior is a GLMB, the exact posterior density is not. In such cases, invoking Proposition 11 allows the exact posterior to be approximated in a principled way, by maintaining the first moment and cardinality distribution of the density.

3.5 APPROXIMATE GLMB FILTERS FOR MERGED MEASUREMENTS

The exact form of the filter as specified in Propositions 9 and 10, is intractable due to the presence of joint densities $p^{(c)}(\boldsymbol{X})$ in the posterior (204). To derive an algorithm that can be implemented in practice, it is assumed that the prior is a GLMB of the form (159). The prediction step then becomes the same as the prediction of the standard GLMB filter, as specified in equations (175)-(181), since the targets are assumed to follow the standard multi-object dynamic model.

Applying the merged measurement likelihood function to a GLMB prior yields a posterior that is no longer a GLMB, but a labelled RFS mixture. However, using the result of Proposition 11, the posterior can be approximated as a GLMB with matching PHD and cardinality distribution. This allows a recursive GLMB filter for the merged measurement likelihood model to be implemented. Note that if a non-standard dynamic model was used in the prediction step, the prior may not be a GLMB. In this case, Proposition 11 could also be applied to the prior to approximate it as a GLMB before proceeding to the update step. The following proposition establishes an approximate GLMB update for the merged measurement likelihood model.

Proposition 12. *If the prior density is a GLMB of the form (159), then under the merged measurement likelihood function (202), the approximate GLMB posterior density that matches the cardinality distribution and PHD of the exact posterior density is given by*

$$\hat{\pi}(\boldsymbol{X}|Z) = \Delta(\boldsymbol{X}) \sum_{c\in\mathbb{C}} \sum_{\substack{\mathcal{U}(\boldsymbol{X})\in\mathcal{P}(\boldsymbol{X}) \\ \theta\in\Theta(\mathcal{U}(\boldsymbol{X}))}} \sum_{I\in\mathcal{F}(\mathbb{L})} \hat{w}_Z^{(c,\theta,I)}(\mathcal{L}(\boldsymbol{X});\mathcal{U}(\mathcal{L}(\boldsymbol{X}))) \times \left[\hat{p}^{(c,\theta,I)}(\cdot|Z;\mathcal{U}(\mathcal{L}(\boldsymbol{X})))\right]^{\boldsymbol{X}}, \tag{234}$$

where

$$\hat{w}_Z^{(c,\theta,I)}(L;\mathcal{U}(L)) = \frac{\delta_I(L)\, w^{(c)}(L) \left[\eta_Z^{(c,\theta)}\right]^{\mathcal{U}(L)}}{\sum_{c\in\mathbb{C}} \sum_{J\subseteq\mathbb{L}} \sum_{\substack{\mathcal{U}(J)\in\mathcal{P}(J) \\ \theta\in\Theta(\mathcal{U}(J))}} w^{(c)}(J) \left[\eta_Z^{(c,\theta)}\right]^{\mathcal{U}(J)}}, \tag{235}$$

$$\hat{p}^{(c,\theta,I)}(x,\ell|Z;\mathcal{U}(L)) = 1_I(\ell)\, p^{(c,\theta)}_{I-\{\ell\}}(x,\ell|Z;\mathcal{U}(L)), \tag{236}$$

$$p^{(c,\theta)}_{\{\ell_1,\ldots,\ell_n\}}(x,\ell|Z;\mathcal{U}(L)) = \int p^{(c,\theta)}\left(\mathcal{U}^{(\ell)}\{(x,\ell),(x_1,\ell_1),\ldots,(x_n,\ell_n)\}|Z\right) d(x_1,\ldots,x_n), \tag{237}$$

$$p^{(c,\theta)}(\boldsymbol{Y}|Z) = \frac{\check{p}^{(c)}(\boldsymbol{Y})\,\check{\psi}_Z(\boldsymbol{Y};\theta)}{\eta_Z^{(c,\theta)}(\mathcal{L}(\boldsymbol{Y}))}, \tag{238}$$

$$\mathcal{U}^{(\ell)}(\boldsymbol{Y}) = \boldsymbol{W}\in\mathcal{U}(\boldsymbol{Y}) \mid \ell\in\mathcal{L}(\boldsymbol{W}), \tag{239}$$

$$\check{p}^{(c)}(\boldsymbol{Y}) = \left[p^{(c)}\right]^{\boldsymbol{Y}}, \tag{240}$$

$$\eta_Z^{(c,\theta)}(L) = \left\langle \check{p}_L^{(c)}(\cdot), (\check{\psi}_Z)_L(\cdot;\theta)\right\rangle, \tag{241}$$

and $\check{\psi}_Z(\boldsymbol{Y};\theta)$ is given by (201).

Proof. The product of the prior distribution and the likelihood function is

$$\pi(\boldsymbol{X})\, g(Z|\boldsymbol{X}) = \Delta(\boldsymbol{X})\,\lambda \sum_{c\in\mathbb{C}} \sum_{\substack{\mathcal{U}(\boldsymbol{X})\in\mathcal{P}(\boldsymbol{X}) \\ \theta\in\Theta(\mathcal{U}(\boldsymbol{X}))}} w^{(c)}(\mathcal{L}(\boldsymbol{X})) \left[\check{\psi}_Z(\cdot;\theta)\right]^{\mathcal{U}(\boldsymbol{X})} \left[p^{(c)}\right]^{\boldsymbol{X}}. \tag{242}$$

Using equation (240),

$$\left[p^{(c)}\right]^{\boldsymbol{X}} = \prod_{\boldsymbol{Y}\in\mathcal{U}(\boldsymbol{X})} \left[p^{(c)}\right]^{\boldsymbol{Y}} = \prod_{\boldsymbol{Y}\in\mathcal{U}(\boldsymbol{X})} \check{p}^{(c)}(\boldsymbol{Y}) = \left[\check{p}^{(c)}\right]^{\mathcal{U}(\boldsymbol{X})} \tag{243}$$

hence (242) can be written as

$$\begin{aligned}\pi(\boldsymbol{X})\, g(Z|\boldsymbol{X}) &= \Delta(\boldsymbol{X})\, \lambda \sum_{c\in\mathbb{C}} \sum_{\substack{\mathcal{U}(\boldsymbol{X})\in\mathcal{P}(\boldsymbol{X}) \\ \theta\in\Theta(\mathcal{U}(\boldsymbol{X}))}} w^{(c)}(\mathcal{L}(\boldsymbol{X})) \left[\breve{p}^{(c)}(\cdot)\, \breve{\psi}_Z(\cdot;\theta)\right]^{\mathcal{U}(\boldsymbol{X})} \\ &= \Delta(\boldsymbol{X})\, \lambda \sum_{c\in\mathbb{C}} \sum_{\substack{\mathcal{U}(\boldsymbol{X})\in\mathcal{P}(\boldsymbol{X}) \\ \theta\in\Theta(\mathcal{U}(\boldsymbol{X}))}} w^{(c)}(\mathcal{L}(\boldsymbol{X})) \left[\eta_Z^{(c,\theta)}\right]^{\mathcal{U}(\mathcal{L}(\boldsymbol{X}))} \left[p^{(c,\theta)}(\cdot|Z)\right]^{\mathcal{U}(\boldsymbol{X})}\end{aligned} \tag{244}$$

and its integral

$$\begin{aligned}\int \pi(\boldsymbol{X})\, g(Z|\boldsymbol{X})\, \delta\boldsymbol{X} = \lambda \sum_{n=0}^{\infty} \frac{1}{n!} \sum_{(\ell_1,\ldots,\ell_n)\in\mathbb{L}^n} \sum_{c\in\mathbb{C}} \sum_{\substack{\mathcal{U}(\{\ell_1,\ldots,\ell_n\})\in\mathcal{P}(\{\ell_1,\ldots,\ell_n\}) \\ \theta\in\Theta(\mathcal{U}(\{\ell_1,\ldots,\ell_n\}))}} \delta_n(|\{\ell_1,\ldots,\ell_n\}|) \\ \times w^{(c)}(\{\ell_1,\ldots,\ell_n\}) \left[\eta_Z^{(c,\theta)}\right]^{\mathcal{U}(\{\ell_1,\ldots,\ell_n\})} \\ \times \int \left[p^{(c,\theta)}(\cdot|Z)\right]^{\mathcal{U}(\{(x_1,\ell_1),\ldots,(x_n,\ell_n)\})} d(x_1,\ldots,x_n).\end{aligned} \tag{245}$$

In the last line of (245), since $\mathcal{U}$ divides the set $\{(x_1,\ell_1),\ldots,(x_n,\ell_n)\}$ into disjoint subsets, each x_i only appears in one $p^{(c,\theta)}(\cdot|Z)$. Each $p^{(c,\theta)}(\cdot|Z)$ is a joint pdf, hence

$$\int \left[p^{(c,\theta)}(\cdot|Z)\right]^{\mathcal{U}(\{(x_1,\ell_1),\ldots,(x_n,\ell_n)\})} d(x_1,\ldots,x_n) = 1,$$

and the integral becomes

$$\int \pi(\boldsymbol{X})\, g(Z|\boldsymbol{X})\, \delta\boldsymbol{X} = \lambda \sum_{L\subseteq\mathbb{L}} \sum_{c\in\mathbb{C}} \sum_{\substack{\mathcal{U}(L)\in\mathcal{P}(L) \\ \theta\in\Theta(\mathcal{U}(L))}} w^{(c)}(L) \left[\eta_Z^{(c,\theta)}\right]^{\mathcal{U}(L)}. \tag{246}$$

Substituting (244) and (246) into the Bayes update equation (56) yields the following labelled RFS mixture posterior

$$\boldsymbol{\pi}(\boldsymbol{X}|Z) = \Delta(\boldsymbol{X}) \sum_{c\in\mathbb{C}} \sum_{\substack{\mathcal{U}(\boldsymbol{X})\in\mathcal{P}(\boldsymbol{X}) \\ \theta\in\Theta(\mathcal{U}(\boldsymbol{X}))}} w_Z^{(c,\theta)}(\mathcal{L}(\boldsymbol{X});\mathcal{U}(\mathcal{L}(\boldsymbol{X})))\, p^{(c,\theta)}(\boldsymbol{X}|Z;\mathcal{U}(\mathcal{L}(\boldsymbol{X})))$$

(247)

where

$$w_Z^{(c,\theta)}(L;\mathcal{U}(L)) = \frac{w^{(c)}(L) \left[\eta_Z^{(c,\theta)}\right]^{\mathcal{U}(L)}}{\sum_{c\in\mathbb{C}} \sum_{J\subseteq\mathbb{L}} \sum_{\substack{\mathcal{U}(J)\in\mathcal{P}(J) \\ \theta\in\Theta(\mathcal{U}(J))}} w^{(c)}(J) \left[\eta_Z^{(c,\theta)}\right]^{\mathcal{U}(J)}}, \tag{248}$$

$$p^{(c,\theta)}\left(\boldsymbol{X}|Z;\mathcal{U}\left(\mathcal{L}\left(\boldsymbol{X}\right)\right)\right)=\left[p^{(c,\theta)}\left(\cdot|Z\right)\right]^{\mathcal{U}(\boldsymbol{X})}. \tag{249}$$

This is not a GLMB, so to facilitate a tractable filtering recursion, Proposition 11 is applied to approximate the posterior density in GLMB form as follows,

$$\begin{aligned}\hat{\boldsymbol{\pi}}\left(\boldsymbol{X}|Z\right)=\Delta\left(\boldsymbol{X}\right)\sum_{c\in\mathbb{C}}\sum_{\substack{\mathcal{U}(\boldsymbol{X})\in\mathcal{P}(\boldsymbol{X})\\ \theta\in\Theta(\mathcal{U}(\boldsymbol{X}))}}\sum_{I\in\mathcal{F}(\mathbb{L})}\hat{w}_{Z}^{(c,\theta,I)}\left(\mathcal{L}\left(\boldsymbol{X}\right);\mathcal{U}\left(\mathcal{L}\left(\boldsymbol{X}\right)\right)\right)\\ \times\left[\hat{p}^{(c,\theta,I)}\left(\cdot|Z,\mathcal{U}\left(\mathcal{L}\left(\boldsymbol{X}\right)\right)\right)\right]^{\boldsymbol{X}}\end{aligned} \tag{250}$$

where

$$\hat{w}_{Z}^{(c,\theta,I)}\left(L;\mathcal{U}\left(L\right)\right)=w_{Z}^{(c,\theta)}\left(I;\mathcal{U}\left(I\right)\right)\delta_{I}\left(L\right), \tag{251}$$

$$\hat{p}^{(c,\theta,I)}\left(x,\ell|Z;\mathcal{U}\left(L\right)\right)=1_{I}\left(\ell\right)p_{I-\{\ell\}}^{(c,\theta)}\left(x,\ell|Z,\mathcal{U}\left(L\right)\right), \tag{252}$$

$$\begin{aligned}&p_{\{\ell_1,\ldots,\ell_n\}}^{(c,\theta)}\left(x,\ell|Z;\mathcal{U}\left(L\right)\right)\\ &\quad=\int p^{(c,\theta)}\left(\left\{\left(x,\ell\right),\left(x_1,\ell_1\right),\ldots,\left(x_n,\ell_n\right)\right\}|Z;\mathcal{U}\left(L\right)\right)d\left(x_1,\ldots,x_n\right).\end{aligned} \tag{253}$$

Substituting (248) into (251) yields the posterior component weight (235). Since (253) marginalises $p^{(c,\theta)}\left(\cdot|Z;\mathcal{U}\left(L\right)\right)$ over all states except for the one corresponding to label ℓ, only the posterior joint density of the group containing label ℓ will feature in the calculation, and all of the others will integrate to unity. Let $\mathcal{U}^{(\ell)}\left(\boldsymbol{X}\right)$ denote the set within partition $\mathcal{U}\left(\boldsymbol{X}\right)$ that contains the label ℓ. By this reasoning, (253) can be equivalently expressed as

$$\begin{aligned}&p_{\{\ell_1,\ldots,\ell_n\}}^{(c,\theta)}\left(x,\ell|Z;\mathcal{U}\left(L\right)\right)\\ &\quad=\int p^{(c,\theta)}\left(\mathcal{U}^{(\ell)}\left(\left\{\left(x,\ell\right),\left(x_1,\ell_1\right),\ldots,\left(x_n,\ell_n\right)\right\}\right)|Z\right)d\left(x_1,\ldots,x_n\right),\end{aligned} \tag{254}$$

which completes the proof of the result. □

Computing the posterior GLMB density (234) will still be very computationally demanding, since it consists of a nested sum over prior components, partitions of the target set, and measurement-to-group associations. In the following subsections, two alternative methods are proposed that can be used to avoid computing all terms in the sum, thus achieving a computationally tractable implementation of the merged measurement GLMB filter.

3.5.1 *Partitioning with Constrained Measurement Assignment*

An obvious way to compute the terms in (250) is to list all possible partitions of the target set, and then generate assignments of measurements to target groups within each partition. Clearly, this procedure will be computationally prohibitive in cases with a large number of targets, due to the exponentially increasing number of partitions. However, it is possible to remove many of the unlikely partitions by exploiting two pieces of information; the cluster structure of the target set, and a model of the sensor's resolution capabilities. Target clustering can be exploited by constructing disjoint subsets of the targets that are known to be well separated, and which have no possibility of sharing measurements. An independent filter is executed for each cluster, and since each of these filters only needs to partition a smaller set of targets, this can lead to a significant reduction in the total number of partitions to be computed [162, 114].

A method of further reducing the partitions for each cluster is to use a sensor resolution model to remove those that are unlikely to occur. Various resolution models have been proposed in the literature in order to approximate the probability of a resolution event (or equivalently, a partition of the targets) [36, 108, 184]. It is possible to use models such as these to eliminate unlikely partitions, by only retaining those that exceed some minimum probability threshold. In this work, a simpler model is used to eliminate the unlikely partitions, since our filter does not require the use of resolution event probabilities. Our model is based on two parameters; the maximum spread of targets in an unresolved group in the measurement space (d_t), and the minimum distance between centroids of unresolved groups in the measurement space (d_g). A partition $\mathcal{U}(\boldsymbol{X})$ is considered to be infeasible if either of the following conditions is true,

$$D_{min}\left(\left\{\bar{Y}; Y \in \mathcal{U}(\boldsymbol{X})\right\}\right) < d_g \tag{255}$$

$$\max\left(\left\{D_{max}(Y); Y \in \mathcal{U}(\boldsymbol{X})\right\}\right) > d_t \tag{256}$$

where $\bar{Y}$ is the mean of the set Y, $D_{min}(Y)$ is the minimum pairwise distance between points in Y, and $D_{max}(Y)$ is the maximum pairwise distance between points in Y. This model can be applied separately to each measurement dimension, and a partition is rejected if condition (255) is satisfied in all dimensions or condition (256) is satisfied

in any dimension. Intuitively, this model enforces the constraint that any unresolved target group cannot be spread over a large distance, and resolved groups cannot appear too close together. By applying these constraints, the unlikely partitions are removed, thereby allowing the filter to devote a larger proportion of its computational resources to the more likely partitions.

Having generated the feasible partitions, the terms of the posterior GLMB density can be computed. In a manner similar to that of the standard GLMB filter [198], a ranked set of alternative measurement assignments is generated using Murty's algorithm [147]. The difference here is that instead of assigning measurements to individual targets, the measurements are assigned to groups of targets within each partition. For a given partition $\mathcal{U}$, prior GLMB component c, and set of labels L, the following cost matrix is constructed

$$-\ln\left(\begin{array}{ccccccc} \phi^{(c)}(\mathcal{U}_1(L),z_1) & \cdots & \phi^{(c)}(\mathcal{U}_1(L),z_m) & \phi^{(c)}(\mathcal{U}_1(L),\varnothing) & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \phi^{(c)}(\mathcal{U}_n(L),z_1) & \cdots & \phi^{(c)}(\mathcal{U}_n(L),z_m) & 0 & \cdots & \phi^{(c)}(\mathcal{U}_n(L),\varnothing) \end{array}\right). \tag{257}$$

where

$$\phi^{(c)}(I,z) = \left\langle p^{(c)}(\cdot,I), \frac{\check{P}_D(\cdot,I)\,\check{g}(z|\cdot,I)}{\kappa(z)} \right\rangle, \tag{258}$$

$$\phi^{(c)}(I,\varnothing) = \left\langle p^{(c)}(\cdot,I), 1-\check{P}_D(\cdot,I) \right\rangle, \tag{259}$$

and $\mathcal{U}_i(L)$ denotes the i-th group of labels in partition $\mathcal{U}(L)$. Note that each row of the cost matrix corresponds to an element of $\mathcal{U}(\boldsymbol{X})$ (a target group), and each column corresponds to a measurement, or a misdetection. This matrix is passed to Murty's algorithm to generate the posterior components with highest weights, in a similar manner to the standard GLMB filter. The algorithm based on this update technique is referred to as the GLMB-MP filter. More details of the update algorithm are given in Section 3.6.1, and pseudo-code is provided in Section 3.8.2.

3.5.2 *Relaxed Measurement Assignment*

The previous section described a method of generating highly weighted components from the posterior, however, it is very computationally demanding due to the need to

compute ranked assignments for every feasible partition of every component in the GLMB prior. A cheaper but approximate method of generating the posterior GLMB components is now proposed, which can potentially alleviate this bottleneck.

Let us neglect for a moment the fact that measurements are generated by groups of targets, and instead, assume that targets give rise to measurements independently, but more than one target may generate the same measurement. Although this seems like a contradiction, the effect of this is mitigated by the fact the component weight calculation is still carried out exactly, so in principle it can achieve similar performance to the partitioned method, provided that enough components are generated.

The likelihood of assigning a measurement to a target is based only on that target's prior, which is captured in a directed graph as shown in Figure 16. In the graph structure, each row corresponds to a target, the first column corresponds to the target being misdetected, and the remaining columns correspond to individual measurement assignments. A start node is placed before the first row, and an end node after the last row. Using this graph, a ranked list of feasible assignments can be obtained using a k-shortest paths algorithm [206]. Note that a given column may be visited several times within a path (in which case the measurement corresponding to that column is assigned to more than one target), but each target can only be assigned a single measurement. Each generated path can be viewed as a one-to-one mapping of measurements to target groups, with the difference being that the grouping is not pre-specified, but arises naturally due to the association of multiple targets to a single measurement.

It is important to note that due to our target independence assumption, the solutions generated by this method will not be in decreasing order of posterior weight. However, under common conditions, this method can be expected to generate components with significant weights, with the advantage that it is computationally much cheaper than the partitioning approach. The filter using this update procedure is referred to as the GLMB-MR filter.

3.6 IMPLEMENTATION OF GLMB FILTERS FOR MERGED MEASUREMENTS

This section describes the measurement update procedures for both the GLMB-MP and GLMB-MR filters introduced in the previous section. For pseudo-code of the

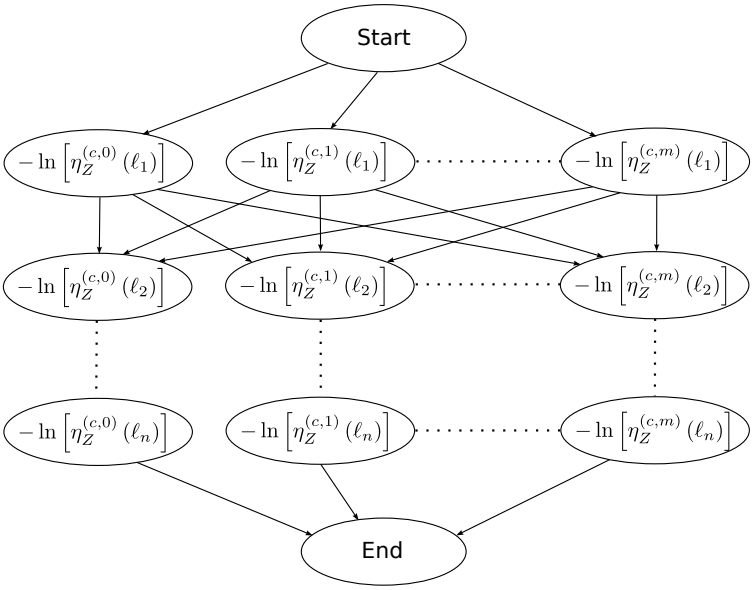


Figure 16: Graph structure used for the relaxed measurement assignment in the GLMB-MR filter. A k-shortest paths algorithm is used to generate alternative assignments of measurements to groups of targets.

measurement update procedures described here, refer to Section 3.8 at the end of this chapter.

3.6.1 *GLMB-MP Measurement Update*

The update routine for the GLMB-MP filter involves generating a set of posterior components for each component in the prior, by enumerating the feasible partitions and measurement-to-group assignments. Note that this procedure can be easily parallelised, since there is no dependency between prior components. The first step is to decide the number of posterior components to generate for each prior component. This is done by first allocating a number of components $N_{cdn}^{(n)}$ to each cardinality n, based on a Poisson distribution centered around the mean of the prior cardinality distribution. This avoids component depletion by ensuring that components are generated for a variety of cardinalities, which is necessary to allow for target births and deaths, even if the predicted cardinality distribution is sharply peaked.

For each cardinality, the number allocated $N_{cdn}^{(n)}$ is divided among the components of cardinality n, where the number $N_{cmp}^{(i)}$ allocated to the i-th component is proportional to its prior weight. Since it is usually not possible to do this exactly for an integer numbers of components, this is carried out by a process of randomised al-

location, where the probability of a posterior component being allocated to a prior component is proportional to the prior component's weight.

For each component, the feasible partitions of the target set are computed using the constraints (255) and (256). The number of allocated posterior components $N_{cmp}^{(i)}$ is then divided equally among these partitions, where the number allocated to the $k-th$ partition is denoted $N_{prt}^{(k)}$. An assignment cost matrix is then computed for each feasible partition according to (257), which is used to obtain a list of the cheapest $N_{prt}^{(k)}$ measurement-to-group assignments in increasing order of cost. Murty's ranked assignment algorithm [147] is used for this purpose, which involves solving a sequence of optimal assignment problems. This can be carried out using a number of well known techniques, including the Hungarian algorithm [113], the Jonker-Volgenant algorithm [95], or the auction algorithm [27]. The implementation used in this study makes use of the auction algorithm.

For each assignment generated by Murty's algorithm, a new posterior component is created, consisting of one updated joint density for each target group according to (218), (241) and (201). Finally, the posterior density is approximated in the form of a GLMB by computing the marginal distribution of each target within each posterior component, in accordance with Proposition 11.

Pseudo-code for the GLMB-MP measurement update procedure described here is provided in Algorithm 20 at the end of this chapter.

3.6.2 *GLMB-MR Measurement Update*

As in the partitioned update procedure, the GLMB-MR generates a set of posterior components for each component in the prior. However, in the GLMB-MR filter, a different method of choosing the components is employed, leading to a cheaper algorithm which is scalable to a larger number of targets per cluster. Here, the enumeration of target partitions and Murty-based ranked assignment, is replaced with a relaxed assignment of measurements to targets. This means that a single measurement may be assigned to multiple targets simultaneously, but each target can still only receive one measurement. This leads to a simpler and computationally cheaper algorithm, but as one would expect, the performance is not as good as the partitioning approach.

For each prior component, the graph in Figure 16 is generated, which consists of one row for each target, and one column for each measurement, with one extra column for misdetections. The k-shortest paths algorithm is applied to this graph to select components for inclusion in the posterior GLMB. The component weights are computed in the same way as in the partitioned update. However, instead of performing joint updates for each group followed by marginalisation, the single target prior distributions are updated independently, and used as an approximation to the marginal posterior densities. This avoids the computational bottleneck of performing a large number of joint multi-target updates (one for every possible target grouping). Instead, only a single independent update needs to be performed for each unique single-target density contained in the prior GLMB.

Pseudo-code for the GLMB-MR measurement update procedure is provided in Algorithm 21 at the end of this chapter.

3.7 SIMULATION RESULTS

This section provides performance results for the two proposed versions of the GLMB-M filter on a simulated multi-target tracking scenario. The sensor generates noisy bearings-only measurements with false alarms and misdetections, and there are multiple crossing targets, each of which follows a white noise acceleration dynamic model. The sensor's motion comprises legs of constant velocity, interspersed with constant turn rate manoeuvres. These manoeuvres improve the observability as the simulation progresses.

The target state space is defined in terms of 2D Cartesian position and velocity vectors, i.e. for target t at time k, $\tilde{x}_k^{(t)} = \begin{bmatrix} x_k^{(t)} & y_k^{(t)} & \dot{x}_k^{(t)} & \dot{y}_k^{(t)} \end{bmatrix}^T$. All targets follow the dynamic model

$$\tilde{x}_{k+1}^{(t)} = F\tilde{x}_k^{(t)} + \Gamma v_k \tag{260}$$

$$F = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \otimes I_2, \quad \Gamma = \begin{bmatrix} T^2/2 \\ T \end{bmatrix} \otimes I_2 \tag{261}$$

where $T = 20s$ is the sensor sampling period, and $v_k \sim \mathcal{N}(0, Q)$ is a 2×1 independent and identically distributed Gaussian process noise vector with $Q = \sigma_v^2 I_2$ where the standard deviation of the target acceleration is $\sigma_v = 0.005 m/s^2$.

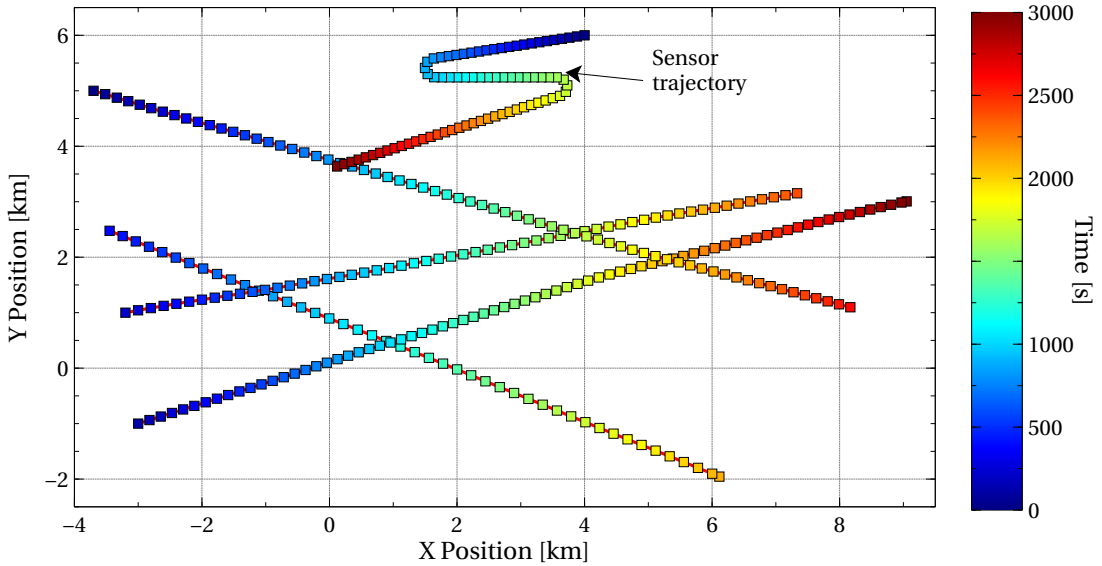


Figure 17: Target-observer geometry for the GLMB-M test scenario.

The measurements are simulated using two alternative methods; one based on a detection-level measurement model, and the other based on an image simulation followed by peak detection. The details of these models are given in the subsections to follow, and in both cases, the measurement function is given by

$$h\left(\tilde{x}_k^{(t)}, \tilde{x}_k^{(s)}\right) = \arctan\left(\frac{x_k^{(t)} - x_k^{(s)}}{y_k^{(t)} - x_k^{(s)}}\right). \tag{262}$$

where $\tilde{x}_k^{(t)}$ is the target state vector, and $\tilde{x}_k^{(s)}$ is the sensor state vector. For this analysis, the pdf of each target is modelled using a single Gaussian, and the measurement updates are carried out using the extended Kalman filter (EKF). It is clearly possible to use more sophisticated non-linear filters, such as the unscented or cubature Kalman filter, or to model the single target pdfs using more accurate representations such as Gaussian mixtures or particle distributions. These techniques may lead to some performance improvement, but this analysis is beyond the scope of this work.

The scenario consists of four targets, with geometry as shown in Figure 17. One target is present at the beginning, with another three arriving during the first 400 seconds, and three terminating during the final 900 seconds. The bearings of all four targets cross one another in the middle of the scenario, during which time, merged measurements will be present in the data. This crossing can be seen around time 1500s in the bearing plot in Figure 18.

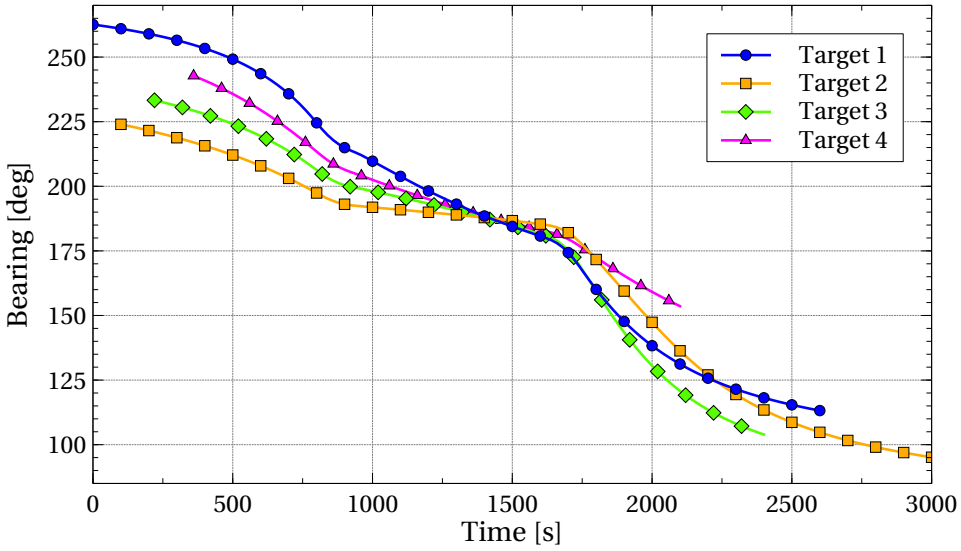


Figure 18: True target bearings for the GLMB-M test scenario.

In all cases, the filter initialises the pdf of each newborn target as a Gaussian with mean range of 10km from the sensor, range standard deviation of 3km, velocity of 5m/s moving directly towards the sensor along the line of bearing, course standard deviation of 50°, and speed standard deviation of 2m/s. The filters generate a maximum of $N_u = 10000$ components in the GLMB density during the update, which is reduced to $N_r = 1000$ (with $N_c = 10$, and $R_m = 10^8$) during the pruning step. All filters were implemented in C++, and executed on an Intel Core i7 2600 processor. The implementation used here does not include parallelisation of the component generation process, and it would be possible to significantly speed up the execution time by doing so.

3.7.1 *Detection-level Simulation Model with Merging*

Let $\mathbb{X}$ be the single-object state space, and $\mathbb{Z}$ the sensor measurement space. At time index k, consider a single sensor with known control input u_k, and let $\boldsymbol{X}_k$ represent the unknown multi-target state. Let $h(\cdot, u_k) : \mathbb{X} \times \mathbb{L} \to \mathbb{Z}$ be the measurement function, $C = \{c_1, c_2, ...c_{N_{cell}}\}$ be a set of disjoint cells covering $\mathbb{Z}$ such that $\bigcup_{i=1}^{N_{cell}} c_i = \mathbb{Z}$ and $c_i \cap c_j = \emptyset, \forall i \neq j$, and let $\boldsymbol{T}_k^{(i)}$ be the set of targets whose true state falls in cell i at time k,

$$\boldsymbol{T}_k^{(i)}(\boldsymbol{X}) = \{\boldsymbol{x} \in \boldsymbol{X} : h(\boldsymbol{x}, u_k) \in c_i\}. \tag{263}$$

Table 6: Measurement noise used in the GLMB-M test scenario.

Cell Width	σ_w		
	1 Target	2 Targets	>2 Targets
1°	0.25°	0.375°	0.5°
2°	0.5°	0.75°	1°
4°	1°	1.5°	2°

Then cell i produces the following target measurement

$$Z_k^{(i)} = \begin{cases} \frac{1}{\left|\boldsymbol{T}_k^{(i)}(\boldsymbol{X})\right|} \sum_{\boldsymbol{x}\in\boldsymbol{T}_k^{(i)}} h(\boldsymbol{x}, u_k) + w_k, & \left|\boldsymbol{T}_k^{(i)}(\boldsymbol{X})\right| > 0 \\ \emptyset, & \left|\boldsymbol{T}_k^{(i)}(\boldsymbol{X})\right| = 0 \end{cases} \tag{264}$$

with probability $P_D\left(\boldsymbol{T}_k^{(i)}(\boldsymbol{X})\right)$, and $Z_k^{(i)} = \emptyset$ with probability $1 - P_D\left(\boldsymbol{T}_k^{(i)}(\boldsymbol{X})\right)$. In the above, w_k is an independent measurement noise vector. The set of target-generated measurements at time k is thus $D_k = \bigcup_{i=1}^{N_{cell}} Z_k^{(i)}$. In addition, at each time k, the sensor generates an RFS of false measurements denoted K_k, with a Poisson cardinality distribution of known mean. The spatial locations of the points in K_k are simulated by repeatedly generating a random point within a randomly chosen unoccupied cell, such that any given cell cannot contain more than one measurement. The overall measurement set is thus given by $Z_k = D_k \cup K_k$.

This model allows us to explicitly control the measurement noise, clutter rate, and resolution, such that the filter parameters can be matched to their true values. The algorithms are tested under three bearing cell widths; 1°, 2° and 4°. The measurement noise is dependent on the number of unresolved targets in the group, and is set according to Table 6.

In these simulations, the detection probability is 0.98, and the mean number of clutter measurements is 30. The OSPA distance is used to analyse the relative performance of the algorithms, with a cutoff of 1km, and order parameter of 2. Figures 19, 20 and 21 show the average OSPA distance and algorithm execution times (over 200 Monte Carlo runs) for the standard GLMB, GLMB-MP, and GLMB-MR filters for the three different cell widths. Also shown are the results of running the standard GLMB filter on measurements with perfect resolution and the same noise level as fully resolved targets in each case.

As one would expect, the standard GLMB filter performs poorly when processing the finite resolution measurements, since it cannot account for measurement merging. When the target bearings begin to cross, a significant jump in OSPA can be observed for the standard GLMB. The reason for this is clear, the filter can only explain the missing measurements as misdetections or target deaths. Since the probability of detection is high, the probability of having several misdetections in a row is very small. Hence the filter believes that targets have died and terminates the tracks, which is also the reason for the associated drop in execution time. When the targets become resolved again, the filter must initiate new tracks (with default prior state), since it has no way of continuing a track that has been previously terminated. For this reason, the standard filter will take a long time to recover from dropped tracks, especially in situations with low observability.

Also as expected, the GLMB-MP filter performs best in all three cases, because by enumerating the partitions of the target set in advance, it is able to obtain better posterior components than the GLMB-MR filter. The drawback of this algorithm is that when the targets get close to each other, the execution time rises dramatically due to the increasing number of feasible partitions. As the cell width increases, the width of the peak in the execution time also increases, because the targets are unresolved for a longer period.

The GLMB-MR filter clearly has higher error, however, it does not suffer from spikes in execution time. In all three cases, the peak execution time of the GLMB-MR is less than 20% of that for the GLMB-MP. This is due to the fact that the latter uses a more expensive ranked assignment algorithm, and needs to carry out many group measurement updates for different partitions of the target set, each of which involves higher dimensional matrices. In contrast, the GLMB-MR only performs a single measurement update for each target, and a cheaper assignment based on k-shortest paths. Hence the GLMB-MR has a relatively consistent run time, similar to that of the standard GLMB filter.

3.7.2 *Image-based Simulation Model*

The simulation model in the previous section allowed us to explicitly control the clutter rate and measurement noise parameters via simulation at the detection level. In this section, the merged measurement filters are applied to data generated through a

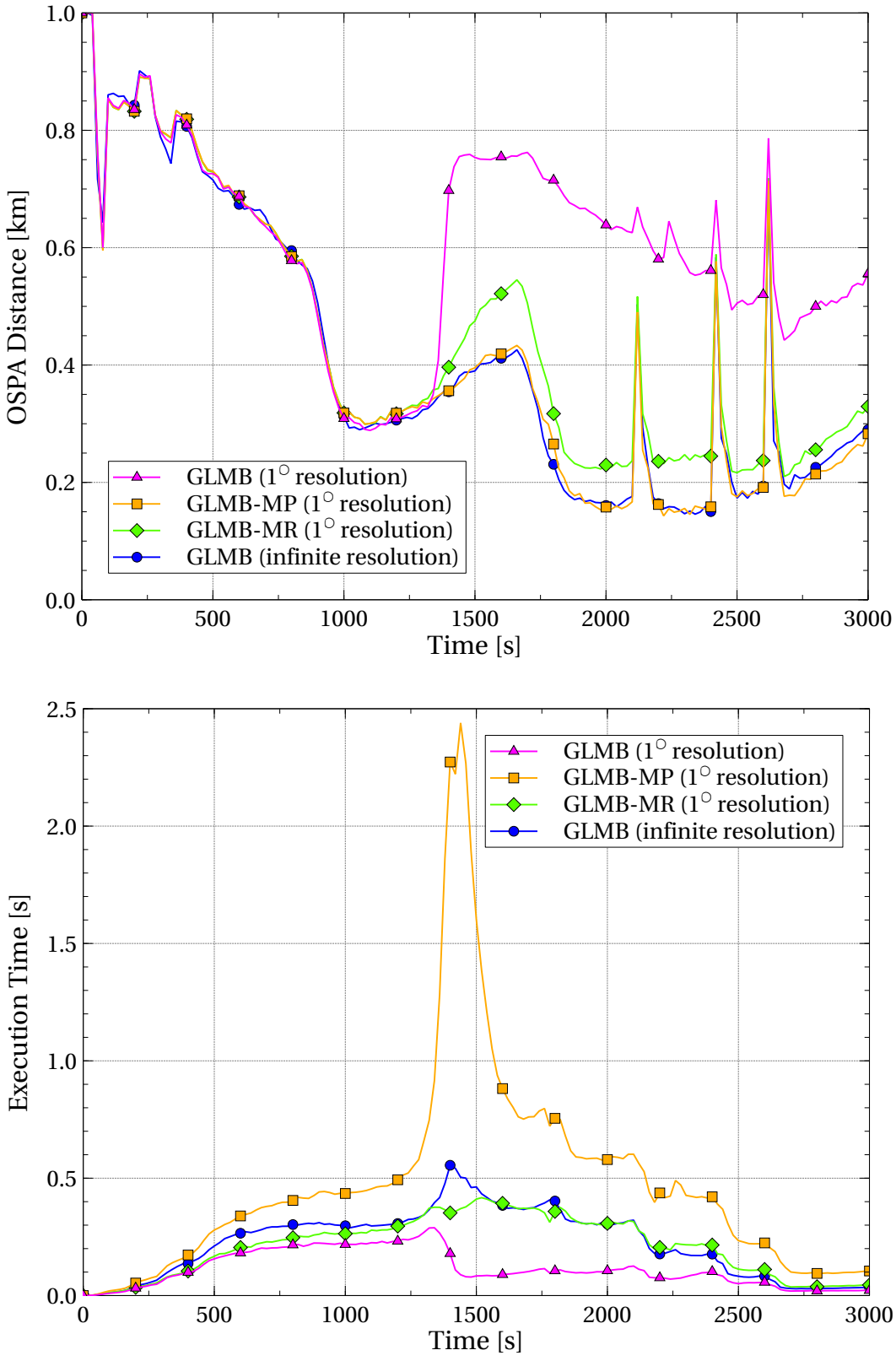


Figure 19: OSPA distance and execution time for GLMB/GLMB-M filters in detection-level simulation with 1° resolution.

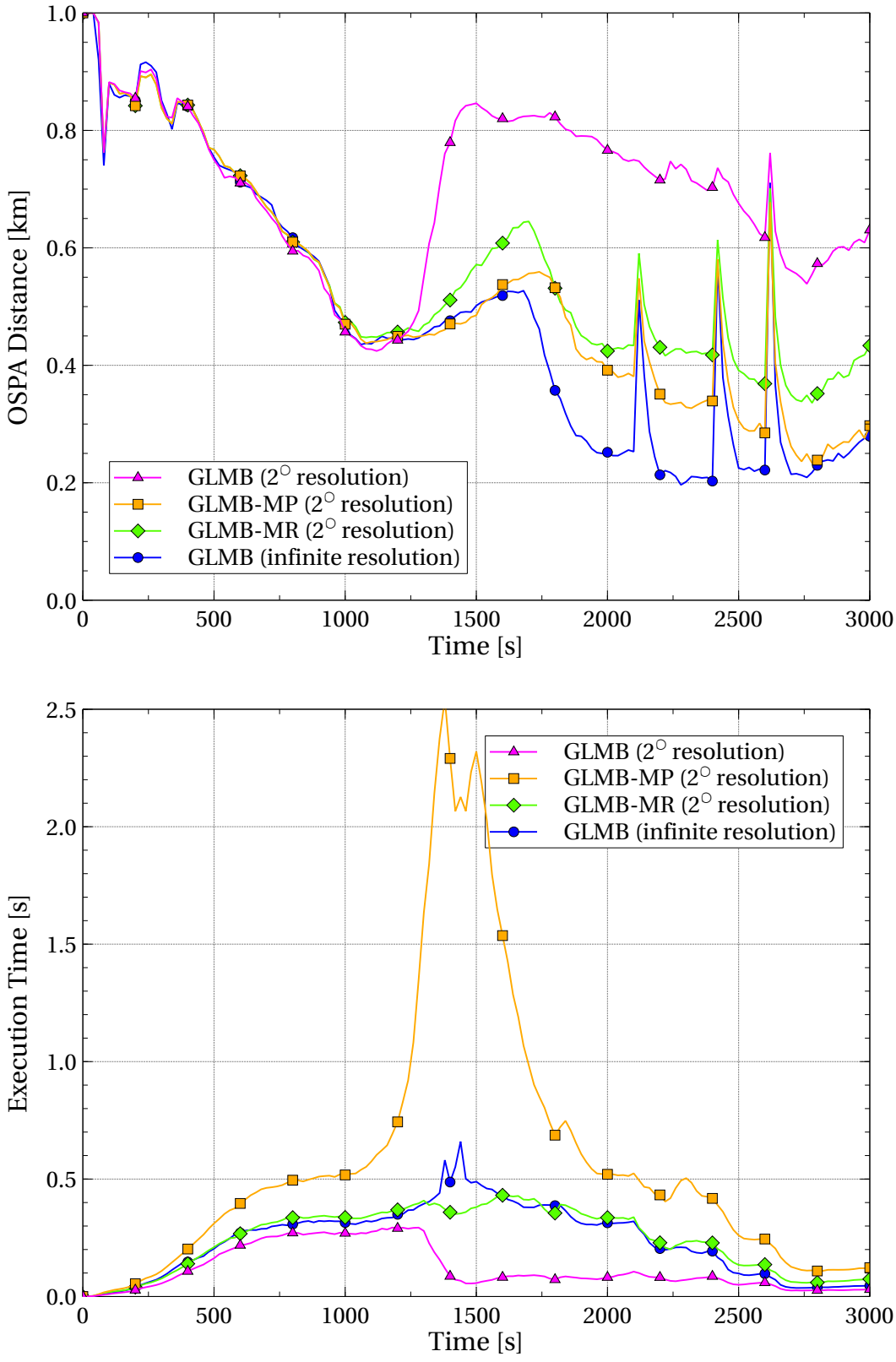


Figure 20: OSPA distance and execution time for GLMB/GLMB-M filters in detection-level simulation with 2° resolution.

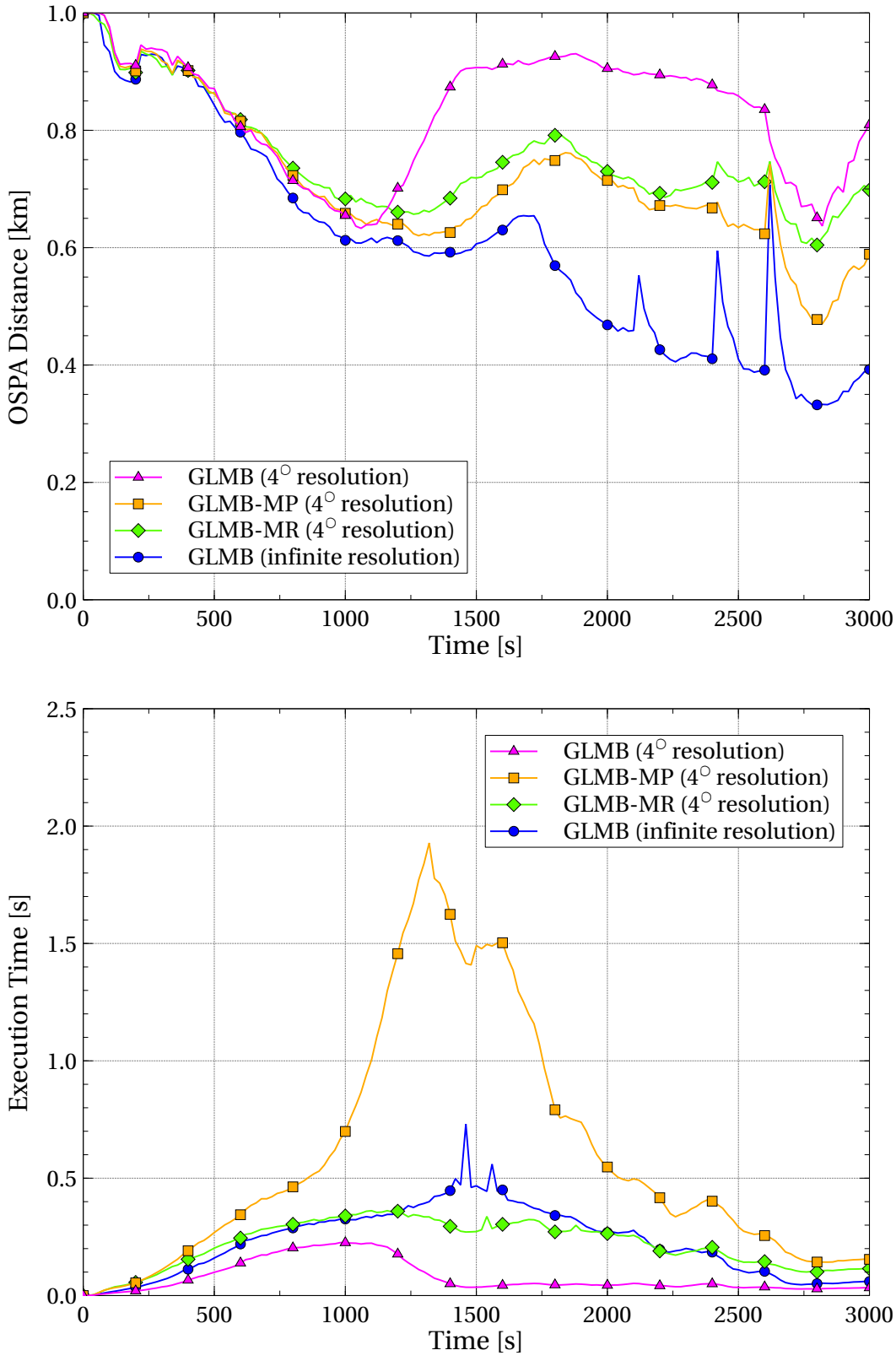


Figure 21: OSPA distance and execution time for GLMB/GLMB-M filters in detection-level simulation with 4° resolution.

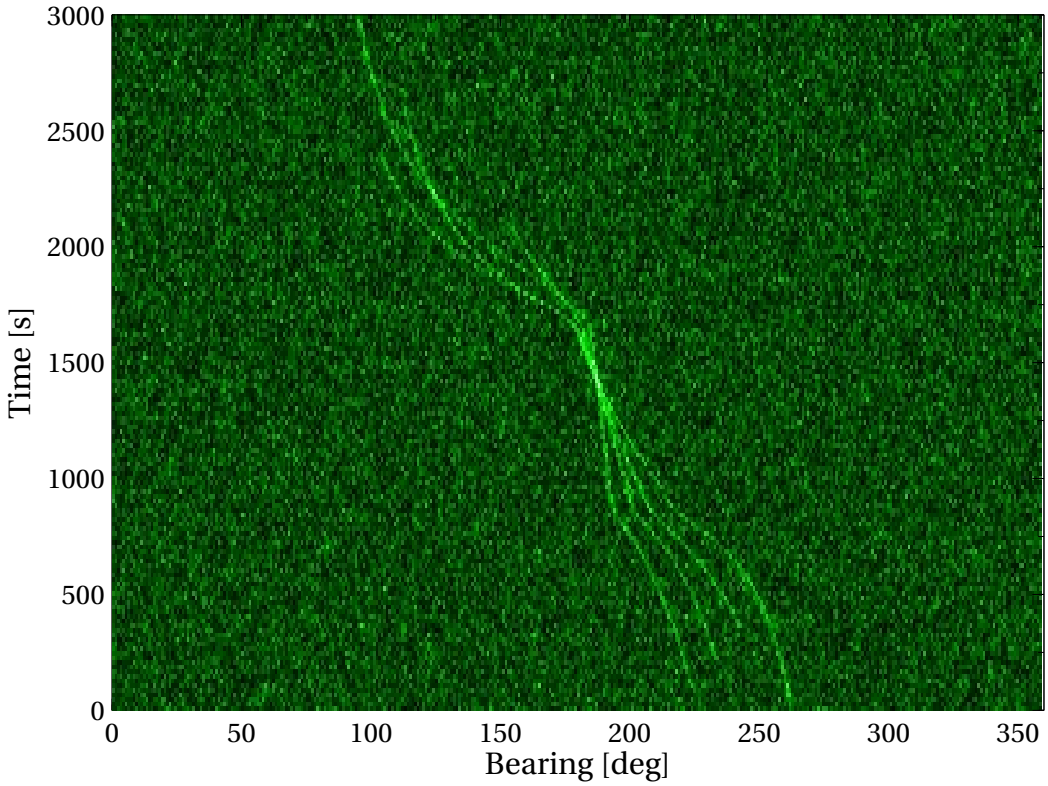


Figure 22: Simulated time-bearing image with 1° cell width.

more realistic simulation model, in which a sensor image is generated, and a simple peak detector used to extract the point measurements. The image is simulated by first generating Rayleigh distributed background noise in each cell. A point spread function is applied to each target to obtain the mean target-generated signal amplitude in each cell, and a Ricean distributed amplitude is then generated for every cell. This is repeated for all targets, and the signal amplitudes added together in each cell. To generate detections, the vector of cell amplitudes is interpolated using a spline function, and the locations of all peaks that exceed a threshold are extracted. Under this model, the combined effect of the cell width and the point spread function determine the distance at which targets become unresolvable. Figure 22 shows an example of the image generated from this model, with cell width of 1°, average signal to noise ratio of 5dB, and a Gaussian point spread function with standard deviation of 0.5°.

Figures 23 and 24 show the average OSPA distance over 200 runs for the standard GLMB, GLMB-MP and GLMB-MR filters, for the case of 1° and 2° cell widths. These results follow the same trend that was observed under the detection-level simulation model. The standard GLMB filter fails due to dropped tracks, and the GLMB-MP generally outperforms the GLMB-MR, albeit with a higher computational cost.

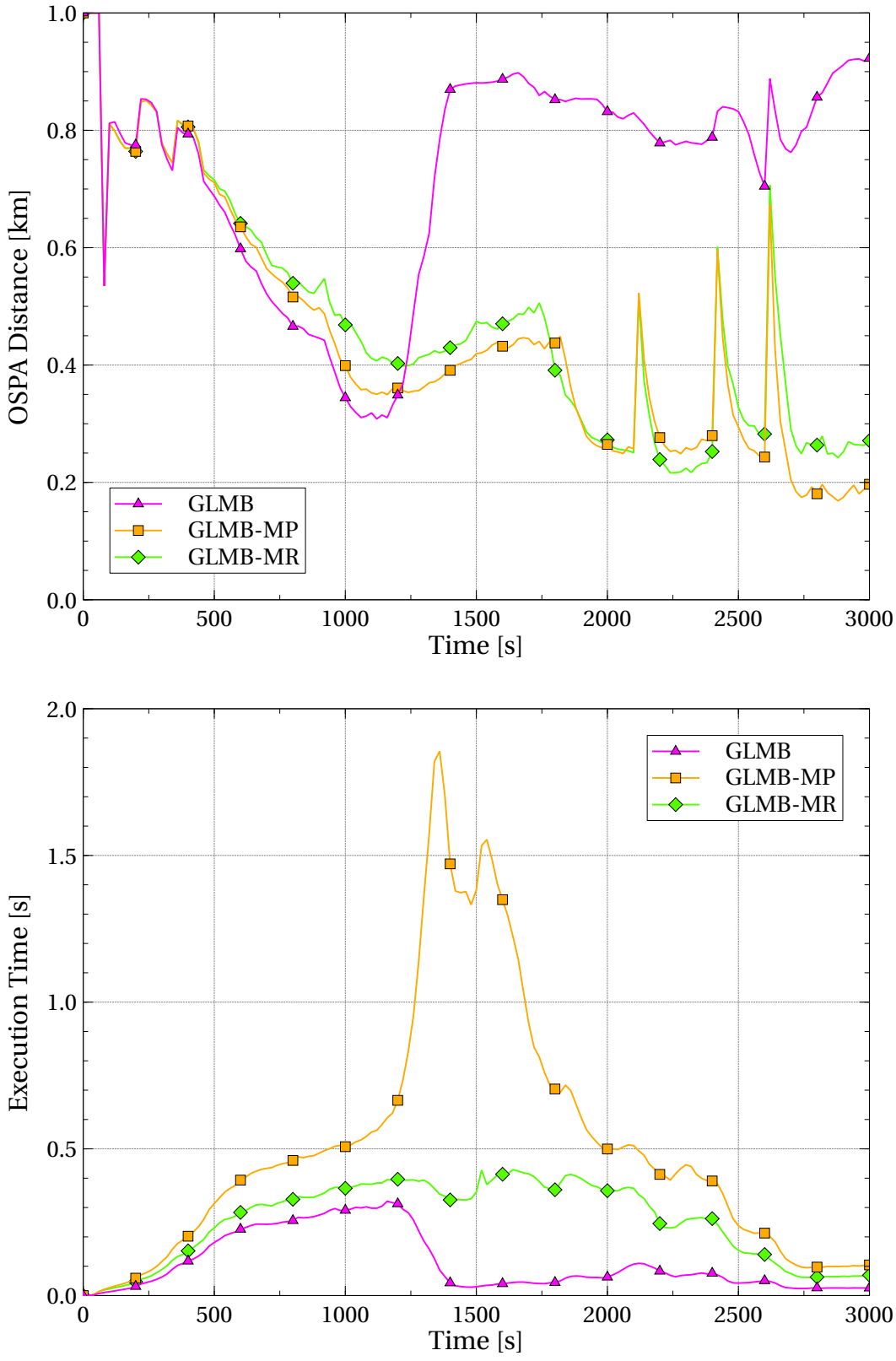


Figure 23: OSPA distance and execution time for GLMB/GLMB-M filters in image-based simulation with 1° cell width.

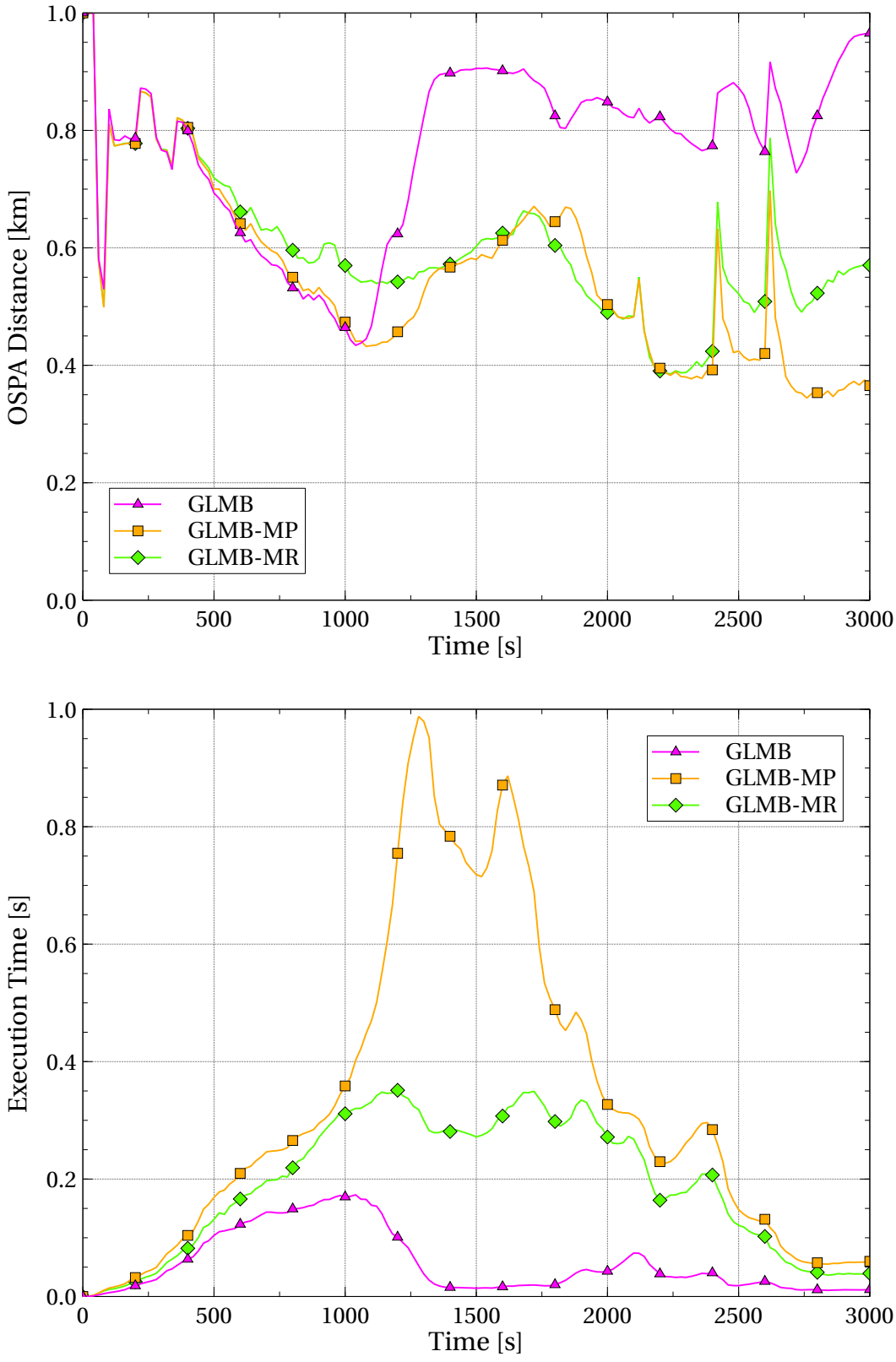


Figure 24: OSPA distance and execution time for GLMB/GLMB-M filters in image-based simulation with 2° cell width.

3.8 PSEUDO-CODE FOR MERGED MEASUREMENT GLMB FILTERS

3.8.1 *Merged Measurement GLMB Recursion*

Pseudo-code for the top-level recursion of the GLMB-M filter is provided in Algorithm 19. Note that the birth density generation, survival prediction, estimate extraction and pruning are the same as those in the standard GLMB filter of Algorithm 3. The only difference is in the update procedure, which can be either the GLMB-MP of GLMB-MR update, as dictated by the situation and the available computational resources.

Algorithm 19: GLMB filter for merged measurements.

Input: Measurements $(Z_{1:N})$, measurement times $(t_{1:N})$
Output: Track estimates (τ_x), track labels (τ_ℓ)

Function $[\tau_x, \tau_\ell] = \texttt{MergedMeasGLMB\_Filter}(Z_{1:N}, t_{1:N})$ **is**
  $\ell_{next} = 1$;
  $\tau_x = \emptyset$;
  $\tau_\ell = \emptyset$;
  $P_{A,1} = \texttt{Zeros}(1, |Z_1|)$;
  for $k = 2 : N$ **do**
    $[\Omega_{B,k}, \ell_{next}] = \texttt{BirthGLMB}(Z_{k-1}, P_{A,k-1}, t_{k-1}, Z_k, t_k, \ell_{next})$;
    if $k = 2$ **then**
      $\Omega_+ = \Omega_{B,k}$;
    else
      $\Omega_{S,k} = \texttt{PredictSurvivalGLMB}(\Omega_{k-1}, t_k)$;
      $\Omega_+ = \texttt{MultiplyGLMB}(\Omega_{S,k}, \Omega_{B,k})$;
    if *GLMB-MP* **then**
      $[\Omega_k, P_{A,k}] = \texttt{PartitionedUpdateGLMB}(\Omega_+, Z_k)$;
    else if *GLMB-MR* **then**
      $[\Omega_k, P_{A,k}] = \texttt{RelaxedUpdateGLMB}(\Omega_+, Z_k)$;
    $[\tau_x, \tau_\ell] = \texttt{ExtractEstimatesGLMB}(\Omega_k, \tau_x, \tau_\ell)$;
    $\Omega_k = \texttt{PruneGLMB}(\Omega_k)$;

3.8.2 *GLMB-MP Update Procedure*

The measurement update procedure for the GLMB-MP filter is specified by Algorithm 20. This algorithm makes use of some base functions, which are described in Table 7.

Algorithm 20: GLMB update with target set partitioning (GLMB-MP).

Input: GLMB density (Φ), measurements (Z)

Parameters: Number of updated components to generate (N_u), detection probability (P_D), clutter intensity (λ), volume of surveillance region (V), minimum spacing between group centroids (d_g), maximum target group spread (d_t)

Output: Updated GLMB density (Ω), probability that each measurement is from a target (P_A)

Function $[\Omega, P_A] = \texttt{PartitionedUpdateGLMB}(\Phi, Z)$ **is**
  for $i = 1 : |Z|$ **do**
    $\theta_i = \emptyset$;
  $N_{cdn} = \texttt{Allocate}\left(N_u, \texttt{Poisson}\left(0 : \Phi.M, \texttt{Mean}\left(\Phi.\rho^{(:)}\right)\right)\right)$;
  for $n = 1 : \Omega.M$ **do**
    $j = 1$;
    $N_{cmp} = \texttt{Allocate}\left(N_{cdn}^{(n)}, \Phi.w^{(n,:)}\right)$;
    for $i = 1 : \Phi.N_n$ **do**
      $\mathcal{P} = \texttt{Partition}\left(\Phi.p^{(n,i,:)}\right)$;
      $\mathcal{P} = \texttt{ConstrainPartitions}\left(\mathcal{P}, d_g, d_t\right)$;
      $N_{prt} = \texttt{Allocate}\left(N_{cmp}^{(i)}, \texttt{Ones}(1, |\mathcal{P}|)/|\mathcal{P}|\right)$;
      for $k = 1 : |\mathcal{P}|$ **do**
        $\mathcal{U} = \mathcal{P}(k)$;
        $C = \texttt{CostMatrixGrouped}\left(Z, \Phi.p^{(n,i,:)}, \mathcal{U}, P_D\right)$;
        $A = \texttt{Murty}\left(C, N_{prt}^{(k)}\right)$;
        for $(a, c) \in A$ **do**
          $t = 1$;
          $\Omega.w^{(n,j)} = \Phi.w^{(n,i)} \times \Phi.\rho^{(n)} \times \lambda^{|Z|} \times V^{-1} \times e^{-c}$;
          for $\mathcal{G} \in \mathcal{U}$ **do**
            $p_g = \texttt{JointDensity}\left(\Phi.p^{(n,i,\mathcal{G})}\right)$;
            $L_g = \left[\Phi.\ell^{(n,i,\mathcal{G})}\right]$;
            if $a(\mathcal{G}) > 0$ **then**
              $p_g = \texttt{JointTargetUpdate}\left(p_g, z_{a(\mathcal{G})}\right)$;
              $\theta_{a(\mathcal{G})} = \left[\theta_{a(\mathcal{G})}; (n, j)\right]$;
            for $k = 1 : |\mathcal{G}|$ **do**
              $\Omega.p^{(n,j,t)} = \texttt{Marginal}\left(p_g, k\right)$;
              $\Omega.\ell^{(n,j,t)} = L_g(k)$;
              $t = t + 1$;
          $j = j + 1$;
  $\Omega = \texttt{NormaliseGLMB}(\Omega)$;
  for $i = 1 : |Z|$ **do**
    $P_A(i) = \texttt{Sum}\left(\Omega.w^{(\theta_i)}\right)$;

Table 7: Functions used in GLMB-MP pseudo code.

Function Name	Description
`Partition` $(\boldsymbol{Y})$	List partitions of the set $\boldsymbol{Y}$.
`ConstrainPartitions` $(\mathcal{P}, d_g, d_t)$	Remove infeasible partitions in $\mathcal{P}$, according to (255)-(256).
`CostMatrixGrouped` $(Z, \boldsymbol{X}, \mathcal{U}, P_D)$	Compute cost matrix (257) for measurement set Z and target set $\boldsymbol{X}$ grouped according to partition $\mathcal{U}$, with detection probability function P_D.
`JointDensity` (P)	Join the pdfs in the set P to create a joint density
`JointTargetUpdate` (p, z)	Update the joint prior target pdf p with measurement z
`Marginal` (p, i)	Compute the marginal distribution of the i-th target from the joint density p

3.8.3 *GLMB-MR Update Procedure*

Algorithm 21 summarises the measurement update procedure of the GLMB-MR filter, which in addition to the base functions already defined in Tables 3 and 7, also makes use of the two functions described in Table 8.

Table 8: Functions used in GLMB-MR pseudo code.

Function Name	Description
`CostGraph` $(Z, \boldsymbol{X})$	Compute cost graph in Figure 16 for measurement set Z and target set $\boldsymbol{X}$.
`Unique` (y)	List the unique elements of the array y.

Algorithm 21: GLMB update with relaxed assignment (GLMB-MR).

Input: Predicted GLMB density (Φ), measurements (Z)

Parameters: Number of updated components to generate (N_u), detection probability (P_D), clutter intensity (λ), volume of surveillance region (V)

Output: Updated GLMB density (Ω), probability that each measurement is from a target (P_A)

Function $[\Omega, P_A] = \texttt{RelaxedUpdateGLMB}(\Phi, Z)$ **is**
  for $i = 1 : |Z|$ **do**
    $\theta_i = \emptyset$;
  $N_{cdn} = \texttt{Allocate}\left(N_u, \texttt{Poisson}\left(0 : \Phi.M, \texttt{Mean}\left(\Phi.\rho^{(:)}\right)\right)\right)$;
  for $n = 1 : \Phi.M$ **do**
    $j = 1$;
    $N_{cmp} = \texttt{Allocate}\left(N_{cdn}^{(n)}, \Phi.w^{(n,:)}\right)$;
    for $i = 1 : \Phi.N_n$ **do**
      $G = \texttt{CostGraph}\left(Z, \Phi.p^{(n,i,:)}\right)$;
      $A = \texttt{ShortestPaths}\left(G, N_{cmp}(i)\right)$;
      for $(a, c) \in A$ **do**
        $t = 1$;
        $\Omega.w^{(n,j)} = \Phi.w^{(n,i)} \times \Phi.\rho^{(n)} \times \lambda^{|Z|} \times V^{-1}$;
        for $m \in \texttt{Unique}(a)$ **do**
          $\mathcal{G} = \{k : a(k) = m\}$;
          $p_g = \texttt{JointDensity}\left(\Phi.p^{(n,i,\mathcal{G})}\right)$;
          if $m > 0$ **then**
            $\Omega.w^{(n,j)} = \Omega.w^{(n,j)} \times P_D(Y) \times g(\boldsymbol{z}_m|Y)$;
            $\theta_m = [\theta_m; (n, j)]$;
          else
            $\Omega.w^{(n,j)} = \Omega.w^{(n,j)} \times (1 - P_D(Y))$;
          for $k = 1 : |\mathcal{G}|$ **do**
            $\Omega.p^{(n,j,t)} = \texttt{TargetUpdate}\left(\Phi.p^{(n,i,\mathcal{G}(k))}, z_m\right)$;
            $\Omega.\ell^{(n,j,t)} = \Phi.\ell^{(n,i,\mathcal{G}(k))}$;
            $t = t + 1$;
        $j = j + 1$;
  $\Omega = \texttt{NormaliseGLMB}(\Omega)$;
  for $i = 1 : |Z|$ **do**
    $P_A(i) = \texttt{Sum}\left(\Omega.w^{(\theta_i)}\right)$;

4 TRACKING MULTIPLE EXTENDED TARGETS

Targets that generate multiple measurements at a given instant in time are commonly known as extended targets. These present a challenge for many tracking algorithms, as they violate a key assumption of the standard measurement model, specifically, that a target may generate at most one measurement at any given time. Extended targets are an important consideration in many scenarios, in particular when the targets are large relative to the sensor resolution.

The contribution of this chapter is the first multi-target tracker for extended targets based on labelled random finite sets. The presented algorithm is capable of tracking an unknown and time-varying number of extended targets in clutter. The full trajectory of each target is estimated, comprising the kinematics, measurement rates and extents. The results of this chapter have been presented in the author's journal article [24][1], and conference paper [20].

Section 4.1 contains a brief introduction on extended target tracking. In Section 4.2, an RFS-based measurement likelihood model for multiple extended targets in clutter is developed. The state space model for a single extended target is specified in Section 4.3, which forms the basis of the proposed GGIW-GLMB and GGIW-LMB filters for extended targets described in Sections 4.4 and 4.5. Section 4.6 presents more details relating to the implementation of these algorithms, and their performance is demonstrated in Sections 4.7 and 4.8, with applications to simulated and real-world measurement data. Finally, pseudo-code for the GGIW-GLMB and GGIW-LMB filters is provided in Section 4.9.

1 The author acknowledges the contributions of Dr Stephan Reuter and Alexander Scheel from Ulm University, Germany, and Dr Karl Granström from the University of Connecticut, USA, for their valuable assistance in the development of the simulation and experimental results.

4.1 INTRODUCTION

A Bayesian approach to multi-target tracking requires models to describe how the measurements are related to the underlying target states. Most traditional trackers use the so-called *standard* measurement model. This is also known as a *point target* model, since it is based on the assumption that each target produces at most one measurement at a given time, and that each measurement originates from at most one target. This model simplifies the development of multi-target trackers, but in practice it is often an unrealistic representation of the true measurement process.

More realistic measurement processes can be handled using more sophisticated *non-standard* measurement models, which may relax the aforementioned assumptions, usually at the expense of increased computation. One example of this is when a group of targets produces a single measurement, known as an unresolved target (or merged measurement) model [21]. This type of model, which was dealt with in Chapter 3, is useful when dealing with low-resolution sensors that cannot generate separate detections for closely spaced targets. On the other hand, higher resolution sensors may produce multiple measurements per target on any given scan. Such cases require the use of an extended target model [58], which is the topic of this chapter.

Measurement models for extended targets typically require two components; a model for the number of measurements generated by each target, and a model for their spatial distribution. These depend strongly on both the sensor characteristics and the type of targets being tracked. For example, in radar tracking, some targets may generate many separate detections, by virtue of the fact that they possess many scatter points. However, other targets may reflect most of the energy away from the receiver, leading to very few detections, or none at all. In general, when a target is far enough away from the sensor, its detections can often be characterised as a cluster of points exhibiting no discernible geometric structure. In such cases, the number of measurements is usually modelled using a Poisson distribution, see for example [58] and [59].

Even in the absence of a specific target structure, it is still possible to estimate the size and shape of a target, known as the target extent. This can be achieved by assuming some general parametric shape for the extent, for which the parameters are estimated based on the spatial arrangement of the observations. An approach that assumes an elliptical extent was proposed in [106], which used a multivariate Gaussian,

parameterised by a random covariance matrix with an inverse Wishart distribution. This was termed a Gaussian inverse Wishart (GIW), and this method enables the target extent to be estimated on-line, instead of requiring prior specification. Further applications and improvements have appeared in [203, 54, 107]. Alternative methods for estimating target extent have also been proposed, see for example [15, 14, 123].

The GIW method has been applied using multi-target filters based on the random finite set (RFS) framework. The probability hypothesis density (PHD) filter, originally developed by Mahler in [126] for the point target model, was modified for extended multi-target filtering in [131]. An implementation of this filter based on the GIW model (GIW-PHD filter) was developed in [67]. The cardinalised PHD (CPHD) filter [129] is a generalisation of the PHD filter, which models the multi-target state as an i.i.d cluster RFS instead of a Poisson RFS. This was applied to extended targets in [124], which also incorporated a modification to the GIW approach [69], enabling the estimation of target measurement rates. This method treats the rate parameter of the Poisson pmf (which characterises the number of measurements generated by a target) as a random variable, whose distribution is modelled as a gamma pdf. This algorithm was called the gamma Gaussian inverse Wishart CPHD (GGIW-CPHD) filter. Extended target PHD and CPHD filters have also been presented in [185, 186], and a general PHD filter that can be applied to extended target filtering as a special case was presented in [38].

The advantage of the (C)PHD filters is that they reduce the computational cost of the Bayes multi-target filter, but to do so, some significant approximations are made. While these approximations avoid explicit data association, their formulation does not explicitly accommodate the estimation of target trajectories. Therefore, if such estimates are required, one must resort to heuristic methods and/or post-processing of the (C)PHD filter output. Furthermore, the PHD filter can produce highly uncertain estimates of the target number due to the Poisson cardinality assumption [126, 199]. Another limitation that affects the CPHD filter is the so-called "spooky" effect [197], which means that when a target is misdetected, some of its probability mass is shifted onto detected targets, regardless of their physical separation. A Bernoulli filter for extended targets was proposed in [166], which does not suffer from these issues, however, it is limited to at most a single target in clutter.

The generalised labelled multi-Bernoulli (GLMB) filter [198, 195] has been shown to outperform both the PHD and CPHD filters, with the added advantage of producing

labelled track estimates, albeit with a higher computational cost. An approximate but computationally cheaper version of this filter was proposed in [164], called the labelled multi-Bernoulli filter (LMB). Also, the first GLMB filter for a non-standard measurement model was developed in [21] (see chapter 3) for cases involving merged measurements, a problem which can be viewed as the dual of the extended target tracking problem.

This chapter presents the development of GLMB and LMB filters for extended multi-target tracking based on the GGIW model. Like the GGIW-CPHD filter of [124], the proposed algorithm requires generating partitions of the measurement set. Since considering all measurement partitions is clearly intractable, it is only possible to process a small subset. It was shown in [123] that using clustering techniques to generate a few alternative partitions can produce good tracking results, so the same approach is adopted in the proposed GLMB filter for extended targets. The resulting algorithms (GGIW-GLMB and GGIW-LMB) are capable of estimating the kinematics and extents of multiple extended targets in clutter, with the advantage of producing continuous target tracks.

4.2 OBSERVATION MODEL FOR MULTIPLE EXTENDED TARGETS

This section develops an RFS-based measurement model for multiple extended targets in clutter. Let us denote the labelled RFS of extended targets that exist at the observation time as $\boldsymbol{X} = \left\{(x_1, \ell_1), \ldots, \left(x_{|\boldsymbol{X}|}, \ell_{|\boldsymbol{X}|}\right)\right\}$. A measurement model is formulated based on the following three assumptions:

A1. A particular extended target with state (x, ℓ) may be detected with probability $P_D(x, \ell)$, or misdetected with probability $1 - P_D(x, \ell)$.

A2. If detected, an extended target with state (x, ℓ) generates a set of detections W with likelihood $\tilde{g}(W|x, \ell)$, which is independent of all other targets.

A3. The sensor generates a Poisson RFS K of false observations with intensity function $\kappa(\cdot)$, which is independent of the target generated observations (i.e. K is distributed according to $\pi_K(K) = e^{-\langle \kappa, 1 \rangle} \kappa^K$).

Denote by $\mathcal{P}_i(Z)$ the set of all partitions that divide a finite measurement set Z into exactly i groups, and by $\mathcal{U}(Z) \in \mathcal{P}_i(Z)$ a particular partition of Z with i groups. For a given multi-target state $\boldsymbol{X}$, denote by $\Theta(\mathcal{U}(Z))$ the space of association mappings $\theta : \mathcal{L}(\boldsymbol{X}) \to \{0, 1, \ldots, |\mathcal{U}(Z)|\}$ such that $\theta(\ell) = \theta(\ell') > 0$ implies $\ell = \ell'$. Finally,

denote by $\mathcal{U}_{\theta(\ell)}(Z)$ the element of the partition $\mathcal{U}(Z)$ corresponding to label ℓ under the mapping θ.

Proposition 13. *Under assumptions A1, A2 and A3, the measurement likelihood function is given by*

$$g(Z|\boldsymbol{X}) = \pi_K(Z) \sum_{i=1}^{|\boldsymbol{X}|+1} \sum_{\substack{\mathcal{U}(Z)\in\mathcal{P}_i(Z) \\ \theta\in\Theta(\mathcal{U}(Z))}} \left[\psi_{\mathcal{U}(Z)}(\cdot;\theta)\right]^{\boldsymbol{X}} \tag{265}$$

where

$$\psi_{\mathcal{U}(Z)}(x,\ell;\theta) = \begin{cases} \frac{P_D(x,\ell)\tilde{g}\left(\mathcal{U}_{\theta(\ell)}(Z)|x,\ell\right)}{[\kappa]^{\mathcal{U}_{\theta(\ell)}(Z)}}, & \theta(\ell) > 0 \\ 1 - P_D(x,\ell), & \theta(\ell) = 0 \end{cases}. \tag{266}$$

Proof. Let us first consider the case of no false detections (i.e. all measurements are target generated). By assumptions A1 and A2, the likelihood of observing a set of detections Y, given a set $\boldsymbol{X}$ of extended targets is given by [129]

$$\pi_D(Y|\boldsymbol{X}) = \sum_{\substack{(W_1,\ldots,W_{|\boldsymbol{X}|}): \\ \uplus_{i=1}^{|\boldsymbol{X}|} W_i = Y}} g'(W_1|x_1,\ell_l)\ldots g'\left(W_{|\boldsymbol{X}|}|x_{|\boldsymbol{X}|},\ell_{|\boldsymbol{X}|}\right), \tag{267}$$

where

$$g'(W|x,\ell) \propto \begin{cases} 1 - P_D(x,\ell), & W = \emptyset \\ P_D(x,\ell)\,\tilde{g}(W|x,\ell), & W \neq \emptyset \end{cases}. \tag{268}$$

A partition of an arbitrary set S is defined to be a disjoint collection of non-empty subsets of S, such that their union is equal to S. Note that in (267), the sets $W_1,\ldots,W_{|\boldsymbol{X}|}$ may be either empty or non-empty, thus, they do not satisfy the definition of a partition of Y. However, the non-empty sets in $W_1,\ldots,W_{|\boldsymbol{X}|}$ do constitute a partition of Y, hence by separating (267) into products over the empty and non-empty W_i's, (267) can be written as

$$\pi_D(Y|\boldsymbol{X}) = [1 - P_D(\cdot)]^{\boldsymbol{X}} \sum_{i=1}^{|\boldsymbol{X}|} \sum_{\mathcal{U}(Y)\in\mathcal{P}_i(Y)} \sum_{1\leq j_1\neq\cdots\neq j_i\leq|\boldsymbol{X}|} \prod_{k=1}^{i} \frac{P_D\left(x_{j_k},\ell_{j_k}\right)\tilde{g}\left(\mathcal{U}_k(Y)|x_{j_k},\ell_{j_k}\right)}{1 - P_D\left(x_{j_k},\ell_{j_k}\right)} \tag{269}$$

where $\mathcal{U}_k(Y)$ denotes the k-th group in partition $\mathcal{U}(Y)$. Following a similar reasoning to [135, pp. 420], this can be expressed as

$$\pi_D(Y|\boldsymbol{X}) = [1 - P_D(\cdot)]^{\boldsymbol{X}} \sum_{i=1}^{|\boldsymbol{X}|} \sum_{\mathcal{U}(Y)\in\mathcal{P}_i(Y)} \sum_{\theta\in\Theta(\mathcal{U}(Y))} \prod_{j:\theta(j)>0} \frac{P_D(x_j,\ell_j)\,\tilde{g}\left(\mathcal{U}_{\theta(j)}(D)\,|x_j,\ell_j\right)}{1-P_D(x_j,\ell_j)}. \tag{270}$$

Let us now consider the case where false observations may also be present. By assumption A3, the set K of false observations has distribution $\pi_K(K)$, and the sets Y and K are independent. The overall measurement set is $Z = Y \cup K$, thus Z is distributed according to the convolution

$$\begin{aligned} g(Z|\boldsymbol{X}) &= \sum_{W\subseteq Z} \pi_K(Z-W)\,\pi_D(W|\boldsymbol{X}) \\ &= \sum_{W\subseteq Z} e^{-\langle\kappa,1\rangle}\kappa^{Z-W}\,[q_D]^{\boldsymbol{X}} \sum_{i=1}^{|\boldsymbol{X}|} \sum_{\substack{\mathcal{U}(W)\in\mathcal{P}_i(W)\\ \theta\in\Theta(\mathcal{U}(W))}} \prod_{j:\theta(j)>0} \frac{P_D(x_j,\ell_j)\,\tilde{g}\left(\mathcal{U}_{\theta(j)}(W)\,|x_j,\ell_j\right)}{1-P_D(x_j,\ell_j)} \\ &= e^{-\langle\kappa,1\rangle}\kappa^{Z}\,[1-P_D(\cdot)]^{\boldsymbol{X}} \sum_{W\subseteq Z} \sum_{i=1}^{|\boldsymbol{X}|} \sum_{\substack{\mathcal{U}(W)\in\mathcal{P}_i(W)\\ \theta\in\Theta(\mathcal{U}(W))}} \prod_{j:\theta(j)>0} \frac{P_D(x_j,\ell_j)\,\tilde{g}\left(\mathcal{U}_{\theta(j)}(W)\,|x_j,\ell_j\right)}{\left(1-P_D(x_j,\ell_j)\right)[\kappa]^{\mathcal{U}_{\theta(j)}(W)}} \end{aligned} \tag{271}$$

where the last line follows from the fact that $\kappa^W = \prod_{j:\theta(j)>0}[\kappa]^{\mathcal{U}_{\theta(j)}(W)}$, since $\mathcal{U}(W)$ is a partition of W. Finally, this can be simplified by treating the set $Z-W$ as an additional element appended to each $\mathcal{U}(W)$, thereby transforming it into a partition of Z. In doing so, the double summation over $W\subseteq Z$ and partitions $\mathcal{U}(W)\in\mathcal{P}_i(W)$ up to size $|\boldsymbol{X}|$, can be expressed as a summation over partitions of Z up to size $|\boldsymbol{X}|+1$ as follows

$$g(Z|\boldsymbol{X}) = \pi_K(Z)\,[1-P_D(\cdot)]^{\boldsymbol{X}} \sum_{i=1}^{|\boldsymbol{X}|+1} \sum_{\substack{\mathcal{U}(Z)\in\mathcal{P}_i(Z)\\ \theta\in\Theta(\mathcal{U}(Z))}} \prod_{j:\theta(j)>0} \frac{P_D(x_j,\ell_j)\,\tilde{g}\left(\mathcal{U}_{\theta(j)}(Z)\,|x_j,\ell_j\right)}{\left(1-P_D(x_j,\ell_j)\right)[\kappa]^{\mathcal{U}_{\theta(j)}(Z)}}. \tag{272}$$

Observe that the $1-P_D(x_j,\ell_j)$ in the denominator cancels out the corresponding term in the product $[1-P_D]^{\boldsymbol{X}}$ when $\theta(j)>0$, leaving one $1-P_D(x_j,\ell_j)$ term for each $j:\theta(j)=0$. Hence, (272) can be equivalently expressed in the form (265)-(266). □

In general, an exact calculation of the likelihood (265) will be numerically intractable, because the sets of measurement partitions and group-to-target mappings

can become extremely large. However, it has been shown that in many practical situations, it is only necessary to consider a small subset of these partitions to achieve good performance [67, 69]. Additionally, the set of group-to-target mappings can be substantially reduced using a ranked assignment algorithm, thereby cutting down the number of insignificant terms in the likelihood even further.

4.3 EXTENDED TARGET STATE-SPACE MODEL

This section describes the extended target state-space model, and the class of probability distributions used to model a single extended target. We begin by introducing some notation:

- $\mathbb{R}^+$ is the space of positive real numbers
- $\mathbb{R}^n$ is the space of real n-dimensional vectors
- $\mathbb{S}^n_{++}$ is the space of $n \times n$ positive definite matrices
- $\mathbb{S}^n_{+}$ is the space of $n \times n$ positive semi-definite matrices
- $\mathcal{GAM}(x_{\mathcal{R}}; \alpha, \beta)$ is the gamma probability density function (pdf) defined on $x_{\mathcal{R}} > 0$, with shape $\alpha > 0$, and inverse scale $\beta > 0$:

$$\mathcal{GAM}(x_{\mathcal{R}}; \alpha, \beta) = \frac{\beta^{\alpha}}{\Gamma(\alpha)} x_{\mathcal{R}}^{\alpha-1} e^{-\beta x_{\mathcal{R}}}$$

- $\mathcal{N}(x_{\mathcal{C}}; m, P)$ is the multivariate Gaussian pdf defined on $x_{\mathcal{C}} \in \mathbb{R}^n$, with mean $m \in \mathbb{R}^n$ and covariance $P \in \mathbb{S}^n_{+}$

$$\mathcal{N}(x_{\mathcal{C}}; m, P) = \frac{1}{\sqrt{(2\pi)^n |P|}} e^{-\frac{1}{2}(x_{\mathcal{C}}-m)^T P^{-1}(x_{\mathcal{C}}-m)}$$

- $\mathcal{IW}_d(x_{\mathcal{E}}; v, V)$ is the inverse Wishart distribution defined on $x_{\mathcal{E}} \in \mathbb{S}^d_{++}$, with degrees of freedom $v > 2d$, and scale matrix $V \in \mathbb{S}^d_{++}$ [71]

$$\mathcal{IW}_d(x_{\mathcal{E}}; v, V) = \frac{2^{-\frac{v-d-1}{2}} |V|^{\frac{v-d-1}{2}}}{\Gamma_d\left(\frac{v-d-1}{2}\right) |x_{\mathcal{E}}|^{\frac{v}{2}}} e^{-\frac{1}{2}\mathrm{tr}\left(V x_{\mathcal{E}}^{-1}\right)}$$

where $\Gamma_d(\cdot)$ is the multivariate gamma function, and $\mathrm{tr}(\cdot)$ takes the trace of a matrix.

- I_d is the idenity matrix of dimension d.
- $A \otimes B$ is the Kronecker product of matrices A and B

The goal is to estimate three pieces of information about each target; the average number of measurements it generates, the kinematic state of its centroid, and its extent. The extended target state is thus modelled as the triple

$$x = (x_{\mathcal{R}}, x_{\mathcal{C}}, x_{\mathcal{E}}) \in \mathbb{R}^+ \times \mathbb{R}^n \times \mathbb{S}^d_{++}, \tag{273}$$

where $x_{\mathcal{R}} \in \mathbb{R}^+$ is the rate parameter of a Poisson distribution that models the number of measurements generated by the target, $x_{\mathcal{C}} \in \mathbb{R}^n$ is a vector that describes the state of the target centroid, and $x_{\mathcal{E}} \in \mathbb{S}^d_{++}$ is a covariance matrix that describes the target extent around the centroid. The density of the rate parameter is modelled as a Gamma distribution, the kinematics as a Gaussian distribution, and the covariance of the extent as an inverse-Wishart distribution. The density of the extended target state is thus the product of these three distributions, denoted as a *gamma Gaussian inverse Wishart* (GGIW) distribution on the space $\mathbb{R}^+ \times \mathbb{R}^n \times \mathbb{S}^d_{++}$, given by

$$\begin{aligned} p(x) &= p_{\mathcal{R}}(x_{\mathcal{R}})\, p_{\mathcal{C}}(x_{\mathcal{C}}|x_{\mathcal{E}})\, p_{\mathcal{E}}(x_{\mathcal{E}}) \\ &= \mathcal{GAM}(x_{\mathcal{R}}; \alpha, \beta) \times \mathcal{N}(x_{\mathcal{C}}; m, P \otimes x_{\mathcal{E}}) \times \mathcal{IW}_d(x_{\mathcal{E}}; v, V) \\ &\triangleq \mathcal{GGIW}(x; \zeta) \end{aligned} \tag{274}$$

where $\zeta = (\alpha, \beta, m, P, v, V)$ is an array that encapsulates the GGIW density parameters. The following sections describe the prediction and Bayes update procedures for a GGIW distribution representing a single extended target.

4.3.1 *GGIW Prediction*

The predicted density $p_+(\cdot)$ of an extended target is given by the following Champan-Kolmogorov equation

$$p_+(x) = \int f(x|x')\, p(x')\, dx', \tag{275}$$

where $p(\cdot) = \mathcal{GGIW}(\cdot; \zeta')$ is the posterior density at the current time with parameters $\zeta' = (\alpha', \beta', m', P', v', V')$, and $f(\cdot|\cdot)$ is the transition density from the current time to the next time. This has no closed form solution, hence we resort to making a

GGIW approximation for $p_+(x)$. First, it is assumed that the transition density can be written as the product [124]

$$f(x|x') = f_{\mathcal{R}}(x_{\mathcal{R}}|x'_{\mathcal{R}})\, f_{\mathcal{C}}(x_{\mathcal{C}}|x_{\mathcal{E}}, x'_{\mathcal{C}})\, f_{\mathcal{E}}(x_{\mathcal{E}}|x'_{\mathcal{E}}), \tag{276}$$

which yields the following predicted density

$$\begin{aligned} p_+(x) = &\int \mathcal{GAM}(x'_{\mathcal{R}}; \alpha', \beta')\, f_{\mathcal{R}}(x_{\mathcal{R}}|x'_{\mathcal{R}})\, dx'_{\mathcal{R}} \\ &\times \int \mathcal{N}(x'_{\mathcal{C}}; m', P' \otimes x_{\mathcal{E}})\, f_{\mathcal{C}}(x_{\mathcal{C}}|x_{\mathcal{E}}, x'_{\mathcal{C}})\, dx'_{\mathcal{C}} \\ &\times \int \mathcal{IW}_d(x'_{\mathcal{E}}; v', V')\, f_{\mathcal{E}}(x_{\mathcal{E}}|x'_{\mathcal{E}})\, dx'_{\mathcal{E}}. \end{aligned} \tag{277}$$

If the dynamic model is linear-Gaussian with the form

$$f_{\mathcal{C}}(x_{\mathcal{C}}|x_{\mathcal{E}}, x'_{\mathcal{C}}) = \mathcal{N}(x_{\mathcal{C}}; (F \otimes I_d)\, x'_{\mathcal{C}}, Q \otimes x_{\mathcal{E}}), \tag{278}$$

the kinematic component (i.e. the second line in (277)) can be solved in closed form as

$$\int \mathcal{N}(x'_{\mathcal{C}}; m', P' \otimes x_{\mathcal{E}}) f_{\mathcal{C}}(x_{\mathcal{C}}|x_{\mathcal{E}}, x'_{\mathcal{C}})\, dx'_{\mathcal{C}} = \mathcal{N}(x_{\mathcal{C}}; m, P \otimes x_{\mathcal{E}}), \tag{279}$$

where

$$m = (F \otimes I_d)\, m', \tag{280}$$

$$P = FP'F^T + Q. \tag{281}$$

Exact analytical solutions still cannot be obtained for the measurement rate and target extension components, which is addressed by applying some additional approximations. The following approximation is used for the measurement rate component, which was proposed in [69],

$$\int \mathcal{GAM}(x'_{\mathcal{R}}; \alpha', \beta')\, f_{\mathcal{R}}(x_{\mathcal{R}}|x'_{\mathcal{R}})\, dx'_{\mathcal{R}} \approx \mathcal{GAM}(x_{\mathcal{R}}; \alpha, \beta), \tag{282}$$

where

$$\alpha = \frac{\alpha'}{\mu}, \tag{283}$$

$$\beta = \frac{\beta'}{\mu}. \tag{284}$$

In the above, $\mu = \frac{1}{1-1/w}$ is an exponential forgetting factor with window length $w > 1$. This approximation is based on the heuristic assumption that $E[x_{\mathcal{R}}] = E[x'_{\mathcal{R}}]$, and $\text{Var}(x_{\mathcal{R}}) = \text{Var}(x'_{\mathcal{R}}) \times \mu$, i.e. the prediction operation retains the expected value of the density, and increases its variance by a factor of μ.

The predicted density for the extension component is approximated as

$$\int \mathcal{IW}_d(x'_{\mathcal{E}}; v', V') f_{\mathcal{E}}(x_{\mathcal{E}}|x'_{\mathcal{E}}) dx'_{\mathcal{E}} \approx \mathcal{IW}_d(x_{\mathcal{E}}; v, V), \tag{285}$$

as proposed in [106], where

$$v = e^{-T/\tau} v', \tag{286}$$

$$V = \frac{v - d - 1}{v' - d - 1} V'. \tag{287}$$

Similarly to the measurement rate component, this approximation assumes that the prediction retains the expected value and increases the uncertainty in the density. For an inverse-Wishart distribution, the degrees of freedom parameter is related to the uncertainty, with lower values yielding increased uncertainty. A temporal decay constant τ is thus used in (286) to govern the reduction in the degrees of freedom. Based on the calculated value for v, the expression for V retains the expected value of the inverse-Wishart distribution through the prediction.

The above yields an approximate representation of the predicted GGIW density $p_+(x) \approx \mathcal{GGIW}(x; \zeta)$, where $\zeta = (\alpha, \beta, m, P, v, V)$ is the array of predicted GGIW density parameters defined by equations (280), (281), (283), (284), (286) and (287).

4.3.2 *GGIW Update*

In the proposed GGIW-(G)LMB filter, each extended target will need to undergo measurement updates using various subsets of the measurements received on each scan. The following describes the procedure for computing the posterior GGIW for a single target with predicted density $p(\cdot) = \mathcal{GGIW}(\cdot; \zeta)$, for a given measurement set W generated by an extended target. Note that unlike the GGIW prediction, the GGIW update has an exact closed form, and does not require any approximations. Each individual detection within W is assumed to be generated from the model

$$z_k = (H \otimes I_d) x_{\mathcal{C},k} + w_k \tag{288}$$

where $H = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$, and $w_k \sim \mathcal{N}(0, x_{\mathcal{E},k})$ is i.i.d. Gaussian measurement noise with covariance given by the target extent matrix $x_{\mathcal{E},k}$. If detected, the target generates a set $W = \{z_1, \ldots, z_{|W|}\}$ of measurements from the model (288), whose cardinality $|W|$ follows a Poisson distribution. The single-target likelihood function for this model is given by [67, 69]

$$\tilde{g}(W|x) = \mathcal{PS}(|W|; x_{\mathcal{R}}) \prod_{j=1}^{|W|} \mathcal{N}(z_j; \tilde{H} x_{\mathcal{C}}, x_{\mathcal{E}}) \tag{289}$$

where $\tilde{H} = H \otimes I_d$, and $\mathcal{PS}(\cdot; \lambda)$ denotes the Poisson pmf with mean λ. Note that (289) will be substituted into (266) for the multiple extended-target likelihood, but in this section, the dependence on the label ℓ is dropped, since we are considering only a single target.

Let us assume that the predicted density $p(x)$ is a GGIW given by (274), and we wish to compute the posterior density via Bayes rule

$$p(x|W) = \frac{\tilde{g}(W|x)\, p(x)}{\int \tilde{g}(W|x')\, p(x')\, dx'}. \tag{290}$$

The numerator of (290) is given by

$$\begin{aligned} \tilde{g}(W|x)\, p(x) = {} & \mathcal{GAM}(x_{\mathcal{R}}; \alpha, \beta)\, \mathcal{N}(x_{\mathcal{C}}; m, P \otimes x_{\mathcal{E}})\, \mathcal{IW}_d(x_{\mathcal{E}}; v, V) \\ & \times \mathcal{PS}(|W|; x_{\mathcal{R}}) \prod_{j=1}^{|W|} \mathcal{N}(z_j | \tilde{H} x_{\mathcal{C}}, x_{\mathcal{E}}). \end{aligned} \tag{291}$$

Notice that the measurement rate and kinematics-extension components are independent, and can thus be treated separately. The measurement rate component is given by [69]

$$\mathcal{GAM}(x_{\mathcal{R}}; \alpha, \beta)\, \mathcal{PS}(|W|; x_{\mathcal{R}}) = \mathcal{GAM}(x_R; \alpha_W, \beta_W)\, \eta_{\mathcal{R}}(W; \zeta) \tag{292}$$

where

$$\alpha_W = \alpha + |W|, \tag{293}$$

$$\beta_W = \beta + 1, \tag{294}$$

$$\eta_{\mathcal{R}}(W; \zeta) = \frac{1}{|W|!} \frac{\Gamma(\alpha_W)\, \beta^{\alpha}}{\Gamma(\alpha)\, (\beta_W)^{\alpha_W}}, \tag{295}$$

and its integral with respect to $x_{\mathcal{R}}$, which will appear in the denominator of (290), is thus given by

$$\int \mathcal{GAM}\left(x_{\mathcal{R}};\alpha,\beta\right)\mathcal{PS}\left(|W|;x_{\mathcal{R}}\right)dx_{\mathcal{R}}=\eta_{\mathcal{R}}\left(W;\zeta\right). \tag{296}$$

Note that in (295), $\eta_{\mathcal{R}}\left(W;\zeta\right)$ is expressed as a function of the measurement set W and the prior GGIW parameters ζ. This calculation corresponds to evaluating a negative-binomial pdf.

The kinematics-extension component is given by [67]

$$\begin{aligned}&\mathcal{N}\left(x_{\mathcal{C}};m,P\otimes x_{\mathcal{E}}\right)\mathcal{IW}_d\left(x_{\mathcal{E}};v,V\right)\prod_{j=1}^{n}\mathcal{N}\left(z_j|\tilde{H}x_{\mathcal{C}},x_{\mathcal{E}}\right)\\&\quad=\mathcal{N}\left(x_{\mathcal{C}};m_W,P_W\otimes x_{\mathcal{E}}\right)\mathcal{IW}\left(x_{\mathcal{E}};v_W,V_W\right)\eta_{\mathcal{C},\mathcal{E}}\left(W;\zeta\right),\end{aligned} \tag{297}$$

where

$$m_W=m+\left(K\otimes I_d\right)\epsilon, \tag{298}$$

$$P_W=P-KSK^T, \tag{299}$$

$$v_W=v+|W|, \tag{300}$$

$$V_W=V+S^{-1}\epsilon\epsilon^T+\Psi, \tag{301}$$

$$\eta_{\mathcal{C},\mathcal{E}}\left(W;\zeta\right)=\frac{\left(\pi^{|W|}|W|\right)^{-\frac{d}{2}}|V|^{\frac{v}{2}}\Gamma_d\left(\frac{v_W}{2}\right)}{S^{\frac{d}{2}}|V_W|^{\frac{v_W}{2}}\Gamma_d\left(\frac{v}{2}\right)}, \tag{302}$$

and

$$K=PH^TS^{-1}, \tag{303}$$

$$S=HPH^T+\frac{1}{|W|}, \tag{304}$$

$$\epsilon=\bar{w}-\left(H\otimes I_d\right)m, \tag{305}$$

$$\Psi=\sum_{w\in W}\left(w-\bar{w}\right)\left(w-\bar{w}\right)^T, \tag{306}$$

$$\bar{w}=\frac{1}{|W|}\sum_{w\in W}w. \tag{307}$$

The integral of (297) with respect to $x_{\mathcal{C}}$ and $x_{\mathcal{E}}$ is

$$\begin{aligned}&\int\int\mathcal{N}\left(x_{\mathcal{C}};m,P\otimes x_{\mathcal{E}}\right)\mathcal{IW}_d\left(x_{\mathcal{E}};v,V\right)\\&\qquad\times\prod_{j=1}^{n}\mathcal{N}\left(z_j|\tilde{H}x_{\mathcal{C}},x_{\mathcal{E}}\right)dx_{\mathcal{C}}dx_{\mathcal{E}}=\eta_{\mathcal{C},\mathcal{E}}\left(W;\zeta\right).\end{aligned} \tag{308}$$

Notice that $\eta_{\mathcal{C},\mathcal{E}}(W;\zeta)$ in (302) is proportional to a matrix variate generalised beta type II pdf, and that the denominator in (290), i.e. the normalising constant for the complete GGIW Bayes update, is given by the product $\eta_{\mathcal{R}}(W;\zeta) \times \eta_{\mathcal{C},\mathcal{E}}(W;\zeta)$. This will be used during the update step of the GGIW-GLMB filter to compute the weights of the posterior density components.

4.3.3 *Illustrative Example of the GGIW Recursion*

To show how the GGIW prediction and update techniques work, this section contains some illustrative examples depicting the input and output of one iteration of the single-target GGIW recursion. Figure 25 shows an example of a posterior GGIW density, where the diagram on the left represents the kinematic and target extension components, and the graph on the right shows the distribution of the measurement rate parameter. The dashed circle represents the 95% confidence region of the positional component of the Gaussian distributed target centroid, and the filled ellipse represents the mean of the inverse-Wishart distributed target extent. Naturally, there is an uncertainty associated with the target extent, but since the inverse-Wishart is a matrix-variate distribution, this uncertainty is difficult to depict on a 2-dimensional plot, thus only the mean is shown. The measurement rate is gamma distributed, with a mean of 5.

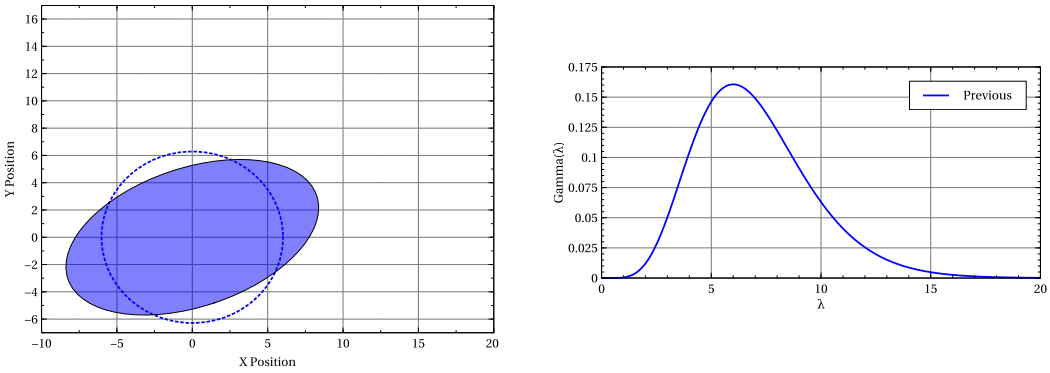


Figure 25: Single-target GGIW example, previous posterior GGIW density.

Figure 26 shows the predicted GGIW distribution, along with the previous posterior. Similarly to a Kalman filter prediction, the mean of the centroid is shifted due to the estimated target velocity, and the uncertainty in the centroid's location is slightly increased. The mean of the target extent remains unchanged. The uncertainty in the extent increases, but again, this is difficult to represent on the plot. The measurement

rate distribution becomes slightly wider to represent a small increase in the measurement rate uncertainty.

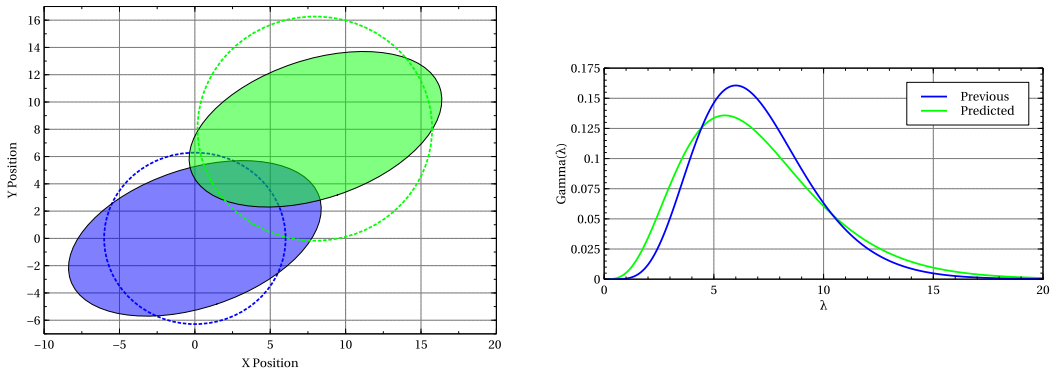


Figure 26: Single-target GGIW example, predicted GGIW density.

Figure 27 shows an example of the posterior GGIW density, updated with a set of two measurements. The uncertainty in the centroid location shrinks, and the mean extent changes to reflect the new measurement data. The measurement rate distribution is shifted down, since only two measurements were generated.

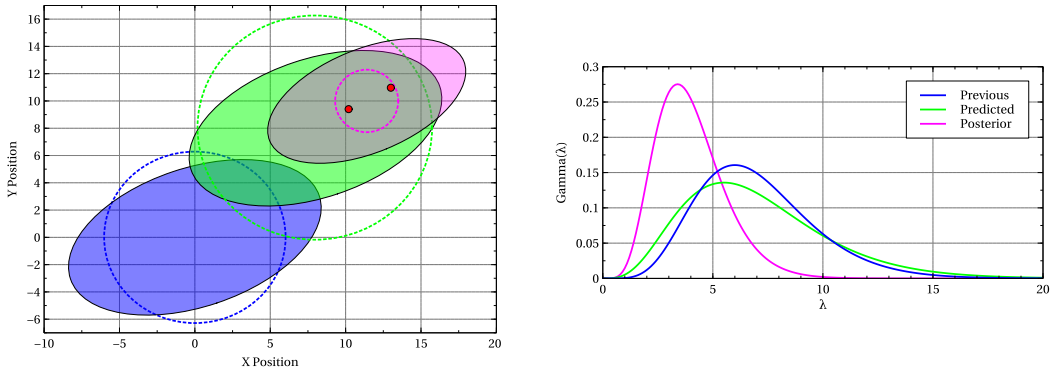


Figure 27: Single-target GGIW example, posterior GGIW updated with two measurements.

Finally, Figure 28 shows an alternative posterior GGIW density, updated with a set of 15 measurements. This time, the uncertainty in the centroid location shrinks down even further than the two-measurement case, and the mean of the posterior extent more closely matches the distribution of the measurements. This can be understood by the fact that a larger set of target generated measurements naturally provides more information about the target's location and extent than a smaller set. The rate distribution is shifted to reflect the fact that the measurement rate has increased.

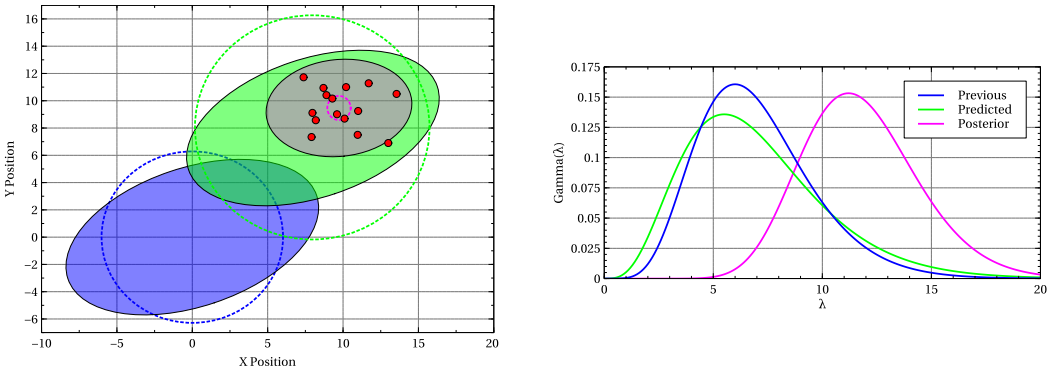


Figure 28: Single-target GGIW example, posterior GGIW updated with 15 measurements.

4.4 GLMB FILTER FOR EXTENDED TARGETS

A GLMB filter for extended targets is now proposed, based on the measurement likelihood and state-space models described in the previous sections. The GLMB filter consists of two steps, prediction and update. Since the standard birth/death model is used for the multi-target dynamics, the prediction step is identical to that of the standard GLMB filter derived in [198]. For completeness, the final prediction equations are revisited here, and the reader is referred to [198] for more details. Denote by $P_S(x,\ell)$ the probability that a target with state (x,ℓ) survives to the next time step. The birth density is an LMB with label space $\mathbb{B}$, weight $w_B(\cdot)$ and single target densities $p_B(\cdot,\ell)$. If the multi-target posterior is a GLMB of the form (159) with label space $\mathbb{L}$, then the predicted multi-target density at the next time step is

$$\boldsymbol{\pi}_+(\boldsymbol{X}) = \Delta(\boldsymbol{X}) \sum_{c\in\mathbb{C}} w_+^{(c)}(\mathcal{L}(\boldsymbol{X})) \left[p_+^{(c)}(\cdot)\right]^{\boldsymbol{X}} \tag{309}$$

where

$$w_+^{(c)}(L) = w_B(L\cap\mathbb{B})\, w_S^{(c)}(L\cap\mathbb{L}), \tag{310}$$

$$p_+^{(c)}(x,\ell) = 1_{\mathbb{L}}(\ell)\, p_S^{(c)}(x,\ell) + (1-1_{\mathbb{L}}(\ell))\, p_B(x,\ell), \tag{311}$$

$$p_S^{(c)}(x,\ell) = \frac{\int P_S(x,\ell)\, f(x|x',\ell)\, p^{(c)}(x',\ell)\, dx'}{\eta_S^{(c)}(\ell)}, \tag{312}$$

$$\eta_S^{(c)}(\ell) = \int\int P_S(x,\ell)\, f(x|x',\ell)\, p^{(c)}(x',\ell)\, dx'dx, \tag{313}$$

$$w_S^{(c)}(J) = \left[\eta_S^{(c)}\right]^J \sum_{I\subseteq\mathbb{L}} 1_I(J) \left[q_S^{(c)}\right]^{I-J} w^{(c)}(I), \tag{314}$$

$$q_S^{(c)}(\ell) = \int (1-P_S(x,\ell))\, p^{(c)}(x,\ell)\, dx. \tag{315}$$

The function $f(\cdot|\cdot,\ell)$ is the single-target transition kernel, which in this case is the GGIW transition defined in Section 4.3.1.

Clearly, the difference between the standard GLMB and extended target GLMB filters will lie in the measurement update procedure, since the measurement likelihood function has a different form. The update for the extended target GLMB is specified by the following proposition.

Proposition 14. *If the prior is a GLMB of the form (159), then under the extended multi-target likelihood function (265), the posterior is a GLMB given by*

$$\boldsymbol{\pi}(\boldsymbol{X}|Z) = \Delta(\boldsymbol{X}) \sum_{c\in\mathbb{C}} \sum_{i=1}^{|\boldsymbol{X}|+1} \sum_{\substack{\mathcal{U}(Z)\in\mathcal{P}_i(Z)\\ \theta\in\Theta(\mathcal{U}(Z))}} w^{(c,\theta)}_{\mathcal{U}(Z)}(\mathcal{L}(\boldsymbol{X})) \left[p^{(c,\theta)}(\cdot|\mathcal{U}(Z))\right]^{\boldsymbol{X}} \tag{316}$$

where

$$w^{(c,\theta)}_{\mathcal{U}(Z)}(L) = \frac{w^{(c)}(L)\left[\eta^{(c,\theta)}_{\mathcal{U}(Z)}\right]^{L}}{\sum_{c\in\mathbb{C}}\sum_{J\subseteq\mathbb{L}}\sum_{i=1}^{|J|+1}\sum_{\substack{\mathcal{U}(Z)\in\mathcal{P}_i(Z)\\ \theta\in\Theta(\mathcal{U}(Z))}} w^{(c)}(J)\left[\eta^{(c,\theta)}_{\mathcal{U}(Z)}\right]^{J}}, \tag{317}$$

$$p^{(c,\theta)}(x,\ell|\mathcal{U}(Z)) = \frac{p^{(c)}(x,\ell)\,\psi_{\mathcal{U}(Z)}(x,\ell;\theta)}{\eta^{(c,\theta)}_{\mathcal{U}(Z)}(\ell)}, \tag{318}$$

$$\eta^{(c,\theta)}_{\mathcal{U}(Z)}(\ell) = \int p^{(c)}(x,\ell)\,\psi_{\mathcal{U}(Z)}(x,\ell;\theta)\,dx \tag{319}$$

in which $\psi_{\mathcal{U}(Z)}(x,\ell;\theta)$ is given by (266).

Proof. The product of the prior distribution and the likelihood is

$$\begin{aligned}
&\boldsymbol{\pi}(\boldsymbol{X})\,g(Z|\boldsymbol{X})\\
&\quad= \Delta(\boldsymbol{X})\,\pi_K(Z)\sum_{c\in\mathbb{C}}\sum_{i=1}^{|\boldsymbol{X}|+1}\sum_{\substack{\mathcal{U}(Z)\in\mathcal{P}_i(Z)\\ \theta\in\Theta(\mathcal{U}(Z))}} w^{(c)}(\mathcal{L}(\boldsymbol{X}))\left[p^{(c)}(\cdot)\,\psi_{\mathcal{U}(Z)}(\cdot;\theta)\right]^{\boldsymbol{X}}\\
&\quad= \Delta(\boldsymbol{X})\,\pi_K(Z)\sum_{c\in\mathbb{C}}\sum_{i=1}^{|\boldsymbol{X}|+1}\sum_{\substack{\mathcal{U}(Z)\in\mathcal{P}_i(Z)\\ \theta\in\Theta(\mathcal{U}(Z))}} w^{(c)}(\mathcal{L}(\boldsymbol{X}))\left[p^{(c,\theta)}(\cdot|\mathcal{U}(Z))\,\eta^{(c,\theta)}_{\mathcal{U}(Z)}(\cdot)\right]^{\boldsymbol{X}}\\
&\quad= \Delta(\boldsymbol{X})\,\pi_K(Z)\sum_{c\in\mathbb{C}}\sum_{i=1}^{|\mathcal{L}(\boldsymbol{X})|+1}\sum_{\substack{\mathcal{U}(Z)\in\mathcal{P}_i(Z)\\ \theta\in\Theta(\mathcal{U}(Z))}} w^{(c)}(\mathcal{L}(\boldsymbol{X}))\left[\eta^{(c,\theta)}_{\mathcal{U}(Z)}(\cdot)\right]^{\mathcal{L}(\boldsymbol{X})}\left[p^{(c,\theta)}(\cdot|\mathcal{U}(Z))\right]^{\boldsymbol{X}}.
\end{aligned} \tag{320}$$

The set integral of (320) with respect to $\boldsymbol{X}$ is given by

$$\begin{aligned}
&\int \pi(\boldsymbol{X})\, g(Z|\boldsymbol{X})\, \delta\boldsymbol{X} \\
&\quad = g_C(Z) \sum_{c\in\mathbb{C}} \int \sum_{i=1}^{|\mathcal{L}(\boldsymbol{X})|+1} \sum_{\substack{\mathcal{U}(Z)\in\mathcal{P}_i(Z)\\ \theta\in\Theta(\mathcal{U}(Z))}} \Delta(\boldsymbol{X})\, w^{(c)}(\mathcal{L}(\boldsymbol{X})) \\
&\qquad \times \left[\eta^{(c,\theta)}_{\mathcal{U}(Z)}\right]^{\mathcal{L}(\boldsymbol{X})} \left[p^{(c,\theta)}(\cdot|\mathcal{U}(Z))\right]^{\boldsymbol{X}} \delta\boldsymbol{X} \\
&\quad = g_C(Z) \sum_{c\in\mathbb{C}} \sum_{j=0}^{\infty} \frac{1}{j!} \sum_{(\ell_1,\ldots,\ell_j)\in\mathbb{L}^j} \sum_{i=1}^{j+1} \sum_{\substack{\mathcal{U}(Z)\in\mathcal{P}_i(Z)\\ \theta\in\Theta(\mathcal{U}(Z))}} \Delta(\{(x_1,\ell_1),\ldots,(x_j,\ell_j)\}) \\
&\qquad \times w^{(c)}(\{\ell_1,\ldots,\ell_j\}) \left[\eta^{(c,\theta)}_{\mathcal{U}(Z)}\right]^{\{\ell_1,\ldots,\ell_j\}} \\
&\qquad \times \int \left[p^{(c,\theta)}(\cdot|\mathcal{U}(Z))\right]^{\{(x_1,\ell_1),\ldots,(x_j,\ell_j)\}} d(x_1,\ldots,x_j) \\
&\quad = g_C(Z) \sum_{c\in\mathbb{C}} \sum_{L\subseteq\mathbb{L}} \sum_{i=1}^{|L|+1} \sum_{\substack{\mathcal{U}(Z)\in\mathcal{P}_i(Z)\\ \theta\in\Theta(\mathcal{U}(Z))}} w^{(c)}(L) \left[\eta^{(c,\theta)}_{\mathcal{U}(Z)}\right]^{L}. \qquad (321)
\end{aligned}$$

In the above, the second line is obtained by applying [198, Proposition 2], and taking the parts that are only label-dependent outside the resulting integral. The last line is obtained by observing the fact that the distinct label indicator function limits the summation over $j : 0 \to \infty$ and $\ell_{1:j} \in \mathbb{L}^j$ to a summation over the subsets of $\mathbb{L}$. Substituting (320) and (321) into (56), yields the posterior density (316). □

Proposition 14 establishes that the GLMB is a conjugate prior with respect to the extended multi-target measurement likelihood function. Since the posterior density contains a sum over partitions of the measurement set, its exact calculation is generally infeasible, because the number of possible partitions is usually very large. A practical implementation of this filter requires that the summation be limited to only a small subset of the measurement partitions. These must be chosen a manner such that the highest weighted posterior components are included in the calculation, and the components with insignificant weights are excluded. More details of our implementation can be found in Section 4.6.1.

4.5 LMB FILTER FOR EXTENDED TARGETS

Having derived the GLMB recursion for extended targets, it is straightforward to obtain the LMB recursion using the technique described in Section 2.5.5. The key princi-

ple of the LMB filter is to simplify the representation of the multi-target density after each update cycle, in order to reduce the algorithm's computational complexity. Instead of maintaining the full GLMB representation from one iteration to the next, it is approximated as an LMB density after each measurement update. In the subsequent iteration, the prediction is carried out using this LMB representation. The predicted LMB is then separated into independent target clusters, and the predicted LMB of each cluster is converted back to a GLMB in preparation for the next measurement update.

The prediction procedure for the extended target LMB filter is the same as the standard LMB filter prediction, except that each single-target density is propagated forward using the approximate GGIW prediction as described in Section 4.3.1. The measurement update also proceeds in a similar manner to the standard LMB update, except that the GLMB update for each cluster is carried out using the result of Proposition 14. This also requires that a collection of feasible partitions of the measurement set be identified prior to performing the update.

After the measurement update, the posterior GLMB (316) for each cluster is approximated by an LMB with matching probability hypothesis density. The parameters of this approximate LMB are given by

$$r^{(\ell)} = \sum_{c\in\mathbb{C}} \sum_{L\subseteq\mathbb{L}_+} \sum_{i=1}^{|L|+1} \sum_{\substack{\mathcal{U}(Z)\in\mathcal{P}_i(Z)\\ \theta\in\Theta(\mathcal{U}(Z))}} w^{(c,\theta)}_{\mathcal{U}(Z)}(L)\, 1_I(\ell)\,, \tag{322}$$

$$p^{(\ell)}(x) = \frac{1}{r^{(\ell)}} \sum_{c\in\mathbb{C}} \sum_{L\subseteq\mathbb{L}_+} \sum_{i=1}^{|L|+1} \sum_{\substack{\mathcal{U}(Z)\in\mathcal{P}_i(Z)\\ \theta\in\Theta(\mathcal{U}(Z))}} w^{(c,\theta)}_{\mathcal{U}(Z)}(L)\, 1_I(\ell)\, p^{(\theta)}(x,\ell)\,. \tag{323}$$

Equation (323) means that the pdf of each track in the LMB becomes a mixture of GGIW densities. To avoid the number of components in these mixtures growing too large, it is necessary to reduce them by a process of pruning and merging. For GGIW mixtures, this can be achieved using the techniques proposed in [69, 68], which have been previously applied in the context of mixture reduction for the GGIW-CPHD filter [124].

4.6 IMPLEMENTATION

This section provides more details on the implementation of both the GLMB and LMB filters for extended targets. We begin in Section 4.6.1 with the prediction and update steps of the GGIW-GLMB filter, then in Section 4.6.2 we describe the modifications necessary to implement the GGIW-LMB filter. Pseudo-code for these functions is provided at the end of the chapter in Section 4.9.

4.6.1 *GGIW-GLMB Filter*

4.6.1.1 *Prediction*

Since the GGIW-GLMB filter uses the standard model for target birth/death, the prediction step of the GGIW-GLMB filter is identical to that of the standard GLMB filter. The only difference is that the single-target prediction is given by the approximate GGIW prediction as discussed in Section 4.3.1. Hence, the prediction procedure is the same as specified in Algorithm 5 (see Chapter 2), where the function `TargetPredict`, is replaced with the GGIW prediction.

4.6.1.2 *Measurement Partitioning*

Similarly to the extended target (C)PHD filter, the main barrier to implementing the extended target GLMB filter is the fact that computing the posterior density in (316) involves taking a sum over all partitions of the measurement set. Even for small measurement sets, exhaustively enumerating the partitions is usually intractable, because the number of possibilities (given by the Bell number) grows combinatorially with the number of elements. Therefore, to make the filter computationally tractable, the update procedure can only consider a small subset of the possible partitions. A list of the most likely measurement partitions is therefore constructed, and passed as a parameter to the update function.

Ideally, the retained partitions should be those that give rise to GLMB components with the highest posterior weights, such that the effect of truncation error is minimised. Although it is difficult to establish a method that can guarantee this, the use of clustering techniques to generate the most likely partitions has been shown to produce favourable results in the context of PHD/CPHD filtering [67, 69]. In the

implementation of the GGIW-GLMB filter, feasible sets of measurement partitions are generated using a combination of distance-based clustering and the expectation-maximisation algorithm, in a similar manner to [67] and [69]. See Algorithms 23 and 24 for pseudo-code of the measurement partitioning procedure.

4.6.1.3 *Update*

After a set of feasible partitions has been generated, each unique grouping of measurements is used to update the single-target GGIW pdfs in the GLMB density, using (298)-(307). The implementation is simplified by assuming that the detection probability is dependent on the target label only (i.e. $P_D(x,\ell) = P_D(\ell)$), which yields the following analytical expression for (319),

$$\eta_{\mathcal{U}(Z)}^{(c,\theta)}(\ell) = \begin{cases} \frac{P_D(\ell)\eta_{\mathcal{R}}\left(\mathcal{U}_{\theta(\ell)}(Z);p^{(c)}\right)\eta_{\mathcal{C},\mathcal{E}}\left(\mathcal{U}_{\theta(\ell)}(Z);p^{(c)}\right)}{[\kappa]^{\mathcal{U}_{\theta(\ell)}(Z)}}, & \theta(\ell) > 0 \\ 1 - P_D(\ell), & \theta(\ell) = 0 \end{cases} \tag{324}$$

where $\eta_{\mathcal{R}}$ and $\eta_{\mathcal{C},\mathcal{E}}$ are given by (295) and (302) respectively. For each predicted component c, and partition $\mathcal{U}(Z)$, the following cost matrix is calculated for the assignment of measurement groups to targets,

$$-\ln\begin{pmatrix} \phi^{(c)}(\ell_1,\mathcal{U}_1(Z)) & \cdots & \phi^{(c)}(\ell_1,\mathcal{U}_m(Z)) & \phi^{(c)}(\ell_1,\emptyset) & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \phi^{(c)}(\ell_n,\mathcal{U}_1(Z)) & \cdots & \phi^{(c)}(\ell_n,\mathcal{U}_m(Z)) & 0 & \cdots & \phi^{(c)}(\ell_n,\emptyset) \end{pmatrix}, \tag{325}$$

where

$$\phi^{(c)}(\ell,W) = \frac{P_D(\ell)\,\eta_{\mathcal{R}}\left(W;p^{(c)}\right)\eta_{\mathcal{C},\mathcal{E}}\left(W;p^{(c)}\right)}{[\kappa]^{W}}, \tag{326}$$

$$\phi^{(c)}(\ell,\emptyset) = 1 - P_D(\ell), \tag{327}$$

and $\mathcal{U}_i(Z)$ denotes the i-th measurement group in partition $\mathcal{U}(Z)$. Each row in (325) corresponds to an extended target, and each column corresponds to either a group of measurements in Z, or a misdetection. Murty's algorithm is then used to generate highly weighted assignments of measurement groups to targets, each of which forms a component in the posterior GLMB density. Pseudo-code for the update procedure is provided in Section 4.9.1 (Algorithm 25).

4.6.2 *GGIW-LMB Filter*

As with the GGIW-GLMB filter, the GGIW-LMB filter uses the standard multi-target birth/death model, thus the prediction step of the GGIW-LMB filter is identical to that of the standard LMB filter, except that the single-target prediction is given by the approximate GGIW prediction as discussed in Section 4.3.1. Hence, the prediction procedure is the same as specified in Algorithm 13, where the function `TargetPredict`, is replaced with the GGIW prediction.

The measurement update of the GGIW-LMB filter differs from the standard LMB filter update, as it requires a list of feasible measurement partitions to be specified, and the measurement update for each cluster is replaced by the GGIW-GLMB update, as specified by Algorithm 25. The overall GGIW-LMB recursion and the measurement update procedure are specified in Section 4.9.2 (Algorithms 26 and 27).

4.7 SIMULATION RESULTS

In this section, the performance of the GGIW-GLMB and GGIW-LMB filters are compared to an extended target CPHD filter[2] [124] using the cardinality estimation error and the OSPA distance [173]. Since the standard OSPA only penalizes cardinality and state errors, a modified version of the OSPA metric [124, Section VI] incorporating measurement rates and target extent is used in the evaluation.

The target centroids follow the dynamic model

$$x_{\mathcal{C},k+1} = \left(F_{k+1|k} \otimes I_d\right) x_{\mathcal{C},k} + v_k \tag{328}$$

where $v_k \sim \mathcal{N}\left(0, Q_{k+1|k}\right)$ is a $d \times 1$ i.i.d. process noise vector, I_d is the identity matrix of dimension d, and $F_{k+1|k}$ and $Q_{k+1|k}$ are

$$F_{k+1|k} = \begin{bmatrix} 1 & T & \frac{1}{2}T^2 \\ 0 & 0 & T \\ 0 & 0 & e^{-T/\theta} \end{bmatrix}, \tag{329}$$

$$Q_{k+1|k} = \left[\Sigma^2\left(1 - e^{-2T/\theta}\right) \operatorname{diag}\left(\begin{bmatrix} 0 & 0 & 1 \end{bmatrix}\right)\right] \otimes x_{\mathcal{E},k+1}. \tag{330}$$

2 Thanks to Dr Karl Granström from the University of Connecticut, USA, for providing the Matlab code for the GGIW-CPHD filter used in these comparisons.

In the above, T is the sampling period, Σ is the scalar acceleration standard deviation, and θ is the manoeuvre correlation time. In these simulations, we use parameter values of $T = 1\,s$, $\theta = 1\,s$ and $\Sigma = 0.1\,m/s^2$.

The forgetting factor used by the filters in (283) for the prediction of target measurement rates is set to $\mu = 1.25$, and the temporal decay constant in (286) for the prediction of the target extent is $\tau = 5\,s$. The probability of target survival is set to $P_S = 0.99$. Both the GGIW-GLMB and GGIW-LMB filters use a static birth model, which is represented as a GGIW-LMB distribution. Within this birth distribution, the parameters of the gamma components are $\alpha_0 = 10$ and $\beta_0 = 1$, the inverse-Wishart component parameters are $\nu_0 = 10$ and $V_0 = 100 \times I_2$, and the kinematic components have means m_0 which are located close to the true target starting positions with zero initial velocity and covariance $P_0 = \text{diag}\left(\begin{bmatrix} 10 & 2.5 \end{bmatrix}\right)^2$. Additionally, a GGIW-LMB filter with adaptive (i.e. measurement driven) birth model is considered, which is referred to as GGIW-LMB-ab. This filter uses the same birth parameter values for α_0, β_0, ν_0 and P_0 as the filters with static birth, however, the values of m_0 and V_0 are computed on-line using the measurements, instead of being predefined.

Three scenarios were simulated, the first two of which were used in [124] to compare the performance of the GGIW-PHD filter and the GGIW-CPHD filter (note that all scenarios are 2-dimensional, i.e. $d = 2$). Scenario 1 runs for 200 time steps, and consists of four targets that appear/disappear at different times. The targets generate measurements with a detection probability of $P_D = 0.8$ and the clutter measurements follow a Poisson distribution with a mean number of 30 per time step. Due to lower detection probability, higher clutter rate, and target birth/death, the estimation of the cardinality is challenging in this scenario. Scenario 2 runs for 100 time steps and consists of two targets that are present for the entire scenario. The two targets are spatially well separated at the beginning, then move in parallel at close distance, before separating again towards the end. In this scenario, the detection probability is $P_D = 0.98$, and two different clutter rates were tested; very light clutter with a Poisson mean of 10 per scan (as used in [123], for the sake of comparison), and very high clutter with a Poisson mean of 5000 per scan. This scenario is used to illustrate the filter performance for the difficult problem of tracking closely spaced targets [67, 69]. Since the target-generated measurements are close together, they often appear as a single cluster in the sensor data, rather than multiple separate clusters. Figures 29 and 30

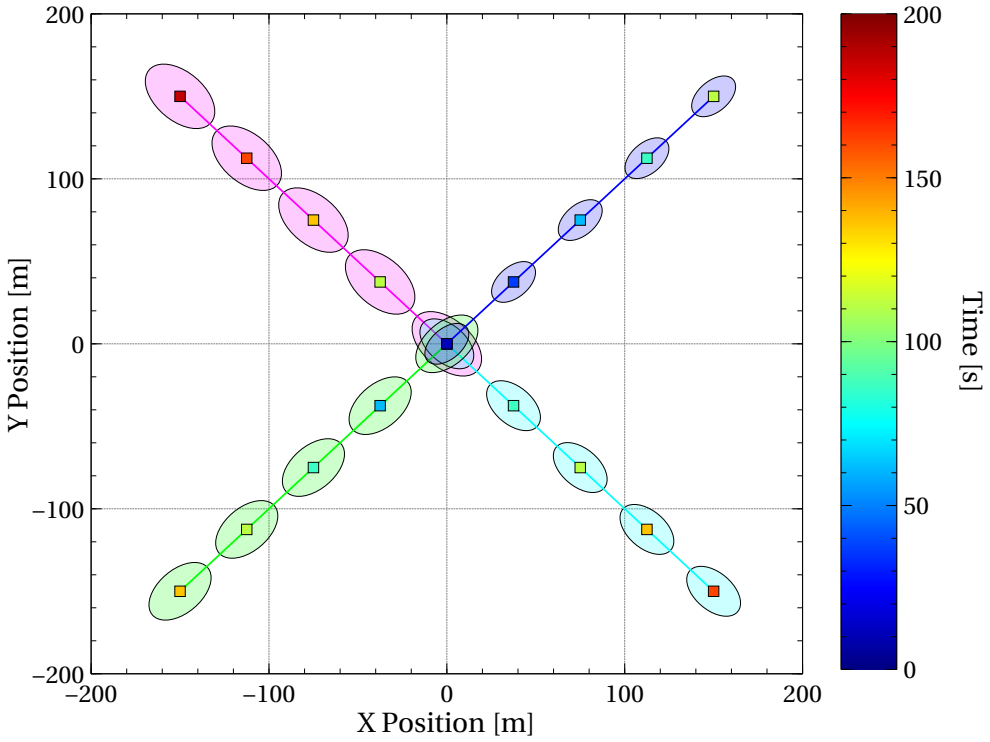


Figure 29: Ground truth for extended-target test scenario 1. The tracks start in the origin at different times and move outwards. The colours of the square markers indicate the time when the target is at that location, and the ellipses represent the true target extents.

depict the true trajectories of the targets for scenarios 1 and 2 respectively, and figure 31 shows an example of the measurements for scenario 2 in the high clutter rate case.

Scenario 3 is used to test the so-called "spooky" effect [197]. The scenario has two targets that are spatially separated by at least 1km for all 50 time steps. The probability of detection was set to $P_D = 0.9$ and clutter Poisson rate was 10. The measurements were generated such that one target is always detected, and the other target is detected on all time steps, except for steps 20, 40 and 41. The ground truth for scenario 3 is shown in Figure 32.

For the first two scenarios, 1000 Monte Carlo runs were carried out in order to compare the performance of the five different GGIW filters: GLMB, LMB, LMB with adaptive birth process (LMB-ab), CPHD and PHD. For scenario 1, Figures 33 and 34 show the cardinality estimation performance, and Figure 33 shows the mean OSPA distances. The PHD filter performs worst, exhibiting highly variable cardinality estimates, and the highest mean OSPA distance. The CPHD filter performs better, by virtue of its improved cardinality estimation capabilities. The GLMB and LMB fil-

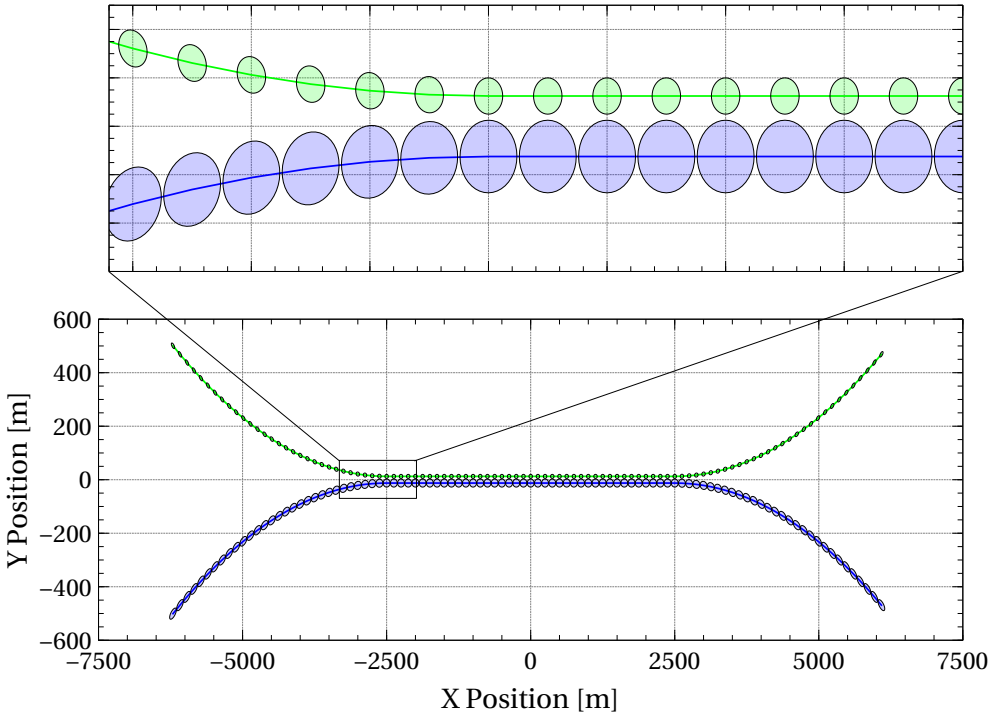


Figure 30: Ground truth for extended-target test scenario 2. The tracks start simultaneously on the left hand side, then move close together, and separate again towards the end of the scenario.

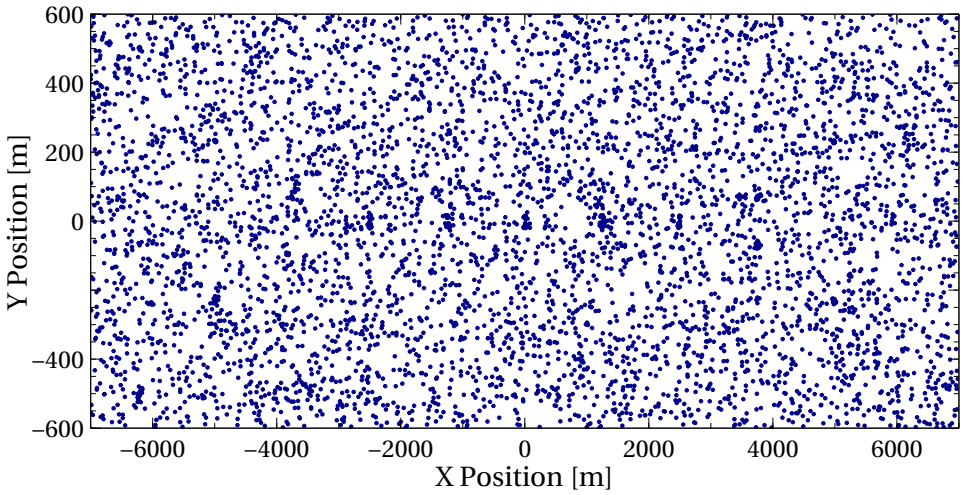


Figure 31: Example measurements for extended-target test scenario 2, high clutter rate case. The target-generated measurements from every tenth scan are shown, against the clutter background from a single scan.

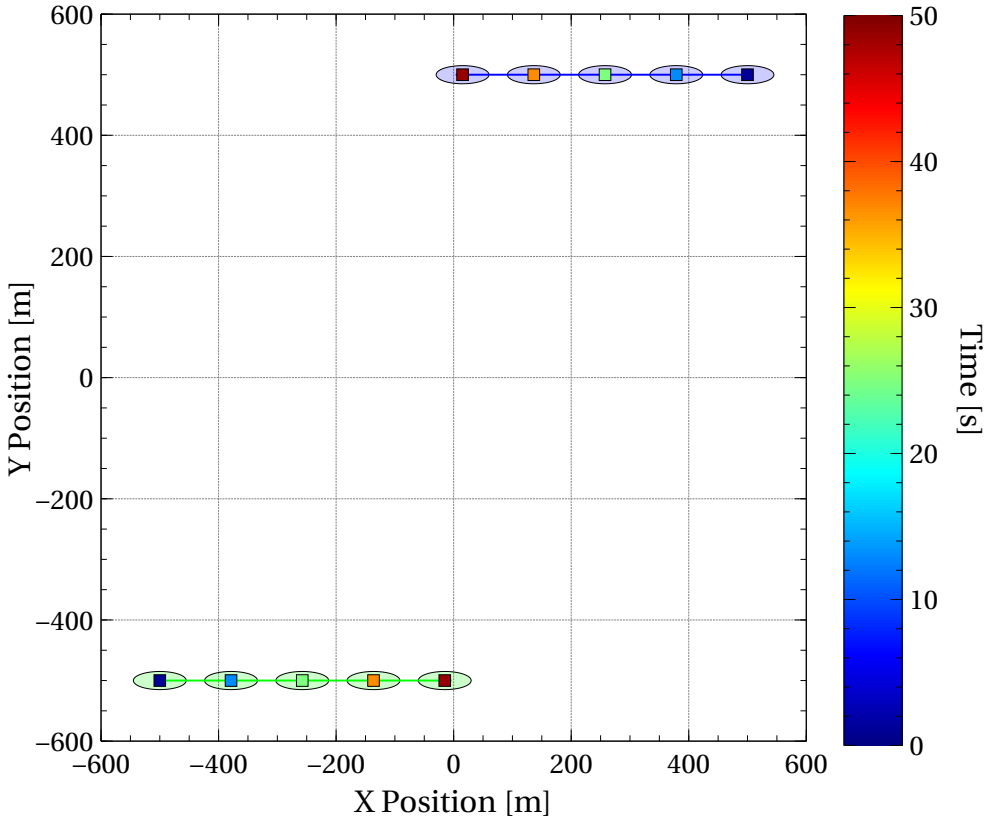


Figure 32: Ground truth for extended-target test scenario 3. The top target starts on the right and moves to the left, and the bottom target starts on the left and moves to the right.

ters have approximately equal performance, with the GLMB performing marginally better on average. The LMB-ab filter takes slightly longer to converge to the correct cardinality due to the unknown birth density, however the filter eventually reaches the same error as the GLMB and LMB with known birth density. This is expected, since the other three filters have the advantage of knowing the region where new targets will appear.

The execution times for the Matlab implementation of the algorithms (mean ± one standard deviation) for scenario 1 are 3.95 ± 3.41s for the GLMB filter, 0.19 ± 0.29s for the LMB filter, and 2.20 ± 0.47s for the CPHD filter. Since the LMB-ab filter only uses clusters with more than four measurements as birth candidates, it is even faster than the LMB filter. Since the LMB filter partitions the tracks and measurements into approximately statistically independent groups [164, 172], its computation times are less than those of the CPHD filter.

Figure 36 shows the mean OSPA distances, and the mean of the absolute cardinality errors for scenario 2. Similar to scenario 1, the GLMB filter outperforms the LMB, CPHD and PHD filters. Again, the cardinality estimate of the LMB-ab filter takes longer to converge to the correct value, however, the LMB-ab filter has lower OSPA distance and smaller cardinality error than the LMB filter after time 78. This is due to the fact that, in some of the runs, the LMB filter lost one of the tracks because the measurement clusters were very close together. Even after the targets move apart, the filter with static birth density is unable to start a new track on the lost target. However, the filter with adaptive birth density has the capability to start a new track at the lost target's current location, leading to improved performance.

The results of the high clutter rate simulation for scenario 2 are shown in Figure 37. This time, the LMB filter with static birth produces significantly larger errors during the middle of the scenario, and this error continues until the end of the simulation. In this case, the higher clutter rate leads to the LMB filter dropping more tracks, which cannot be reinitialized under the static birth model. The LMB with adaptive birth also suffers from a similar increase in error when the targets are closely spaced, however, the error improves dramatically towards the end, as it initiates new tracks for those that were previously dropped. The results also show that GLMB filter is able to maintain the tracks more reliably than the LMB filter in the case of high clutter, however, as mentioned previously, this comes with an increase in computational cost.

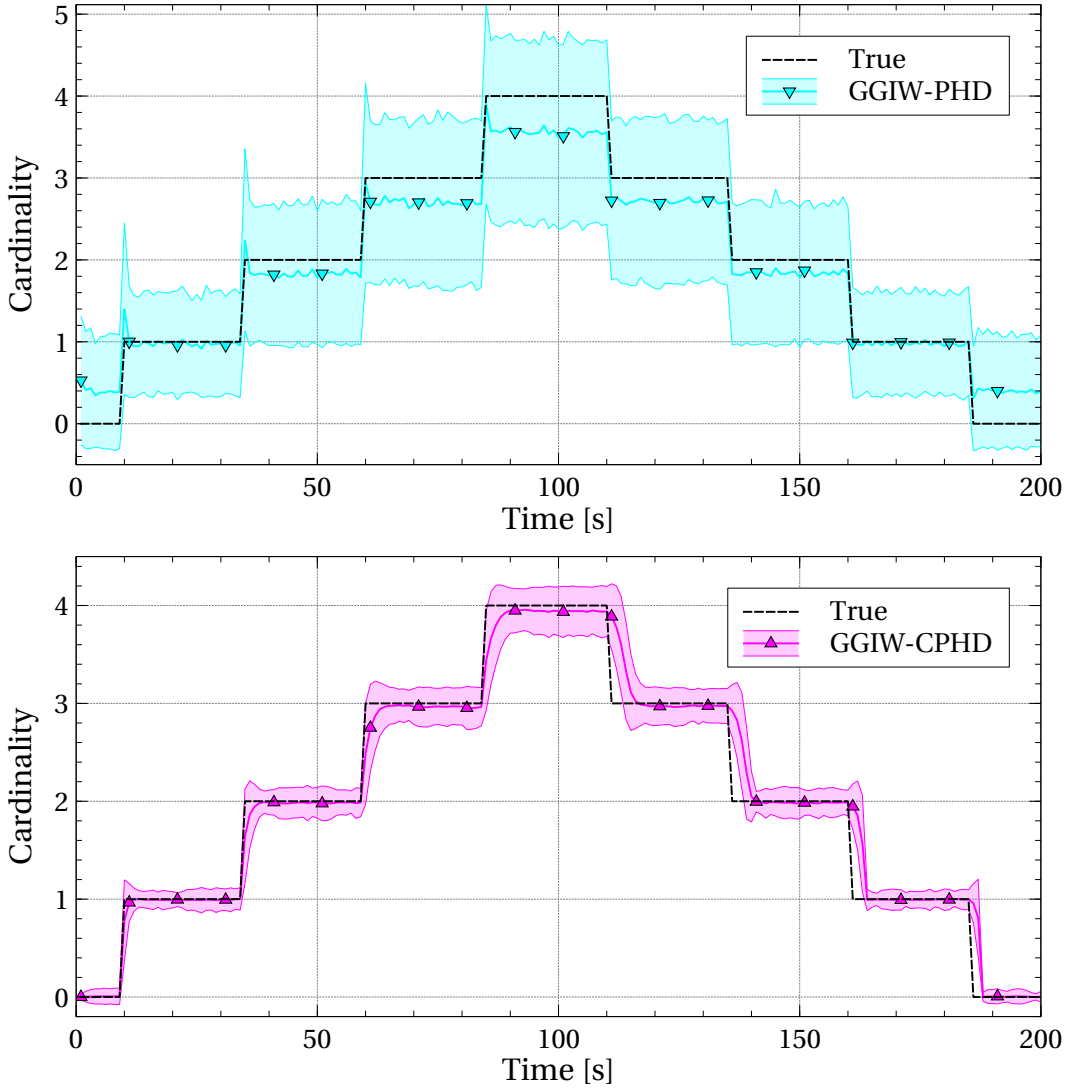


Figure 33: Cardinality estimation performance for GGIW-PHD and GGIW-CPHD filters in extended-target test scenario 1.

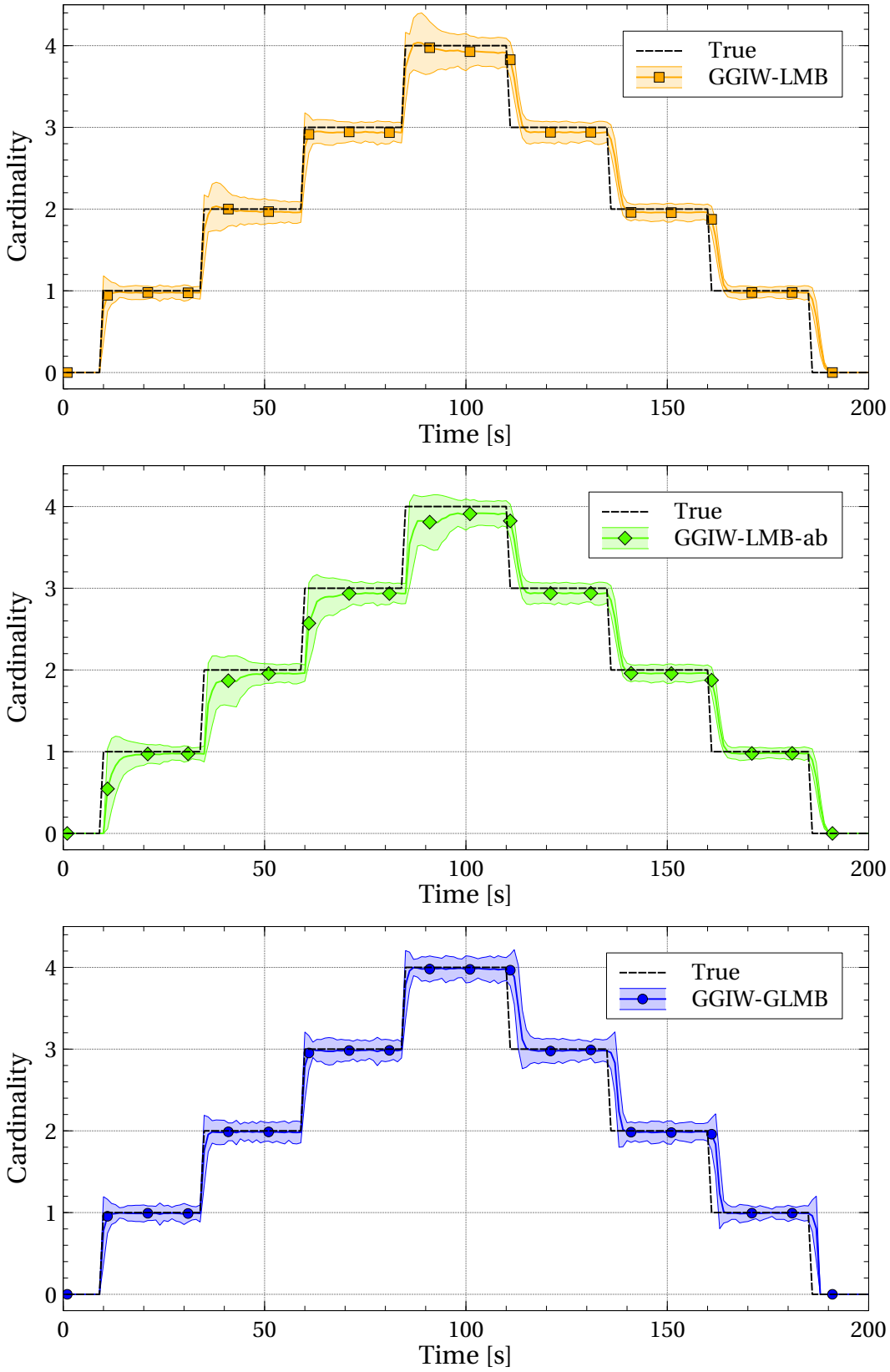


Figure 34: Cardinality estimation performance for GGIW-LMB, GGIW-LMB-ab and GGIW-GLMB filters in extended-target test scenario 1.

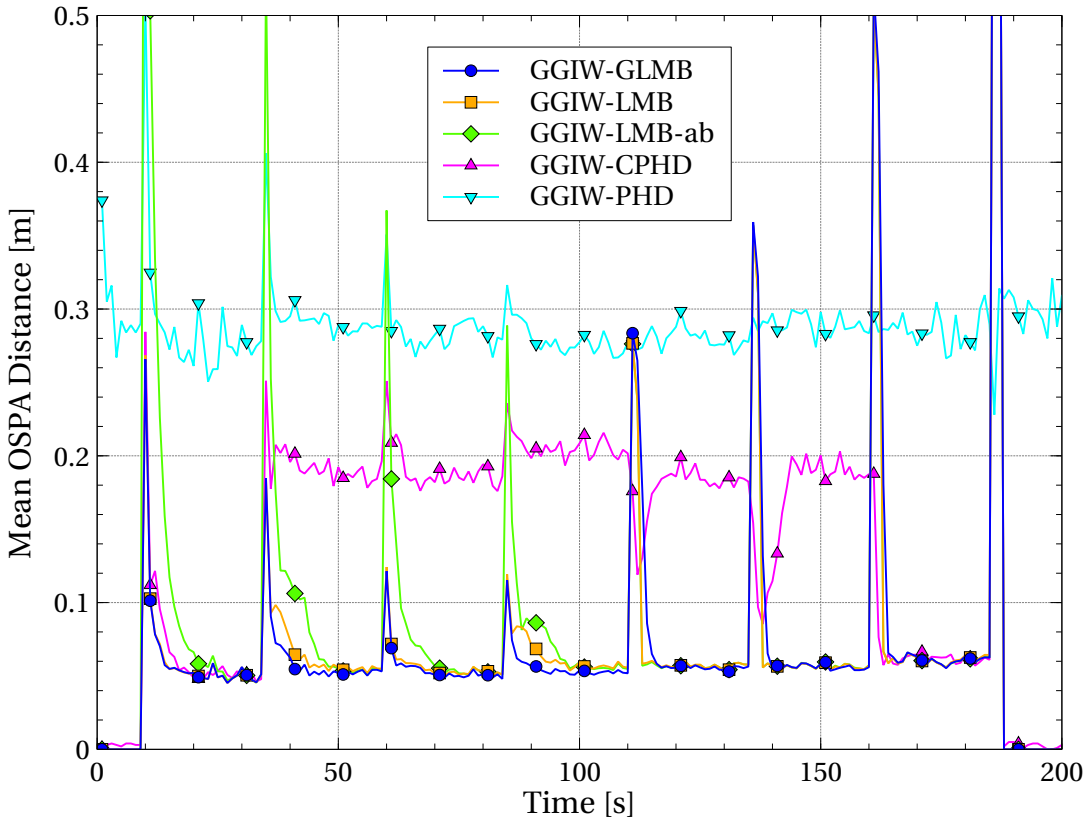


Figure 35: OSPA distance for extended-target filters in test scenario 1.

In scenario 3 we compare the PHD, CPHD and LMB filters. The estimated weights (for the PHD and CPHD) and existence probabilities (for the LMB) for a single run are shown in Figure 38. The PHD filter suffers from a positive bias (weight around 1.1), and the weight drops quickly when there are missed detections. The CPHD clearly suffers from the "spooky" effect [197], as the weight of the detected target increases when the other target is misdetected. In comparison, the LMB filter performs better, as the probability of existence of the detected target is unaffected when the other target is not detected. Also, the decrease in the existence probability following the missed detections is more conservative compared to the decrease in the weights for the PHD and CPHD filters.

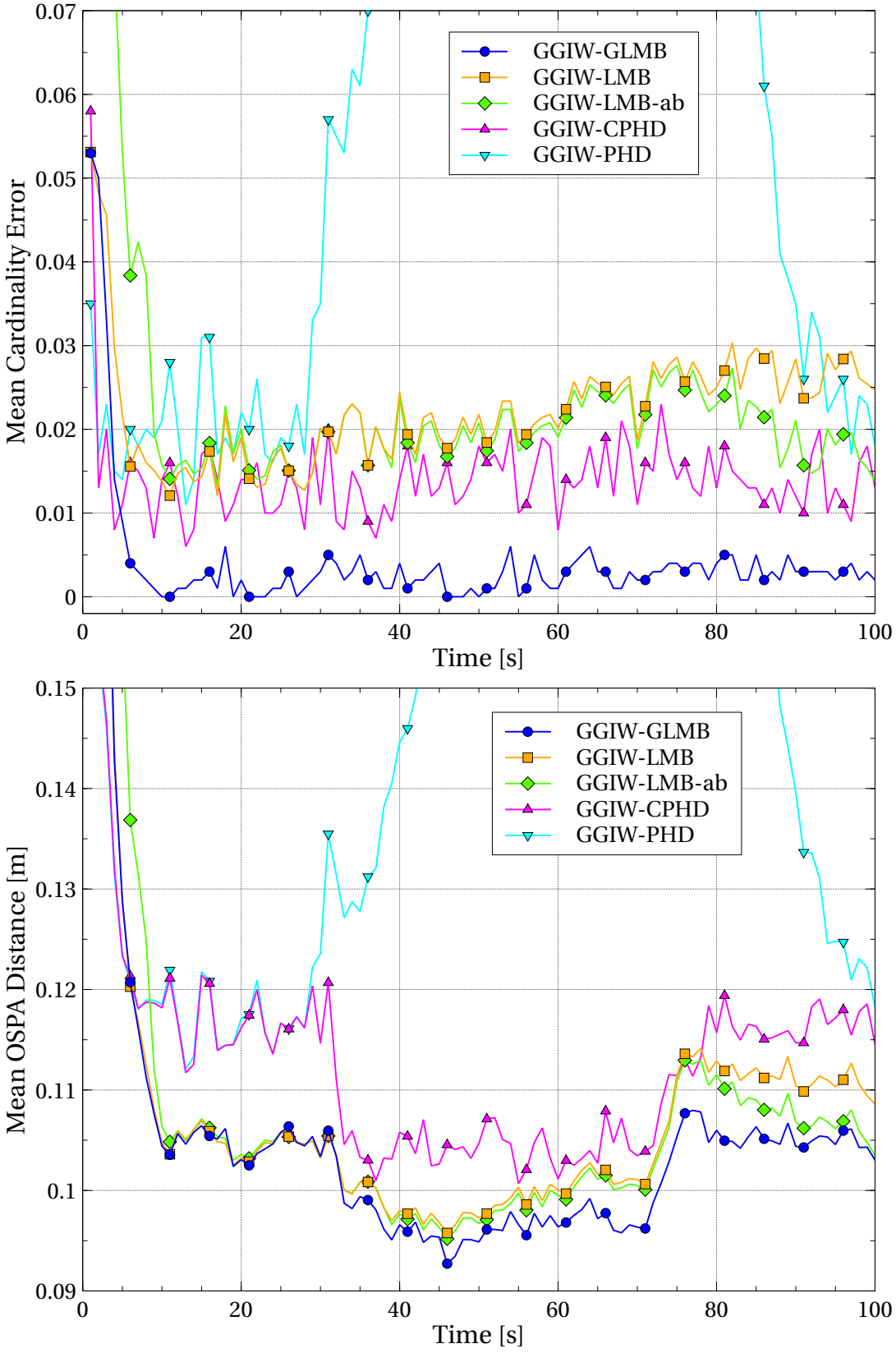


Figure 36: Cardinality error and OSPA distance for extended-target filters in test scenario 2, low clutter case.

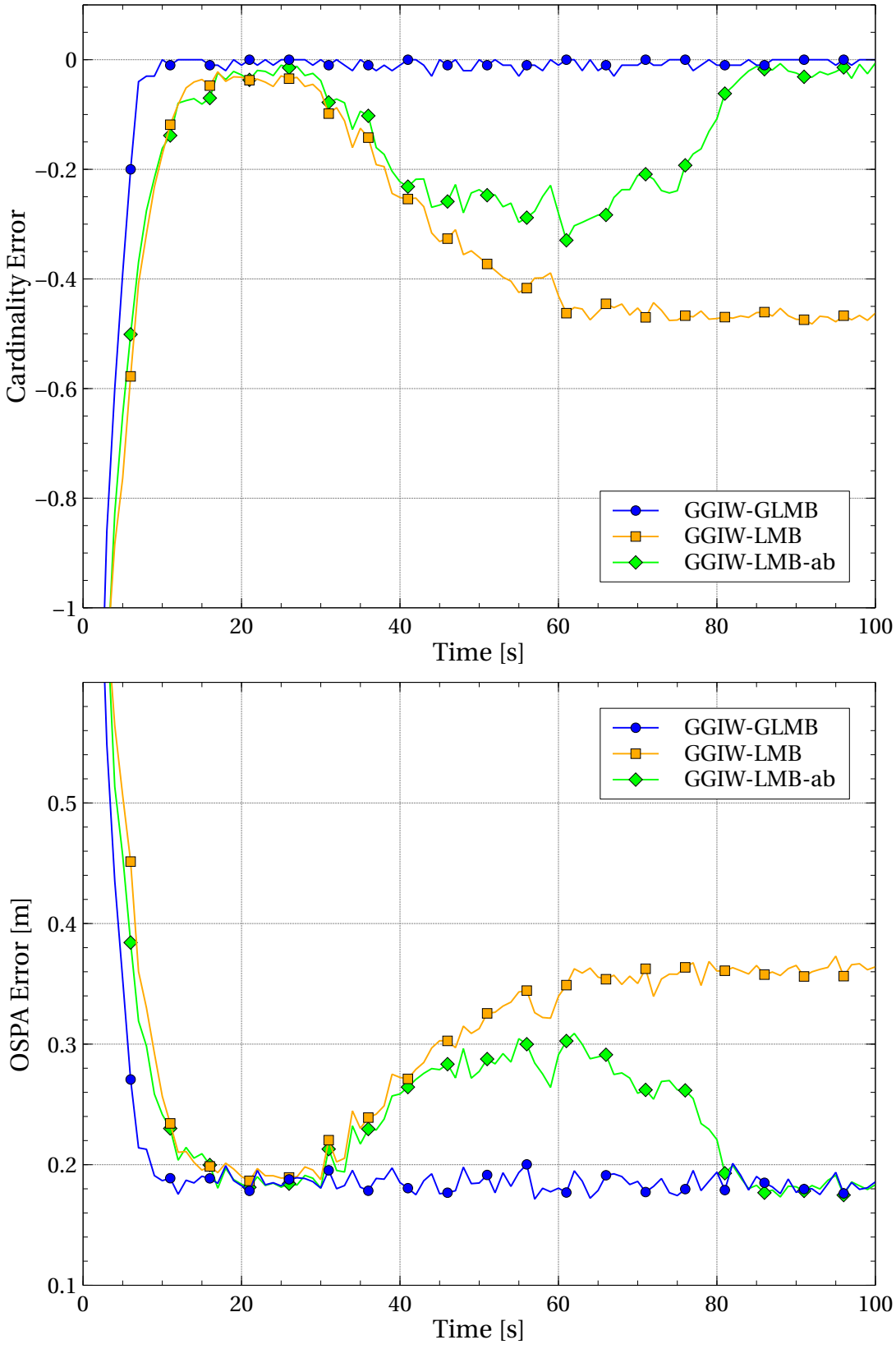


Figure 37: Cardinality error and OSPA distance for extended-target filters in test scenario 2, high clutter case.

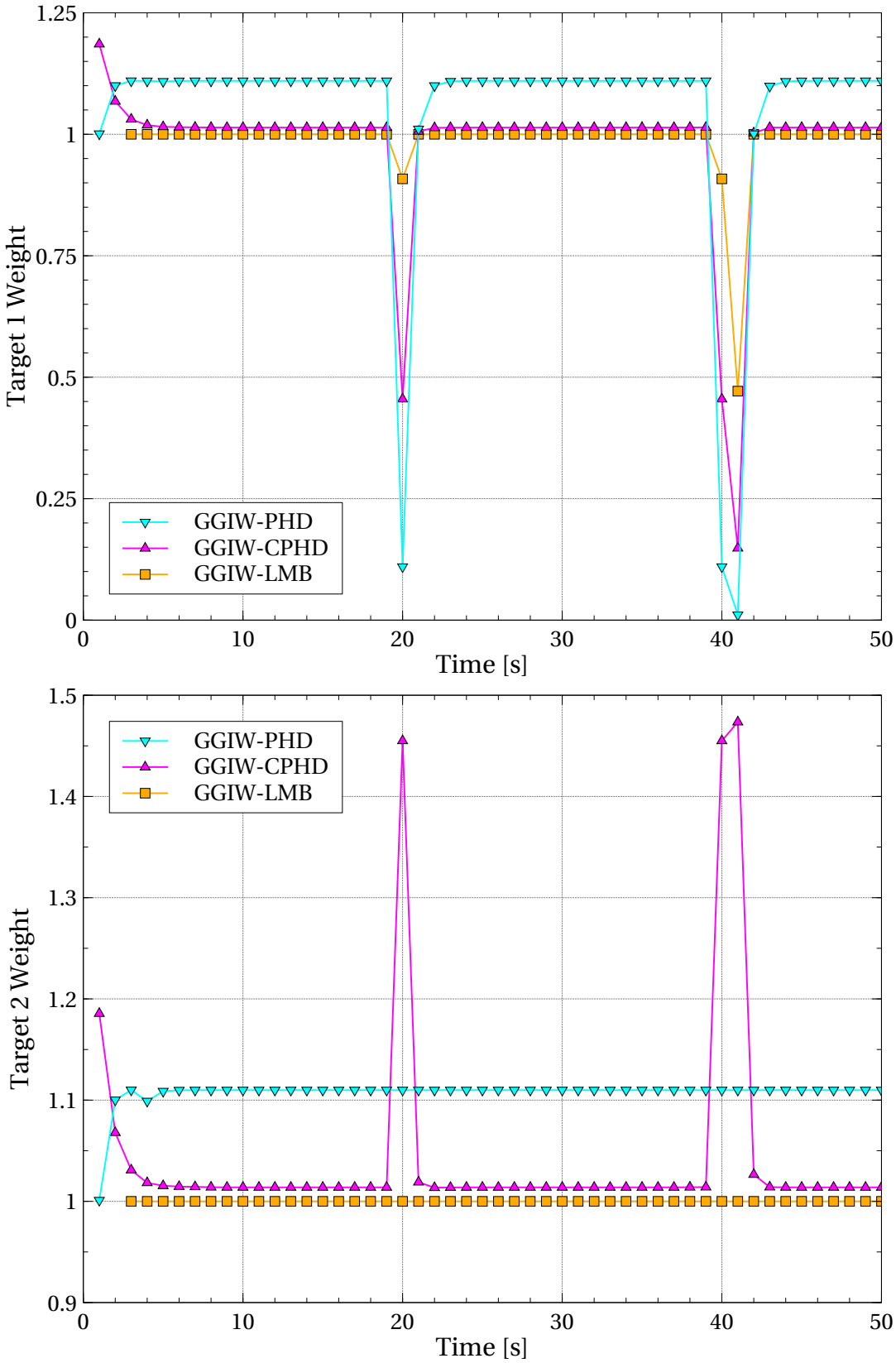


Figure 38: Results for extended-target test scenario 3. The lines show the estimated weights for two targets, which are approximately 1km away from each other. Target 1 is misdetected at time steps 20, 40 and 41. Note that the weight of target 2 in the GGIW-CPHD filter increases when this occurs, but this does not happen in the GGIW-LMB filter.

4.8 EXPERIMENTAL RESULTS

To demonstrate the proposed method on a real world scenario, the GGIW-LMB filter is applied to pedestrian tracking with laser range-finders[3]. In contrast to targets with distinct shapes, such as vehicles or buildings, pedestrians do not exhibit specific structure in laser scans, appearing instead as a random cluster of points. Hence, the GGIW measurement model is well suited to this application, since it assumes that the measurements are distributed normally around the target centroid. For this experiment, two pedestrians were recorded while walking on a parking lot using three Ibeo Lux laser sensors, which are mounted in the front bumper of the vehicle. Before being passed to the tracking filter, the laser returns from each sensor were thinned by removing measurements that lie outside the region of interest, thus excluding measurements from parked vehicles. The sensor was stationary during the experiment, and both pedestrians were wandering around the surveillance region, and are in close proximity to each other mid-way through the scenario. Figure 39 shows camera footage from this instant.

Figure 39: Photograph of pedestrian tracking scenario. The two pedestrians are close together at around t = 6.5 seconds.

3 Thanks to Dr Stephan Reuter and Alexander Scheel from Ulm University, Germany, for providing the experimental data used in this section.

A constant velocity dynamic model is used to track the pedestrians, i.e., the parameters of (328) are $d = 2$ and

$$F_{k+1|k} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}, \quad Q_{k+1|k} = \sigma^2 \begin{bmatrix} \frac{T^4}{4} & \frac{T^3}{2} \\ \frac{T^3}{2} & T^2 \end{bmatrix} \otimes x_{\mathcal{E},k+1}, \tag{331}$$

where the sampling period is $T = 0.08\,s$ and the standard deviation of the process noise is $\sigma = 4\,m/s^2$. Due to the decreased dimension of the motion model, the measurement matrix in (288) is $H = \begin{bmatrix} 1 & 0 \end{bmatrix}$, and since the birth locations are unknown, the LMB filter with adaptive birth model is used in this scenario. Other filter parameters are similar to those used in Section 4.7.

The results from the GGIW-LMB-ab filter are depicted in Figure 40. The dashed lines show approximate ground truth trajectories, which were obtained by manually labelling the pedestrians in the raw laser scans, and the solid lines show the estimated trajectories. In addition, two-sigma ellipses representing the target extents as well as the corresponding measurements are also shown for selected time instants.

The filter is able to track both pedestrians continuously, even when they are very close together. Especially in this situation, the multi-target representation as (G)LMB facilitates finding consistent association hypotheses and maintaining tracks over time. Note that the strongly fluctuating measurements on different moving parts of the human body, such as legs, arms and torso, make precise estimation of the target centroid positions difficult, in both the manual labelling process, and for the tracker itself. This explains most of the deviation between labelled ground truth and estimated trajectories. The estimates of the pedestrian extent vary over time, as demonstrated by the changing ellipses in Figure 40. This is again due to fluctuating measurements, which can be attributed mostly to leg movement. When the targets are close to the sensors, and the laser range-finders provide detailed scans of the legs, a periodic adaption of the target extent following the motion of the legs with each stride could be observed.

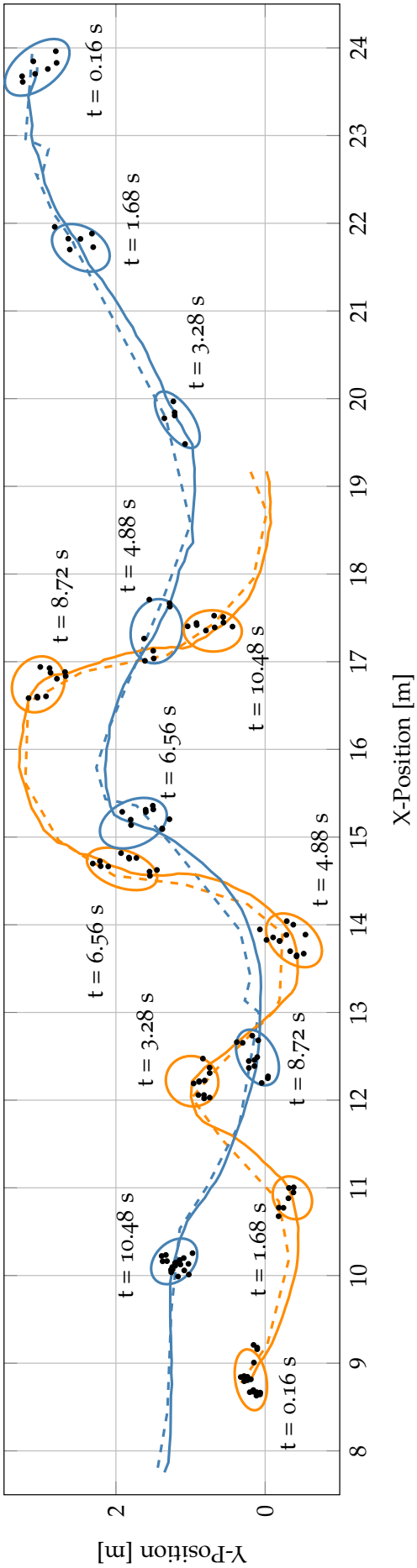


Figure 40: Ground truth (dashed) and estimated (solid) target trajectories in the pedestrian tracking scenario. Estimated two-sigma ellipses and corresponding laser measurements (black) are plotted for selected time steps.

4.9 PSEUDO-CODE FOR EXTENDED TARGET GLMB/LMB FILTERS

This section provides pseudo-code for the GGIW-GLMB and GGIW-LMB filters. The single-target prediction and update functions used in these algorithms are defined in Table 9.

Table 9: Single-target prediction and update functions used in the GGIW-(G)LMB pseudo-code.

Function	Description
`TargetPredict` (p, δ_t)	For a given GGIW density p at the current time, compute predicted GGIW density at δ_t seconds into the future, using equations (280), (281), (283), (284), (286) and (287)
`TargetUpdate`(p, W)	For a given prior GGIW density p, compute the posterior GGIW density conditioned on measurement set W, using equations (298)-(307).

4.9.1 *GGIW-GLMB Filter*

The complete recursion for the GGIW-GLMB filter is given in Algorithm 22, and the measurement update procedure is given in Algorithm 25. The method used for partitioning is summarised in Algorithms 23 and 24. In the partitioning functions, the notation $\min(D > 0)$ denotes finding the smallest non-zero value in the matrix D, the function `NthBestComponent` (Φ, n) returns the GLMB component in Φ with the n-th highest weight, and the function `ExpectationMaximisation` (Z, P) returns a clustering of the measurements in the set Z, computed using the EM algorithm, initialised on the extended-target estimates in component P.

Algorithm 22: GGIW-GLMB filter recursion.

Input: Measurements ($Z_{1:N}$), measurement times ($t_{1:N}$)
Output: Track estimates (τ_x), track labels (τ_ℓ)

Function $[\tau_x, \tau_\ell] = \texttt{GGIW\_GLMB\_Filter}\,(Z_{1:N}, t_{1:N})$ **is**
    $\ell_{next} = 1$;
    $\tau_x = \emptyset$;
    $\tau_\ell = \emptyset$;
    $P_{A,1} = \texttt{Zeros}\,(1, |Z_1|)$;
    for $k = 2 : N$ **do**
        $[\Omega_{B,k}, \ell_{next}] = \texttt{BirthGLMB}\,(Z_{k-1}, P_{A,k-1}, t_{k-1}, Z_k, t_k, \ell_{next})$;
        if $k = 2$ **then**
            $\Omega_+ = \Omega_{B,k}$;
        else
            $\Omega_{S,k} = \texttt{PredictSurvivalGLMB}\,(\Omega_{k-1}, t_k - t_{k-1})$;
            $\Omega_+ = \texttt{MultiplyGLMB}\,(\Omega_{S,k}, \Omega_{B,k})$;
        $\mathcal{P} = \texttt{FeasiblePartitions}\,(Z_k, \Omega_+)$;
        $[\Omega_k, P_{A,k}] = \texttt{UpdateGGIW\_GLMB}\,(\Omega_+, Z_k, \mathcal{P})$;
        $[\tau_x, \tau_\ell] = \texttt{ExtractEstimatesGLMB}\,(\Omega_k, \tau_x, \tau_\ell)$;
        $\Omega_k = \texttt{PruneGLMB}\,(\Omega_k)$;

Algorithm 23: Measurement set partitioning for the GGIW-(G)LMB filter.

Input: Measurement set (Z), predicted GLMB (Φ)
Parameters: Number of EM partitions (N_{em})
Output: List of partitions ($\mathcal{P}$)

Function $\mathcal{P} = \texttt{FeasiblePartitions}\,(Z, \Phi)$ **is**
    $\mathcal{P} = \texttt{DistancePartition}\,(Z)$;
    for $n = 1 : |N_{em}|$ **do**
        $P = \texttt{NthBestComponent}\,(\Phi, n)$;
        $\mathcal{U} = \texttt{ExpectationMaximisation}\,(Z, P)$;
        $\mathcal{P} = [\mathcal{P}; \mathcal{U}]$;

Algorithm 24: Distance partitioning algorithm.

Input: Measurement set (Z)
Parameters: Minimum distance (D_{min}), maximum distance (D_{max})
Output: List of partitions ($\mathcal{P}$)

Function $\mathcal{P} = \texttt{DistancePartition}(Z)$ **is**
  $n = |Z|$;
  for $i = 1 : n$ **do**
    for $j = 1 : n$ **do**
      $D(i,j) = \|Z_i - Z_j\|$;
  if $\min(D > 0) > D_{max}$ **then**
    $\mathcal{P}(1) = \{Z\}$;
  else if $\max(D) < D_{min}$ **then**
    $\mathcal{P}(1) = \{\{Z_1\}, \ldots, \{Z_n\}\}$;
  else
    $k = 1$;
    $\mathcal{P}(k) = \emptyset$;
    $d = \min(D > 0)$;
    while $d < D_{max}$ **do**
      $I = \{1, \ldots, n\}$;
      $i \sim \texttt{Uniform}(I)$;
      $Y = \{i\}$;
      $W = \{i\}$;
      $I = I - W$;
      while $|I| > 0$ **do**
        $J = \{j \in I | \min[\{D(j,k) \forall k \in W\}] < d\}$;
        if $|J| > 0$ **then**
          $Y = Y \cup J$;
          $W = J$;
          $I = I - J$;
          if $|I| = 0$ **then**
            $\mathcal{P}(k) = \mathcal{P}(k) \cup \{Z_Y\}$;
            for $i \in Y$ **do**
              for $j \in Y$ **do**
                $D(i,j) = 0$;
        else
          $\mathcal{P}(k) = \mathcal{P}(k) \cup \{Z_Y\}$;
          $i \sim \texttt{Uniform}(I)$;
          $Y = \{i\}$;
          $W = \{i\}$;
          $I = I - W$;
          if $|I| = 0$ **then**
            $\mathcal{P}(k) = \mathcal{P}(k) \cup \{Z_Y\}$;
      $k = k + 1$;
      $\mathcal{P}(k) = \emptyset$;
      $d = \min(D > 0)$;

Algorithm 25: GGIW-GLMB measurement update.

Input: Predicted GLMB (Φ), measurements (Z), measurement partitions ($\mathcal{P}$)
Parameters: Maximum number of updated components (N), detection probability (P_D), clutter intensity (λ), surveillance volume (V)
Output: Posterior GLMB (Ω), prob that each measurement is from a target (P_A)

Function $[\Omega, P_A] = \texttt{UpdateGGIW\_GLMB}(\Phi, Z, \mathcal{P})$ **is**
  for $i = 1 : |Z|$ **do**
    $\theta_i = \emptyset$;
  $N_{cdn} = \texttt{Allocate}\left(N, \texttt{Poisson}\left(1 : \Phi.M; \texttt{Mean}\left(\Phi.\rho^{(:)}\right)\right)\right)$;
  for $n = 1 : \Phi.M$ **do**
    $k = 1$;
    $N_{cmp} = \texttt{Allocate}\left(N_{cdn}(n), \Phi.w^{(n,:)}\right)$;
    for $i = 1 : \Phi.N_n$ **do**
      $N_{prt} = \texttt{Allocate}\left(N_{cmp}(i), \texttt{Uniform}(1 : |\mathcal{P}|)\right)$;
      for $j = 1 : |\mathcal{P}|$ **do**
        $\mathcal{U} = \mathcal{P}(j)$;
        $C = \texttt{CostMatrixAssign}\left(\mathcal{U}(Z), \Phi.p^{(n,i,:)}, P_D\right)$;
        $A = \texttt{Murty}\left(C, N_{prt}(j)\right)$;
        for $(a, c) \in A$ **do**
          $\Omega.w^{(n,k)} = \Phi.\rho^{(n)} \times \Phi.w^{(n,i)} \times e^{-c}$;
          for $l = 1 : n$ **do**
            if $a(l) > 0$ **then**
              $\Omega.p^{(n,k,l)} = \texttt{TargetUpdate}\left(\Phi.p^{(n,i,l)}, \mathcal{U}_{a(l)}(Z)\right)$;
              $\theta_{a(\ell)} = \left[\theta_{a(\ell)}; (n, j)\right]$;
            else
              $\Omega.p^{(n,k,l)} = \Phi.p^{(n,i,l)}$;
            $\Omega.\ell^{(n,k,l)} = \Phi.\ell^{(n,i,l)}$;
            $k = k + 1$;
  $\Omega = \texttt{NormaliseGLMB}(\Omega)$;
  for $i = 1 : |Z|$ **do**
    $P_A(i) = \texttt{Sum}(\Omega.w(\theta_i))$;

4.9.2 *GGIW-LMB Filter*

The GGIW-LMB filter recursion and measurement update routine are specified in Algorithms 26 and 27.

Algorithm 26: GGIW-LMB filter recursion.

Input: Measurements $(Z_{1:N})$, measurement times $(t_{1:N})$
Output: Track estimates (τ_x), track labels (τ_ℓ)

Function $[\tau_x, \tau_\ell] = \texttt{GGIW\_LMB\_Filter}\left(Z_{1:N}, t_{1:N}\right)$ **is**
  $\ell_{next} = 1$;
  $\tau_x = \emptyset$;
  $\tau_\ell = \emptyset$;
  $P_{A,1} = \texttt{Zeros}\left(1, |Z_1|\right)$;
  for $k = 2 : N$ **do**
    $\left[\tilde{\Omega}_{B,k}, \ell_{next}\right] = \texttt{BirthLMB}\left(Z_{k-1}, P_{A,k-1}, t_{k-1}, Z_k, t_k, \ell_{next}\right)$;
    if $k = 2$ **then**
      $\tilde{\Omega}_+ = \tilde{\Omega}_{B,k}$;
    else
      $\tilde{\Omega}_{S,k} = \texttt{PredictSurvivalLMB}\left(\tilde{\Omega}_{k-1}, t_k - t_{k-1}\right)$;
      $\tilde{\Omega}_+ = \texttt{JoinLMB}\left(\tilde{\Omega}_{S,k}, \tilde{\Omega}_{B,k}\right)$;
    $\mathcal{P} = \texttt{FeasiblePartitions}\left(Z, \Omega_+\right)$;
    $\left[\tilde{\Omega}_k, P_{A,k}\right] = \texttt{UpdateGGIW\_LMB}\left(\tilde{\Omega}_+, Z_k, \mathcal{P}\right)$;
    $\left[\tau_x, \tau_\ell\right] = \texttt{ExtractEstimatesLMB}\left(\tilde{\Omega}_k, \tau_x, \tau_\ell\right)$;

Algorithm 27: GGIW-LMB measurement update.

Input: Predicted LMB density $(\tilde{\Phi})$, measurements (Z), feasible measurement partitions $(\mathcal{P})$
Output: Posterior LMB density $(\tilde{\Omega})$, probability that each measurement is from a target (P_A)

Function $\left[\tilde{\Omega}, P_A\right] = \texttt{UpdateGGIW\_LMB}\left(\tilde{\Phi}, Z, \mathcal{P}\right)$ **is**
  $C = \texttt{ClusterTracks}\left(\tilde{\Phi}, Z\right)$;
  $\tilde{\Omega} = \emptyset$;
  $P_A = \texttt{Zeros}\left(1, |Z|\right)$;
  for $W \in C$ **do**
    $\Phi_W = \texttt{LMBtoGLMB}\left(\tilde{\Phi}, W\right)$;
    $\left[\Omega_W, P_{A,W}\right] = \texttt{UpdateGGIW\_GLMB}\left(\Phi_W, Z, \mathcal{P}\right)$;
    $\tilde{\Omega}_W = \texttt{ApproximateLMB}\left(\Omega_W\right)$;
    $\tilde{\Omega} = \texttt{JoinLMB}\left(\tilde{\Omega}, \tilde{\Omega}_W\right)$;
    $P_A = P_A + P_{A,W}$;

5 SENSOR CONTROL FOR MULTI-TARGET TRACKING

As demonstrated in previous chapters, the generalised labeled multi-Bernoulli (GLMB) is a family of tractable models that affords great deal of flexibility for tackling multi-target tracking problems. In this chapter, we derive closed form expressions for the void probability functional and the Cauchy-Schwarz divergence for GLMB models, with a view to applying them to the complementary problem of multi-target sensor control. The proposed analytic void probability functional is a necessary and sufficient statistic that uniquely characterises a GLMB, while the proposed analytic Cauchy-Schwarz divergence provides a tractable measure of similarity between GLMBs. We demonstrate the use of both results on a sensor control problem, formulated as a constrained partially observed Markov decision process (POMDP). Within the constrained POMDP, the reward is based on the Cauchy-Schwarz divergence, and the constraint is based on the void probability. This work has been presented in the author's journal submission [22], which is currently under review, and partially in the author's conference paper [23].

This chapter is organized as follows. After a brief introduction in Section 5.1, Section 5.2 provides some necessary background on random finite sets, void probabilities and the Cauchy-Schwarz divergence. In Section 5.3 we describe the GLMB point process model and its properties, including analytic expressions for the void probability and Cauchy-Schwarz divergence. In Section 5.4 we present an application of the new results, by developing a solution to a sensor control problem for multi-target tracking.

5.1 INTRODUCTION

Point patterns are ubiquitous in nature, for example the states of objects in multi-object systems such as the coordinates of molecules in a liquid/crystal, trees in a forest, stars in a galaxy and so on [43, 181, 141]. Point processes (specifically simple finite point processes, or random finite sets) are probabilistic models for point patterns, derived from stochastic geometry - the study of random geometrical objects ranging from collections of points to arbitrary closed sets [181, 140]. Point process theory provides the tools for characterising the underlying laws of the point patterns and entails a diverse range of application areas, such as forestry [180], geology [155], biology [137, 91], physics [176], computer vision [8, 82], wireless networks [7, 72, 73], communications [28, 4], multi-target tracking [126, 136], and robotics [145, 116, 46].

In addition to the probability distribution, the *void-probability functional* (or simply *void probabilities*) is another fundamental descriptor of a point process [43, 181, 141]. The void probability on a given region is the probability that it contains no points of the point process. Rényi's celebrated theorem states that the probability law of a simple point process is uniquely determined by the void probabilities on the bounded Borel sets [43, 181, 141]. Analytic expressions for the void probabilities are available for point processes such as Poisson and independent and identically distributed (i.i.d.) cluster. In general, the void probabilities constitute an intuitive and powerful descriptor that also characterises the more general random closed sets via Choquet's capacity theorem [138, 140].

Apart from characterising point processes, measuring their similarities/differences is essential in the study of point patterns. Information-based divergences are fundamental in the statistical analysis of random variables [42], and divergences such as Kullback-Leibler, Rényi, Csiszár-Morimoto (or Ali-Silvey), and Cauchy-Schwarz have been developed for point processes [136, 134, 128, 169, 77, 170]. However, in general these divergences cannot be computed analytically. Arguably the most tractable of these is the Cauchy-Schwarz divergence, which admits a closed form expression for Poisson point processes. Indeed, the Cauchy-Schwarz divergence between two Poisson point processes is given by half the squared L_2-distance between their intensity functions [77]. Moreover, this result has also been extended to mixtures of Poisson point processes [77].

While the Poisson model enjoys many elegant analytical properties, it is too simplistic for problems such as multi-target tracking, and a more suitable model is the generalised labeled multi-Bernoulli (GLMB) [198]. For the standard multi-object system model, the family of GLMB densities is a conjugate prior that is also closed under the Chapman-Kolmogorov equation [198]. Thus, in tracking applications, the posterior density of the point process at each time epoch is a GLMB, which can be tractably computed using the algorithm developed in [195]. Recent applications to problems in multi-object systems such as multiple target tracking, sensor management, and simultaneous localization and mapping [46, 164, 21, 65, 157], suggest that the GLMB is a versatile model that offers good trade-offs between tractability and fidelity.

In this chapter, we derive a closed-form void-probability functional and Cauchy-Schwarz divergence for GLMBs. Given the theoretical significance of these results, their derivations are remarkably simple. Moreover, these results provide additional tools to tackle more complex problems in multi-object systems. To demonstrate the use of both results, we develop a principled solution to an observer trajectory optimisation problem for multi-target tracking, where the goal is to obtain the most accurate estimates for an unknown and time-varying number of targets, whilst maintaining a safe distance from all of the targets. Using GLMBs to model the collection of targets, we formulate the problem as a constrained partially observed Markov decision process, with a Cauchy-Schwarz divergence based reward function and a void-probability based constraint, both of which can be evaluated analytically.

5.2 BACKGROUND

A *random finite set* (RFS) X, defined on the space $\mathbb{X}$, is a random variable taking values in $\mathcal{F}(\mathbb{X})$, i.e. a finite set-valued random variable. Both the number of elements in an RFS and their values are random. There are several constructs for specifying the probability law of an RFS. The most convenient of these for the exposition of the two main results of this chapter is the *belief* (or *containment*) *functional* B, given by

$$B(S) = \Pr(X \subseteq S) \tag{332}$$

for any (closed) $S \subseteq \mathbb{X}$ [181, 140, 136, 138]. In fact, the belief functional uniquely determines the probability law of a random closed set (and hence an RFS) via Choquet's capacity theorem.

5.2.1 *Void Probability Functional*

The void probability of an RFS X on a (compact) subset $S \subseteq \mathbb{X}$ is the probability that S contains no points of X, i.e. $\Pr(X \cap S = \emptyset)$, or equivalently the probability that X is contained in the complement of S, i.e. $\Pr(X \subseteq \mathbb{X} - S)$ [181, 140]. Thus, in terms of the belief functional the *void probability* (or *avoidance*) *functional* Q is given by

$$Q(S) = B(\mathbb{X} - S). \tag{333}$$

As a consequence of Choquet's capacity theorem [140, 138], the probability law of an RFS is uniquely defined by the void probability functional. Rényi also established, using a different line of argument, that the law of a simple point process is uniquely determined by the void probabilities on the bounded Borel sets of $\mathbb{X}$ [43, 181, 141].

The void probability functional is a more intuitive descriptor of an RFS than its probability distribution. Void probabilities on different regions provide a sense of how the number and locations of the points in an RFS are distributed across the state space. The concept of void probability is also directly applicable to the more general class of random closed sets. Indeed, Choquet's capacity theorem implies that the void probabilities uniquely determine the probability law of a random closed set [181, 140, 138]. Note that the void probability on a given region for the union of two independent RFSs is simply the product of their individual void probabilities.

5.2.2 *Cauchy-Schwarz Divergence*

Geometrically, the Cauchy-Schwarz divergence determines the information "difference" between random variables, from the angle subtended by their probability densities [88, 89]. Algebraically, it is based on the Cauchy-Schwarz inequality for the inner product between the probability densities of the random variables.

In [77], the Cauchy-Schwarz divergence was extended to random finite sets via the inner product of their probability densities relative to the reference measure μ, as defined by equation (48). Using the relationship between probability density and

belief density, which was discussed in Section 2.2.1, the Cauchy-Schwarz divergence between two RFSs, with respective belief densities ϕ and φ, is given by

$$D_{CS}(\phi,\varphi) = -\ln \frac{\int K^{|X|}\phi(X)\,\varphi(X)\,\delta X}{\sqrt{\int K^{|X|}\phi^2(X)\,\delta X \int K^{|X|}\varphi^2(X)\,\delta X}}. \tag{334}$$

Note that $D_{CS}(\phi,\varphi)$ is invariant to the unit of hyper-volume K (using the same line of argument as in [77, Section III-A]).

5.2.3 *Poisson Point Process*

The *intensity function* of an RFS X, is a non-negative function v (on $\mathbb{X}$) whose integral over any (Borel) $S \subseteq \mathbb{X}$ gives the expected number of elements of the RFS that are in S [43, 181, 141], i.e.

$$\mathbb{E}[|X \cap S|] = \langle 1_S, v\rangle. \tag{335}$$

Since $\langle 1_S, v\rangle$ is the expected number of points of X in the region S, the intensity value $v(x)$ can be interpreted as the instantaneous expected number of points per unit hyper-volume at x. Thus, in general, $v(x)$ is not dimensionless, but has units of K^{-1}. The intensity function is the first moment of an RFS, and can be computed from the belief density π by [126, 136]

$$v(x) = \int \pi(\{x\} \cup X)\,\delta X. \tag{336}$$

A *Poisson* point process is completely characterized by its intensity function v. The cardinality of a Poisson point process is Poisson distributed with mean $\langle 1, v\rangle$, and conditional on the cardinality, its elements are independently and identically distributed (i.i.d.) according to the probability density $v(\cdot) / \langle 1, v\rangle$ [43, 181, 141]. It is implicit that $\langle 1, v\rangle$ is finite, since we only consider simple-finite point processes.

The void probability functional and belief density of a Poisson point process with intensity function v are given, respectively, by [181, 141, 126, 136]

$$Q(S) = e^{-\langle 1_S, v\rangle}, \tag{337}$$

$$\pi(X) = e^{-\langle 1, v\rangle} v^X. \tag{338}$$

Moreover, the Cauchy-Schwarz divergence between two Poisson point processes is given by half the squared L^2-distance between their intensity functions [77]. As a consequence, the Bhattacharyya distance between the probability distributions of two Poisson point processes is the squared Hellinger distance between their respective intensity measures. For Gaussian mixture intensity functions, the Cauchy-Schwarz divergence can be evaluated analytically. These results were also extended to mixtures of Poisson point processes [77].

5.3 GENERALISED LABELLED MULTI-BERNOULLI

The Poisson point process is endowed with many elegant mathematical properties [43, 181, 103], including analytic void probabilities and Cauchy-Schwarz divergence, but it is rather simplistic for many practical problems. Tracking multiple targets based on observed data is a fundamental problem that arises in multi-object systems, with applications spanning several disciplines. For most data models, the posterior distributions of the underlying point processes are not Poisson [126, 136]. Although Poisson approximations, such as probability hypothesis density (PHD) filters, are numerically attractive [126], the Poisson model can neither capture the dependence between the points, nor permit the inference of the trajectories of the points over time.

The generalised labelled multi-Bernoulli (GLMB) is a class of tractable models that can be used in multi-target tracking, which alleviates the limitations of the Poisson model [198, 195]. Although sophisticated models in the spatial point process literature such as Cox, Neyman-Scott, Gauss-Poisson, Markov (or Gibbs) [43, 181, 141], are able to accommodate interactions such as repulsion, attraction or clustering, they cannot capture exactly the general inter-point dependencies in the posterior distribution that transpires through the data. In other words, they are not conjugate with respect to the standard multi-object measurement likelihood function. Moreover, these models are neither amenable to on-line computation, nor to the inference of trajectories.

In this section, we revisit the GLMB model [198] and some of its analytical properties. In addition, we present closed form expressions for the void probability functional and the Cauchy-Schwarz divergence for the GLMB.

5.3.1 *GLMBs and their Properties*

A GLMB is a labeled RFS with belief density on $\mathcal{F}(\mathbb{X}\times\mathbb{L})$ of the form

$$\boldsymbol{\pi}(\mathbf{X}) = \Delta(\mathbf{X}) \sum_{c\in\mathbb{C}} w^{(c)}(\mathcal{L}(\mathbf{X})) \left[p^{(c)}\right]^{\mathbf{X}}, \tag{339}$$

where $\Delta(\mathbf{X}) \triangleq \delta_{|\mathbf{X}|}(|\mathcal{L}(\mathbf{X})|)$ is the *distinct label indicator*, $\mathbb{C}$ is a discrete and finite index set, each $p^{(c)}(\cdot,\ell)$ is a probability density on $\mathbb{X}$, and each $w^{(c)}(L)$ is non-negative with

$$\sum_{L\subseteq\mathbb{L}} \sum_{c\in\mathbb{C}} w^{(c)}(L) = 1. \tag{340}$$

By convention, the functions $p^{(c)}(x,\ell)$ are measured in units of K^{-1}, and consequently, $\boldsymbol{\pi}(\boldsymbol{X})$ has units of $K^{-|X|}$. The belief density (339) is a mixture of multi-object exponentials, with each component consisting of a weight $w^{(c)}(\mathcal{L}(\mathbf{X}))$ that depends only on the labels of $\mathbf{X}$, and a multi-object exponential $\left[p^{(c)}\right]^{\mathbf{X}}$. Such a structure provides the flexibility for the GLMB to capture the dependence between points that transpires via the data, and also admits a number of convenient analytical properties, which are summarised as follows.

- For the standard multi-object system model that accounts for thinning, Markov shifts and superposition, the GLMB family is a conjugate prior, and is also closed under the Chapman-Kolmogorov equation [198].

- The GLMB density can be approximated to any L_1-norm error by truncating components [195]. More precisely, let us explicitly denote the dependence on the index set $\mathbb{C}$ of a (possibly unnormalised) GLMB density by

$$\mathbf{f}_{\mathbb{C}}(\mathbf{X}) = \Delta(\mathbf{X}) \sum_{c\in\mathbb{C}} w^{(c)}(\mathcal{L}(\mathbf{X})) \left[p^{(c)}\right]^{\mathbf{X}}, \tag{341}$$

  and let $\|\mathbf{f}\|_1 \triangleq \int |\mathbf{f}(\mathbf{X})|\,\delta\mathbf{X}$ denote the L_1-norm of $\mathbf{f}:\mathcal{F}(\mathbb{X}\times\mathbb{L})\to\mathbb{R}$. If $\mathbb{D}\subseteq\mathbb{C}$ then

$$||\mathbf{f}_{\mathbb{C}} - \mathbf{f}_{\mathbb{D}}||_1 = \sum_{c\in\mathbb{C}-\mathbb{D}} \sum_{L\subseteq\mathbb{L}} w^{(c)}(L). \tag{342}$$

- The cardinality distribution and intensity function of a GLMB are respectively given by

$$\Pr\left(|X|=n\right) = \sum_{c\in\mathbb{C}} \sum_{L\subseteq\mathbb{L}} \delta_n\left[|L|\right] w^{(c)}\left(L\right), \tag{343}$$

$$v\left(x,\ell\right) = \sum_{c\in\mathbb{C}} p^{(c)}\left(x,\ell\right) \sum_{L\subseteq\mathbb{L}} 1_L\left(\ell\right) w^{(c)}\left(L\right). \tag{344}$$

- The GLMB is flexible enough to approximate any labelled RFS density, by matching the intensity function and cardinality distribution. Furthermore, there is a simple closed form that minimises the Kullback-Leibler divergence between the labelled RFS density and its GLMB approximation [157].

As shown above, the GLMB family possesses some useful analytical properties. There is also an elegant characterisation of the GLMB using the probability generating functional (p.g.fl.) by Mahler [136]. In the following subsection, we derive two additional properties of the GLMB, which have some potentially useful applications.

5.3.2 *Void Probability Functional*

Proposition 15. *For a GLMB with belief density $\boldsymbol{\pi}$ of the form (339), the void probability functional is given by*

$$Q_\pi\left(S\right) = \sum_{L\subseteq\mathbb{L}} \sum_{c\in\mathbb{C}} w^{(c)}\left(L\right) \prod_{\ell\in L} \left\langle 1-1_S\left(\cdot\right), p^{(c)}\left(\cdot,\ell\right)\right\rangle. \tag{345}$$

Proof. Using (333) and (46), the void probability functional can be expressed as

$$Q_{\boldsymbol{\pi}}\left(S\right) = \int_{\mathbb{X}-S} \boldsymbol{\pi}\left(X\right)\delta X \tag{346}$$

$$= \int_{\mathbb{X}-S} \Delta\left(\boldsymbol{X}\right) \sum_{c\in\mathbb{C}} w^{(c)}\left(\mathcal{L}\left(\boldsymbol{X}\right)\right) \left[p^{(c)}\left(\cdot\right)\right]^{\boldsymbol{X}} \delta\boldsymbol{X}. \tag{347}$$

Applying Lemma 3 from [198], yields the result

$$Q_{\boldsymbol{\pi}}\left(S\right) = \sum_{L\subseteq\mathbb{L}} \sum_{c\in\mathbb{C}} w^{(c)}\left(L\right) \left[\int_{\mathbb{X}-S} p^{(c)}\left(x,\cdot\right)dx\right]^L \tag{348}$$

$$= \sum_{L\subseteq\mathbb{L}} \sum_{c\in\mathbb{C}} w^{(c)}\left(L\right) \prod_{\ell\in L} \left\langle 1-1_S\left(\cdot\right), p^{(c)}\left(\cdot,\ell\right)\right\rangle. \tag{349}$$

□

In simple cases, the inner product $\left\langle 1 - 1_S, p^{(c)}(\cdot, \ell) \right\rangle$ may be computable in closed form, in which case the void probability has an exact analytical solution. However, in more general cases, a closed form may not exist, and it must therefore be computed using numerical methods such as cubature or Monte Carlo integration. This yields an approximation to the true void probability for the GLMB.

In general, the computational complexity is $\mathcal{O}(N + M)$, where N is the number of pairs $(c, L) \in \mathbb{C} \times \mathcal{F}(\mathbb{L})$ such that $w^{(c)}(L) \neq 0$, and M is the number of unique single-object densities $p^{(c)}(\cdot, \ell)$. In many applications, the GLMB may contain a large number of single-object densities which are common across many elements of the sum, in which case the inner product only needs to be computed once for each unique single-object density. The general method to calculate the void probability for a GLMB over a region S is given in Algorithm 28.

Algorithm 28: GLMB void probability calculation.

Input: GLMB density (Φ), region (S)
Output: Void probability (P_{void})

Function $P_{void} = \texttt{VoidProbabilityGLMB}(\Phi, S)$ **is**
  $P_{void} = 0$;
  for $n = 1 : \Phi.M$ **do**
    for $i = 1 : \Phi.N_n$ **do**
      $c = \Phi.\rho^{(n)} \times \Phi.w^{(n,i)}$;
      for $j = 1 : n$ **do**
        $c = c \times \left[1 - \texttt{Integrate}\left(\Phi.p^{(n,i,j)}, S\right)\right]$;
      $P_{void} = P_{void} + c$;

The analytic expression for the GLMB void probability functional is of theoretical interest in itself, since it provides an alternative means of completely specifying a GLMB point process. However, it also holds significant practical interest, since it can be used to compute statistics that can conceivably be applied in a wide range of real-world problems.

In multi-object estimation and control, the GLMB void probability functional could supply useful information that can be applied in tasks such as trajectory planning (e.g. for collision avoidance), sensor management (e.g. focussing sensor resources on regions where targets are likely to be present), or the provision of situational awareness (e.g. advance warning of possible collisions between objects). The GLMB is a flexible model which has been used to develop algorithms for target tracking [198, 195, 163] and simultaneous localization and mapping [46]. Indeed, the GLMB

has been applied in autonomous vehicle systems [163], where trajectory planning and situational awareness for collision avoidance are paramount. It has also been applied to the tracking of orbital space debris [93, 94], for which the scheduling and management of observation equipment is a significant issue, as well as the planning of satellite trajectories to minimize the probability of collision with tracked debris.

5.3.3 *Cauchy-Schwarz Divergence*

Using the definition in equation (334), we show that the Cauchy-Schwarz divergence between two GLMBs can be expressed in closed form.

Proposition 16. *For two GLMBs with belief densities*

$$\boldsymbol{\phi}(\boldsymbol{X}) = \Delta(\boldsymbol{X}) \sum_{c\in\mathbb{C}} w_{\boldsymbol{\phi}}^{(c)}(\mathcal{L}(\boldsymbol{X})) \left[p_{\boldsymbol{\phi}}^{(c)}(\cdot)\right]^{\boldsymbol{X}}, \tag{350}$$

$$\boldsymbol{\psi}(\boldsymbol{X}) = \Delta(\boldsymbol{X}) \sum_{d\in\mathbb{D}} w_{\boldsymbol{\psi}}^{(d)}(\mathcal{L}(\boldsymbol{X})) \left[p_{\boldsymbol{\psi}}^{(d)}(\cdot)\right]^{\boldsymbol{X}}, \tag{351}$$

where both $p_{\boldsymbol{\phi}}^{(c)}$ and $p_{\boldsymbol{\psi}}^{(d)}$ are measured in units of K^{-1}, the Cauchy-Schwarz divergence between $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$ is given by

$$D_{CS}(\boldsymbol{\phi},\boldsymbol{\psi}) = -\ln\left(\frac{\langle\boldsymbol{\phi},\boldsymbol{\psi}\rangle_K}{\sqrt{\langle\boldsymbol{\phi},\boldsymbol{\phi}\rangle_K\langle\boldsymbol{\psi},\boldsymbol{\psi}\rangle_K}}\right), \tag{352}$$

where

$$\begin{aligned}
\langle\boldsymbol{\phi},\boldsymbol{\psi}\rangle_K &= \sum_{L\subseteq\mathbb{L}}\sum_{c\in\mathbb{C}}\sum_{d\in\mathbb{D}} w_{\boldsymbol{\phi}}^{(c)}(L)\, w_{\boldsymbol{\psi}}^{(d)}(L) \prod_{\ell\in L} K\left\langle p_{\boldsymbol{\phi}}^{(c)}(\cdot,\ell), p_{\boldsymbol{\psi}}^{(d)}(\cdot,\ell)\right\rangle \\
\langle\boldsymbol{\phi},\boldsymbol{\phi}\rangle_K &= \sum_{L\subseteq\mathbb{L}}\sum_{c\in\mathbb{C}}\sum_{d\in\mathbb{C}} w_{\boldsymbol{\phi}}^{(c)}(L)\, w_{\boldsymbol{\phi}}^{(d)}(L) \prod_{\ell\in L} K\left\langle p_{\boldsymbol{\phi}}^{(c)}(\cdot,\ell), p_{\boldsymbol{\phi}}^{(d)}(\cdot,\ell)\right\rangle \\
\langle\boldsymbol{\psi},\boldsymbol{\psi}\rangle_K &= \sum_{L\subseteq\mathbb{L}}\sum_{c\in\mathbb{D}}\sum_{d\in\mathbb{D}} w_{\boldsymbol{\psi}}^{(c)}(L)\, w_{\boldsymbol{\psi}}^{(d)}(L) \prod_{\ell\in L} K\left\langle p_{\boldsymbol{\psi}}^{(c)}(\cdot,\ell), p_{\boldsymbol{\psi}}^{(d)}(\cdot,\ell)\right\rangle
\end{aligned} \tag{353}$$

Proof. If $\boldsymbol{\phi}(\boldsymbol{X})$ and $\boldsymbol{\psi}(\boldsymbol{X})$ are two GLMBs defined by (350) and (351), the inner product is given by

$$\begin{aligned}
\langle\boldsymbol{\phi},\boldsymbol{\psi}\rangle_K &= \int K^{|\boldsymbol{X}|}\boldsymbol{\phi}(\boldsymbol{X})\,\boldsymbol{\psi}(\boldsymbol{X})\,\delta\boldsymbol{X} \\
&= \int K^{|\boldsymbol{X}|}\Delta(\boldsymbol{X}) \sum_{c\in\mathbb{C}} w_{\boldsymbol{\phi}}^{(c)}(\mathcal{L}(\boldsymbol{X})) \left[p_{\boldsymbol{\phi}}^{(c)}(\cdot)\right]^{\boldsymbol{X}} \sum_{d\in\mathbb{D}} w_{\boldsymbol{\psi}}^{(d)}(\mathcal{L}(\boldsymbol{X})) \left[p_{\boldsymbol{\psi}}^{(d)}(\cdot)\right]^{\boldsymbol{X}} \delta\boldsymbol{X} \\
&= \int \Delta(\boldsymbol{X}) \sum_{c\in\mathbb{C}}\sum_{d\in\mathbb{D}} w_{\boldsymbol{\phi}}^{(c)}(\mathcal{L}(\boldsymbol{X}))\, w_{\boldsymbol{\psi}}^{(d)}(\mathcal{L}(\boldsymbol{X})) \left[K p_{\boldsymbol{\phi}}^{(c)}(\cdot)\, p_{\boldsymbol{\psi}}^{(d)}(\cdot)\right]^{\boldsymbol{X}} \delta\boldsymbol{X}.
\end{aligned} \tag{354}$$

Making use of [198, Lemma 3], this becomes

$$\begin{aligned}\left\langle \boldsymbol{\phi}, \boldsymbol{\psi} \right\rangle_K &= \sum_{L \subseteq \mathbb{L}} \sum_{c \in \mathbb{C}} \sum_{d \in \mathbb{D}} w_{\boldsymbol{\phi}}^{(c)}(L)\, w_{\boldsymbol{\psi}}^{(d)}(L) \left[K \int p_{\boldsymbol{\phi}}^{(c)}(x, \cdot)\, p_{\boldsymbol{\psi}}^{(d)}(x, \cdot)\, dx \right]^L \\ &= \sum_{L \subseteq \mathbb{L}} \sum_{c \in \mathbb{C}} \sum_{d \in \mathbb{D}} w_{\boldsymbol{\phi}}^{(c)}(L)\, w_{\boldsymbol{\psi}}^{(d)}(L) \prod_{\ell \in L} K \left\langle p_{\boldsymbol{\phi}}^{(c)}(\cdot, \ell), p_{\boldsymbol{\psi}}^{(d)}(\cdot, \ell) \right\rangle \end{aligned} \tag{355}$$

and similarly for $\left\langle \boldsymbol{\phi}, \boldsymbol{\phi} \right\rangle_K$ and $\left\langle \boldsymbol{\psi}, \boldsymbol{\psi} \right\rangle_K$. Substituting these into (334), yields the result (352)-(353). □

In cases where the inner product between two single-object densities $p_{\boldsymbol{\phi}}^{(c)}$ and $p_{\boldsymbol{\psi}}^{(d)}$ of the GLMBs has an analytical solution, then the Cauchy-Schwarz divergence between the two GLMBs can also be evaluated analytically. However, where this is not possible, numerical approximations can be used to evaluate the inner products. The common case in which the single-object densities are Gaussian mixtures, does admit an analytical solution, as established in the following proposition.

Proposition 17. *Let $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$ be two GLMBs of the form (350) and (351) in which the single-object densities are Gaussian mixtures, i.e.*

$$p_{\boldsymbol{\phi}}^{(c)}(x, \ell) = \sum_{i=1}^{N_{\boldsymbol{\phi}}^{(c)}} \omega_{\boldsymbol{\phi},i}^{(c)}(\ell)\, \mathcal{N}\left(x; m_{\boldsymbol{\phi},i}^{(c)}(\ell), P_{\boldsymbol{\phi},i}^{(c)}(\ell)\right), \tag{356}$$

$$p_{\boldsymbol{\psi}}^{(d)}(x, \ell) = \sum_{i=1}^{N_{\boldsymbol{\psi}}^{(d)}} \omega_{\boldsymbol{\psi},i}^{(d)}(\ell)\, \mathcal{N}\left(x; m_{\boldsymbol{\psi},i}^{(d)}(\ell), P_{\boldsymbol{\psi},i}^{(d)}(\ell)\right), \tag{357}$$

where both $p_{\boldsymbol{\phi}}^{(c)}$ and $p_{\boldsymbol{\psi}}^{(d)}$ are measured in units of K^{-1}. The Cauchy-Schwarz divergence between $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$ can be expressed in analytical form, since $\left\langle \boldsymbol{\phi}, \boldsymbol{\psi} \right\rangle_K$ in (353) reduces to

$$\left\langle \boldsymbol{\phi}, \boldsymbol{\psi} \right\rangle_K = \sum_{L \subseteq \mathbb{L}} \sum_{c \in \mathbb{C}} \sum_{d \in \mathbb{D}} w_{\boldsymbol{\phi}}^{(c)}(L)\, w_{\boldsymbol{\psi}}^{(d)}(L) \left[\gamma_{\phi,\psi}\right]^L \tag{358}$$

where

$$\gamma_{\boldsymbol{\phi},\boldsymbol{\psi}}(\ell) = \sum_{i=1}^{N_{\boldsymbol{\phi}}^{(c)}} \sum_{j=1}^{N_{\boldsymbol{\psi}}^{(d)}} \omega_{\boldsymbol{\phi},i}^{(c)}(\ell)\, \omega_{\boldsymbol{\psi},j}^{(d)}(\ell)\, \mathcal{N}\left(m_{\boldsymbol{\phi},i}^{(c)}(\ell); m_{\boldsymbol{\psi},j}^{(d)}(\ell), P_{\boldsymbol{\phi},i}^{(c)}(\ell) + P_{\boldsymbol{\psi},j}^{(d)}(\ell)\right), \tag{359}$$

and similarly for $\left\langle \boldsymbol{\phi}, \boldsymbol{\phi} \right\rangle_K$ and $\left\langle \boldsymbol{\psi}, \boldsymbol{\psi} \right\rangle_K$.

Proof. Substituting (356) and (357) into the inner product in (353), gives

$$\left\langle p_{\boldsymbol{\phi}}^{(c)}(\cdot,\ell), p_{\boldsymbol{\psi}}^{(d)}(\cdot,\ell)\right\rangle = \sum_{i=1}^{N_{\boldsymbol{\phi}}^{(c)}}\sum_{j=1}^{N_{\boldsymbol{\psi}}^{(d)}} \omega_{\boldsymbol{\phi},i}^{(c)}(\ell)\,\omega_{\boldsymbol{\psi},j}^{(d)}(\ell) \times \int \mathcal{N}\left(x; m_{\boldsymbol{\phi},i}^{(c)}(\ell), P_{\boldsymbol{\phi},i}^{(c)}(\ell)\right)\mathcal{N}\left(x; m_{\boldsymbol{\psi},j}^{(d)}(\ell), P_{\boldsymbol{\psi},j}^{(d)}(\ell)\right)dx, \tag{360}$$

which is measured in units of K^{-1}. Applying the identity for a product of two Gaussians [161, pp. 200], and multiplying by K, we are left with the unitless quantity

$$K\left\langle p_{\boldsymbol{\phi}}^{(c)}(\cdot,\ell), p_{\boldsymbol{\psi}}^{(d)}(\cdot,\ell)\right\rangle = \gamma_{\boldsymbol{\phi},\boldsymbol{\psi}}(\ell), \tag{361}$$

where $\gamma_{\boldsymbol{\phi},\boldsymbol{\psi}}(\ell)$ is given by (359). Substituting (361) into (353), yields the result (358). □

Due to the nested summations in (353) or (358), a naive implementation will have a computational complexity of $\mathcal{O}\left(MN + M^2 + N^2\right)$, where M and N are the number of components[1] of $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$ with non-zero weights. This leads to the summations being taken only over pairs of components with matching label sets. It is therefore possible to reduce computation using associative data structures to facilitate finding these matching pairs. Although this does not reduce the worst case complexity, the average will be significantly better.

Note that if $w_{\boldsymbol{\phi}}^{(c)}(L) = 0$ or $w_{\boldsymbol{\psi}}^{(d)}(L) = 0$ for all triples $(L,c,d) \in \mathcal{F}(\mathbb{L}) \times \mathbb{C} \times \mathbb{D}$, then $\langle \boldsymbol{\phi}, \boldsymbol{\psi} \rangle_K$ will evaluate to zero, (i.e. $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$ are orthogonal) leading to a Cauchy-Schwarz divergence of infinity. This is an intuitive result as two such GLMBs have no non-zero components with matching labels. The general method for computing the Cauchy-Schwarz divergence between two GLMBs is given by Algorithm 29.

1 A GLMB $\boldsymbol{\phi}$ is completely parameterised by the set $\left\{\left(w_{\boldsymbol{\phi}}^{(c)}(L), p_{\boldsymbol{\phi}}^{(c)}\right) : (c,L) \in \mathbb{C} \times \mathcal{F}(\mathbb{L})\right\}$ and we refer to each element $\left(w_{\boldsymbol{\phi}}^{(c)}(L), p_{\boldsymbol{\phi}}^{(c)}\right)$ of this set as a component of a GLMB.

Algorithm 29: GLMB Cauchy-Schwarz divergence calculation.

Input: GLMB density (Φ), GLMB density (Ψ)
Output: Inner product (ψ)

Function $\psi = \texttt{InnerProductGLMB}\left(\Phi, \Psi\right)$ **is**
  $\psi = 0$;
  for $n = 1 : \texttt{min}\left(\Phi.M, \Psi.M\right)$ **do**
    for $i = 1 : \Phi.N_n$ **do**
      for $j = 1 : \Psi.N_n$ **do**
        if $\Phi.\ell^{(n,i,:)} = \Omega.\ell^{(n,j,:)}$ **then**
          $\gamma = \Phi.\rho^{(n)} \times \Psi.\rho^{(n)} \times \Phi.w^{(n,i)} \times \Psi.w^{(n,j)}$;
          for $k = 1 : n$ **do**
            $\gamma = \gamma \times \texttt{Integrate}\left(\Phi.p^{(n,i,k)} \times \Psi.p^{(n,j,k)}\right)$;
          $\psi = \psi + \gamma$;

Input: GLMB density (Φ), GLMB density (Ψ)
Output: Cauchy-Schwarz divergence (D)

Function $D = \texttt{CauchySchwarzGLMB}\left(\Phi, \Psi\right)$ **is**
  $\psi_{\Phi,\Phi} = \texttt{InnerProductGLMB}\left(\Phi, \Phi\right)$;
  $\psi_{\Omega,\Omega} = \texttt{InnerProductGLMB}\left(\Psi, \Psi\right)$;
  $\psi_{\Phi,\Omega} = \texttt{InnerProductGLMB}\left(\Phi, \Psi\right)$;
  $D = -\ln\left(\psi_{\Phi,\Psi} \times \psi_{\Phi,\Phi}^{-1/2} \times \psi_{\Psi,\Psi}^{-1/2}\right)$;

5.4 APPLICATION TO SENSOR MANAGEMENT

In this section, we apply the proposed closed-form solutions for the Cauchy-Schwarz divergence and void probabilities to a multi-target sensor management problem. In most target tracking scenarios, the sensor may perform various actions that can have a significant impact on the quality of the observed data, and can therefore influence the estimation performance of the tracking system. Typically, such actions might include changes in the position, orientation or motion of sensor platforms [175, 187, 156, 70], changes to sensor deployment and utilisation [188, 74], or altering the sensor operating parameters such as the beam pattern [112, 111], or transmit waveform [101, 154]. The control actions affect the information content of the received data, which in turn affects the system's ability to detect, track, and identify the targets.

Often, the control decisions are driven by manual intervention, which provides no guarantee of optimality. The goal of automatic sensor management is to determine the best control actions, based on some optimality criteria. This has the potential to improve tracking performance, by making control decisions in a systematic and optimal manner that accounts for the prevailing conditions.

5.4.1 *Problem Description*

In this application, the aim is to perform sensor control in the context of a multi-target tracking system that is based upon the standard models of multi-object dynamics and multi-object observations as defined in Sections 2.5.2 and 2.5.3. Although we concentrate on the standard model in this work, the same principles can be applied to other sensor models, such as those described in Chapters 3 and 4.

The quality of the observations (i.e. the detection probability and measurement noise) is dependent on the state of the objects and the sensor itself, for example, objects that are further away from the sensor generally have lower probability of detection and higher measurement noise. For this reason, the control of the sensor can have a significant influence on the tracking performance.

Here we address the problem of controlling the motion of a single sensor platform, with the aim of optimising the tracking performance under the aforementioned dynamic and observation models. Since the control actions affect the observation quality, the goal is to design a scheme which can automatically select control actions that

yield the most "informative" observations. This is a difficult problem due to the unknown and time-varying number of targets, and the uncertainty in the multi-object state due to the measurement noise, object detection/misdetection, and unknown measurement origin.

5.4.2 *Control Strategy*

We now proceed to formulate the control problem as a partially observable Markov decision processes (POMDP) [26, 142, 122, 35]. In general terms, the elements of a POMDP are as follows.

1. The system dynamics is a Markov process.
2. The observations follow a known distribution, conditioned on the state and the sensor control action.
3. The true state of the system is unknown, but we have access to the posterior probability density function (pdf) of the state conditioned on past observations.
4. The benefit of performing a given action can be expressed by a reward function, which characterises the objectives of the control system.

In this case, the dynamics is modelled by the transition density (169) and the observations are modelled by the likelihood function (173). The posterior pdf of the system state is modelled as a GLMB of the form (339), since it enables tractable estimation of object trajectories to inform the control strategy. In general, there are two broad categories of reward function that can be used in a POMDP, namely, "task-based" and "information-based" reward. Task-based reward functions (see for example [65]) are useful in situations where the control problem can be formulated in terms of a single well-defined objective. However, in situations where this does not exist, the information-based approach is more appropriate, as it strives to capture the information gain in an overall sense (for example [169]). In this example, the reward function is formulated in terms of an information divergence between prior and posterior multi-object densities, which bears a strong relationship to the improvement in estimation accuracy, since higher information divergence indicates greater information gain, which leads to more accurate track estimates.

It is also possible to place constraints on the control problem, which is useful in cases where it is foreseeable that certain actions might result in undesirable side-

effects. For example, to guarantee the safety or covertness of the sensor platform, we might want to ensure that no targets enter a predefined exclusion region around the sensor. This can be achieved using a constrained POMDP [84, 190], in which the goal is to find the control action that maximises the reward function, subject to one or more constraints.

Within the POMDP framework, the most computationally tractable strategy is to use myopic open-loop feedback control [26], with a discrete action space. The term "myopic" means that the algorithm only decides one control action at a time, rather than planning multiple actions into the future.

At the time that a control action is performed, we have no knowledge of the posterior density that would arise from taking that action. Since this precludes calculation of the true information divergence, its expectation with respect to all possible future measurements is taken [169, 127]. More precisely, let us begin by defining the following notation

- $\boldsymbol{\pi}_k(\cdot|Z_{1:k})$ is the posterior density at time k,
- $\mathbb{A}_k$ is a discrete space of control actions at time k,
- H is the length of the control horizon,
- $\boldsymbol{\pi}_{k+H}(\cdot|Z_{1:k})$ is the predicted density at time $k+H$ given measurements up to time k,
- $Z_{k+1:k+H}(\alpha)$ is the collection of measurement sets that would be observed from times $k+1$ up to $k+H$, if control action $\alpha \in \mathbb{A}_k$ was executed at time k,
- $V_k(\alpha)$ is the exclusion region around the sensor at time k under control action α,
- $Q_\pi(S)$ is the void probability functional corresponding to the multi-object density π over region S,
- P_{vmin} is the minimum void probability threshold.

The optimal control action is given by maximising the expected value of a reward function $\mathcal{R}_{k+H}(\cdot)$ over the space of allowable actions [169],

$$\alpha_{opt} = \arg\max_{\alpha \in \mathbb{A}_k} \mathbb{E}\left[\mathcal{R}_{k+H}(\alpha)\right], \tag{362}$$

subject to the constraint

$$\min_{i\in\{1,\ldots,H\}} \left[Q_{\pi_{k+i}(\boldsymbol{X}|Z_{1:k})} \left(V_i \left(\alpha \right) \right) \right] > P_{vmin}. \tag{363}$$

where the expectation is taken with respect to the future measurement sets $Z_{k+1:k+H}(\alpha)$.

In general, Monte Carlo integration is used to compute the expected reward in (362) because analytic solutions are not available. For each control action α, this involves drawing samples $Z^{(i)}_{k+1:k+H}(\alpha)$ for $i = 1, \ldots, N$, then computing the reward $\mathcal{R}^{(i)}_{k+H}(\alpha)$ conditioned on each sample. The samples $Z^{(i)}_{k+1:k+H}(\alpha)$ are obtained by first sampling from $\pi_k(\cdot|Z_{1:k})$ using Algorithm 11, then propagating each sample through the transition model up to the horizon time using Algorithm 1, and finally simulating a set of measurements from time $k+1$ to time $k+H$ using Algorithm 2. An estimate of the expected reward is given by the mean of the reward over all the samples,

$$\mathbb{E}\left[\mathcal{R}_{k+H}(\alpha) \right] \approx \frac{1}{N} \sum_{i=1}^{N} \mathcal{R}^{(i)}_{k+H}(\alpha). \tag{364}$$

Since we are sampling directly from the current distribution of $Z_{k+1:k+H}$, using the true transition and measurement models, this method converges to the true expectation of the reward as the number of samples is increased.

In (364), $\mathcal{R}^{(i)}_{k+H}(\alpha)$ is usually computed by Monte Carlo integration, for example [169]. Hence, the variance of the Monte Carlo estimate of the expected reward will depend on the number of samples used to calculate each $\mathcal{R}^{(i)}_{k+H}(\alpha)$, as well as the number of measurement samples N. On the other hand, a closed form expression for $\mathcal{R}^{(i)}_{k+H}(\alpha)$ would lead to a smaller variance in the estimate of the expected reward, by the principle of Rao-Blackwellisation [34].

The constraint (363) is the minimum value of the void probability up to the control horizon, where the value at time $k+i$ is computed based on the predicted density at that time, given measurements up to time k. The constraint is satisfied if this minimum value is greater than the threshold P_{vmin}.

5.4.3 *Reward Function*

For the reason discussed in Section 5.4.2, the existence of a closed form reward function is desirable in POMDPs. This would be particularly beneficial in this application, since the difference between the expected rewards of the various control actions can

be quite small, and may become obscured by the variance induced by the Monte Carlo estimation (364). Any reduction in this variance will clearly help in correctly identifying the optimal control action.

For the case of the GLMB, common information divergence measures such as the Kullback-Leibler or Rényi divergences cannot be expressed in analytical form. Thus, their use in this problem would require Monte Carlo integration, resulting in a higher variance in the expected reward, as well as increased computational load. To alleviate this, we use the Cauchy-Schwarz divergence between prior and posterior GLMB densities as the reward function, i.e.

$$\mathcal{R}_{k+H}^{(i)}(\alpha) = D_{CS}\left(\boldsymbol{\pi}_{k+H}(\boldsymbol{X}|Z_{1:k}), \boldsymbol{\pi}_{k+H}\left(\boldsymbol{X}|Z_{1:k}, Z_{k+1:k+H}^{(i)}(\alpha)\right)\right). \tag{365}$$

Algorithm 32 (see Section 5.4.6) demonstrates how to approximate the expected reward for each control action in a set $\mathbb{A}$.

5.4.4 *Constraint*

To enforce the constraint, we compute the void probability over an exclusion region around the sensor, for the predicted GLMB density at each time step up to the horizon. We use a circular exclusion region, centered at the sensor location, with radius r_V. Evaluation of the void probability requires integrating each single-object density in the GLMB over the exclusion region. For a 2-dimensional Gaussian pdf and a circular region, this does not have an analytical solution. Hence, we use adaptive cubature to approximate these integrals, before using them to compute the void probability.

The goal is to ensure that the separation between sensor and targets always exceeds r_V, i.e. the control action α is feasible if the constraint (363) is satisfied. That is, for each action we find the minimum void probability up to the horizon, and enforce the constraint that this minimum must exceed the threshold P_{vmin}, otherwise the action cannot be selected. Algorithm 33 (see Section 5.4.6) shows how to compute the void probability for a set of control actions.

5.4.5 *Simulation Results*

In this section, the control strategy is applied to the problem of observer trajectory optimisation for multi-target tracking. This application involves a single sensor that provides bearing and range measurements, where the noise on the measured bearings is constant for all targets, but the range noise is state-dependent, increasing as the true range between the sensor and target increases. The detection probability is also range-dependent, reducing as the range increases. Targets closer to the sensor are therefore detected with both higher probability and accuracy, and vice versa for targets that are further away.

For this problem, the state-dependency of the measurement noise and detection probability will be the main effect driving the control, and one would expect the algorithm to move the sensor towards the targets, in order to minimise noise and maximise the detection probability. However, in the presence of multiple targets, this can easily lead to conflicting control influences. The goal of the control algorithm is to resolve these conflicts, by attempting to provide a decision that optimises the multi-target estimation performance in an overall sense.

The target kinematics are modelled using 2D Cartesian position and velocity vectors $\tilde{x}_k = \begin{bmatrix} x_k^{(t)} & \dot{x}_k^{(t)} & y_k^{(t)} & \dot{y}_k^{(t)} \end{bmatrix}^T$, and they are assumed to move according to the following discrete white noise acceleration model,

$$\begin{aligned} \tilde{x}_k &= F\tilde{x}_{k-1} + \Gamma v_{k-1}, \\ F &= \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \otimes I_2, \qquad \Gamma = \begin{bmatrix} T^2/2 \\ T \end{bmatrix} \otimes I_2 \end{aligned} \tag{366}$$

where T is the sampling period, $v_{k-1} \sim \mathcal{N}(0, Q)$ is a 2×1 independent and identically distributed Gaussian process noise vector with $Q = \sigma_v^2 I_2$, where σ_v is the standard deviation of the target acceleration. The sensor measures the target bearing and range, where the measurement corresponding to a target state x_k and sensor position $u_k = \begin{bmatrix} x_k^{(s)} & y_k^{(s)} \end{bmatrix}$ at time k is given by

$$z_k = h(\tilde{x}_k, u_k) + w_k(\tilde{x}_k, u_k) \tag{367}$$

where $w_k(\tilde{x}_k, u_k) \sim \mathcal{N}\left(0, \text{diag}\left(\left[\begin{array}{cc} \sigma_\theta^2 & \sigma_r^2(\tilde{x}_k, u_k) \end{array}\right]\right)\right)$ is a 2×1 Gaussian measurement noise vector, and the measurement function h is given by

$$h(\tilde{x}_k, u_k) = \left[\begin{array}{c} \arctan\left(\frac{y_k^{(t)} - y_k^{(s)}}{x_k^{(t)} - x_k^{(s)}}\right) \\ \sqrt{\left(x_k^{(t)} - x_k^{(s)}\right)^2 + \left(y_k^{(t)} - y_k^{(s)}\right)^2} \end{array}\right]. \tag{368}$$

The variance of the bearing measurement noise σ_θ^2 is a fixed constant for all targets, but the variance of the range noise σ_r^2 is a function of the target and sensor states. In this example, we model the range noise in a piecewise manner as follows, in which $\mathcal{D}(\tilde{x}_k, u_k)$ denotes the true distance between a target with state $\tilde{x}_k$ and the sensor with state u_k,

$$\sigma_r^2(\tilde{x}_k, u_k) = \begin{cases} (\eta R_1)^2, & \mathcal{D}(\tilde{x}_k, u_k) \leq R_1 \\ (\eta \mathcal{D}(\tilde{x}_k, u_k))^2, & R_1 < \mathcal{D}(\tilde{x}_k, u_k) < R_2 \\ (\eta R_2)^2, & \mathcal{D}(\tilde{x}_k, u_k) \geq R_2 \end{cases}, \tag{369}$$

i.e. the noise standard deviation is the true range multiplied by the factor η, but the minimum is capped at ηR_1, and the maximum is capped at ηR_2. The detection probability is modeled using the following function of the true range,

$$P_D(\tilde{x}_k, u_k) = \frac{\mathcal{N}\left(\mathcal{D}(\tilde{x}_k, u_k); 0, \sigma_D^2\right)}{\mathcal{N}\left(0; 0, \sigma_D^2\right)}, \tag{370}$$

where σ_D controls the rate at which the detection probability drops off as the range increases.

To illustrate the performance of the control, we apply it to two different simulated scenarios. The first has a time-varying number of targets, and demonstrates how the algorithm adapts to the changing conditions over time. The second scenario consists of targets which are scattered in several different locations and moving in different directions. In scenario 1, the expected control behaviour is fairly clear from looking at the target-observer geometry. However, in scenario 2, the expected behaviour is not so obvious.

For both scenarios, the sensor sampling interval is $T = 10$s, the clutter rate is 100 per scan, the detection probability spread parameter is $\sigma_D = 20$km, the process noise on the target trajectories is $\sigma_v = 0.01\text{m/s}^2$, the bearing measurement noise is $\sigma_\theta = 2^\circ$, and the parameters of the range measurement noise are $R_1 = 1$km, $R_2 = 10$km

and $\eta = 0.1$. The space of possible control actions is discretised at 20° intervals, i.e. the allowed course changes are $\{-180°, -160°, \ldots, 0°, \ldots, 160°, 180°\}$. For the control calculations, the number of samples is $N = 50$, the sensor sampling interval is $T = 80s$, and the horizon length is $H = 5$ (i.e. the effective control lookahead is 400s). The exclusion radius for the void probability calculation is $r_v = 1$km, and the void probability threshold is $P_{vmin} = 0.95$.

5.4.5.1 *Scenario 1*

The first scenario runs for 4000 seconds and consists of seven targets, six of which enter the scene during the first 250s, with one more appearing at time 1700s. Three of the targets terminate between times 1300s and 1600s. The sensor platform is stationary for the first 400s, then starts moving with constant speed of 7m/s, undergoing course changes every 400s in order to improve the target estimates. The true target trajectories are depicted in Figure 41.

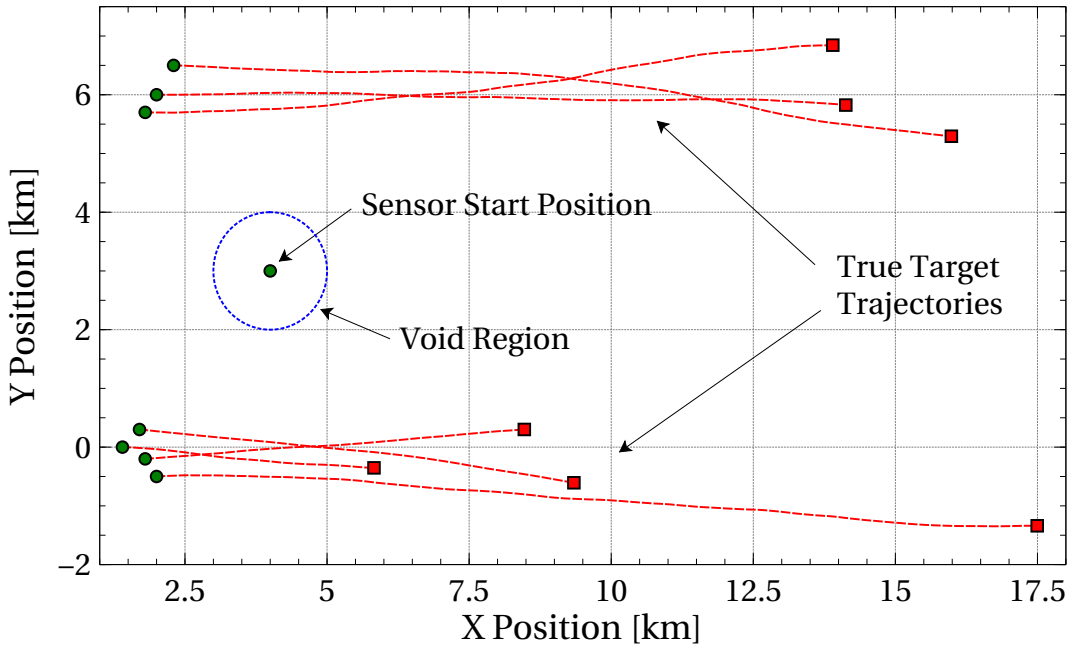


Figure 41: Control scenario 1, true targets trajectories. Note that the targets appear and disappear at different times, which is not represented on the plot.

To evaluate the control performance, we compare the estimation accuracy under the proposed control scheme, against the case of a stationary sensor, and the case where the sensor performs randomly chosen course changes. For each of the three cases, we have performed 100 Monte Carlo runs, and used a multi-target miss distance known as the optimal sub-pattern assignment (OSPA) metric [173], to quantify

the positional error between the filter estimates and the ground truth. Figure 42 shows the average of the OSPA distance versus time where the OSPA cutoff parameter is $c = 200$m and the order parameter is $p = 2$. These results show that the proposed control strategy provides the best estimation performance. In the cases where the sensor is stationary or undergoing randomly chosen actions, the performance is significantly worse, since they have no mechanism for positioning the sensor in the most favourable location. This demonstrates that the use of Cauchy-Schwarz divergence as the reward function has been effective at reducing the estimation error of the system.

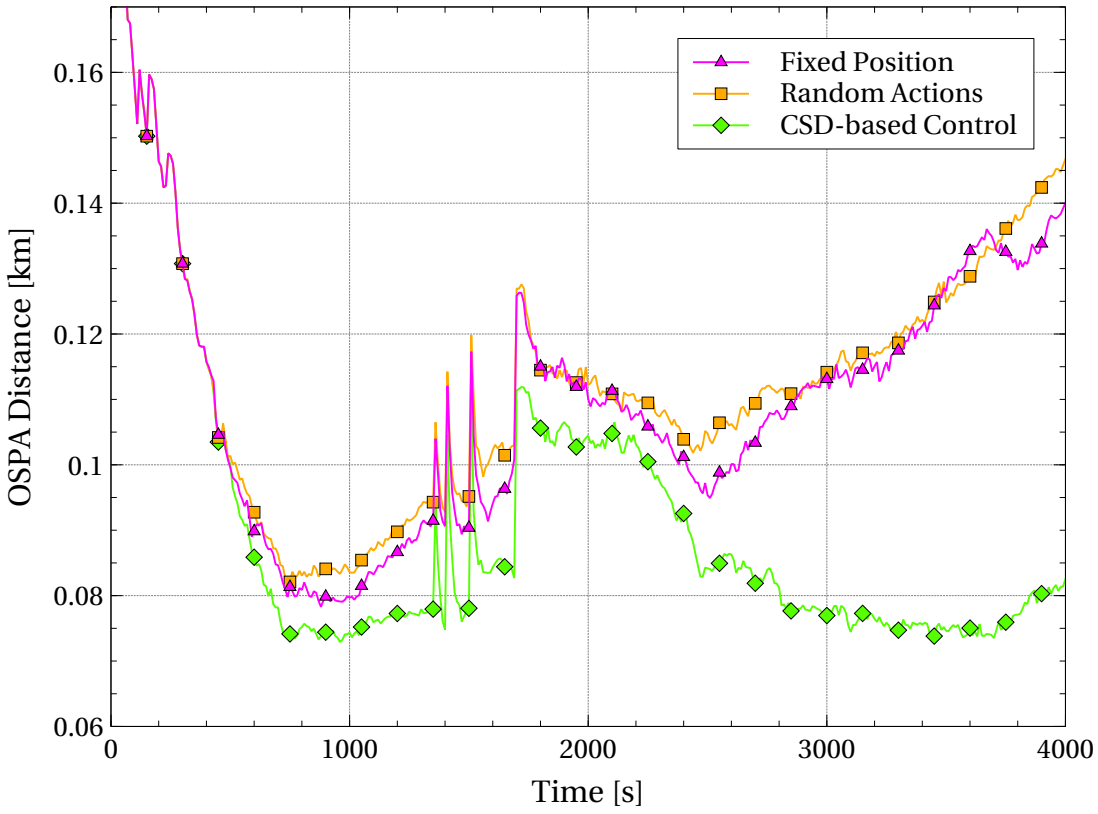


Figure 42: Control scenario 1, OSPA results. The cases of fixed sensor location and randomised control actions perform about the same. The control based on the Cauchy-Schwarz divergence and void probability results in much lower tracking error.

Figure 43 shows a heatmap summarising the paths taken by the sensor over the 100 Monte Carlo runs. From this diagram, the general trend of the controlled sensor's trajectory can be observed. Intuition would suggest that the sensor should move closer to the areas with the higher concentration of targets, which agrees with the trend shown in Figure 43.

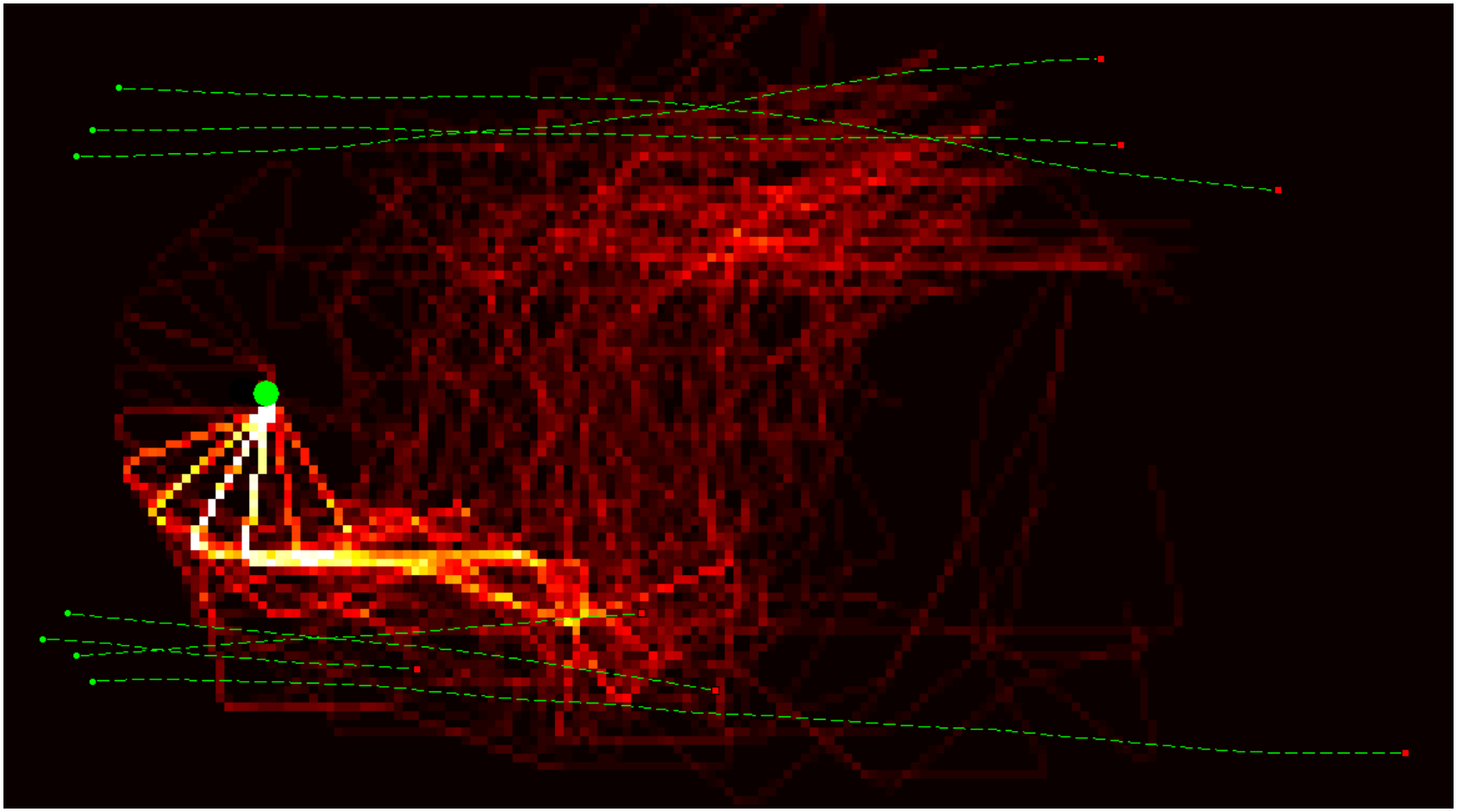


Figure 43: Control scenario 1, heatmap of the sensor location over 100 Monte Carlo runs. Brighter colors represent locations that were visited more frequently. The sensor usually starts by moving towards the four targets at the bottom. After some of those targets terminate, the sensor moves towards the three targets at the top. A few exceptions to this behaviour can be seen, but the general trend is clearly visible.

To demonstrate the operation of the control scheme, we now show a single run which exhibits the typical control behaviour. We have shown the scenario geometry and expected Cauchy-Schwarz divergence for each potential action at five different time instants; at 400s when the first decision is made (Figure 44), the third decision at time 1200s (Figure 45), the sixth decision at time 2400s (Figure 46), the eighth decision at time 3200s (Figure 47), and finally, at the end of the scenario at time 4000s (Figure 48).

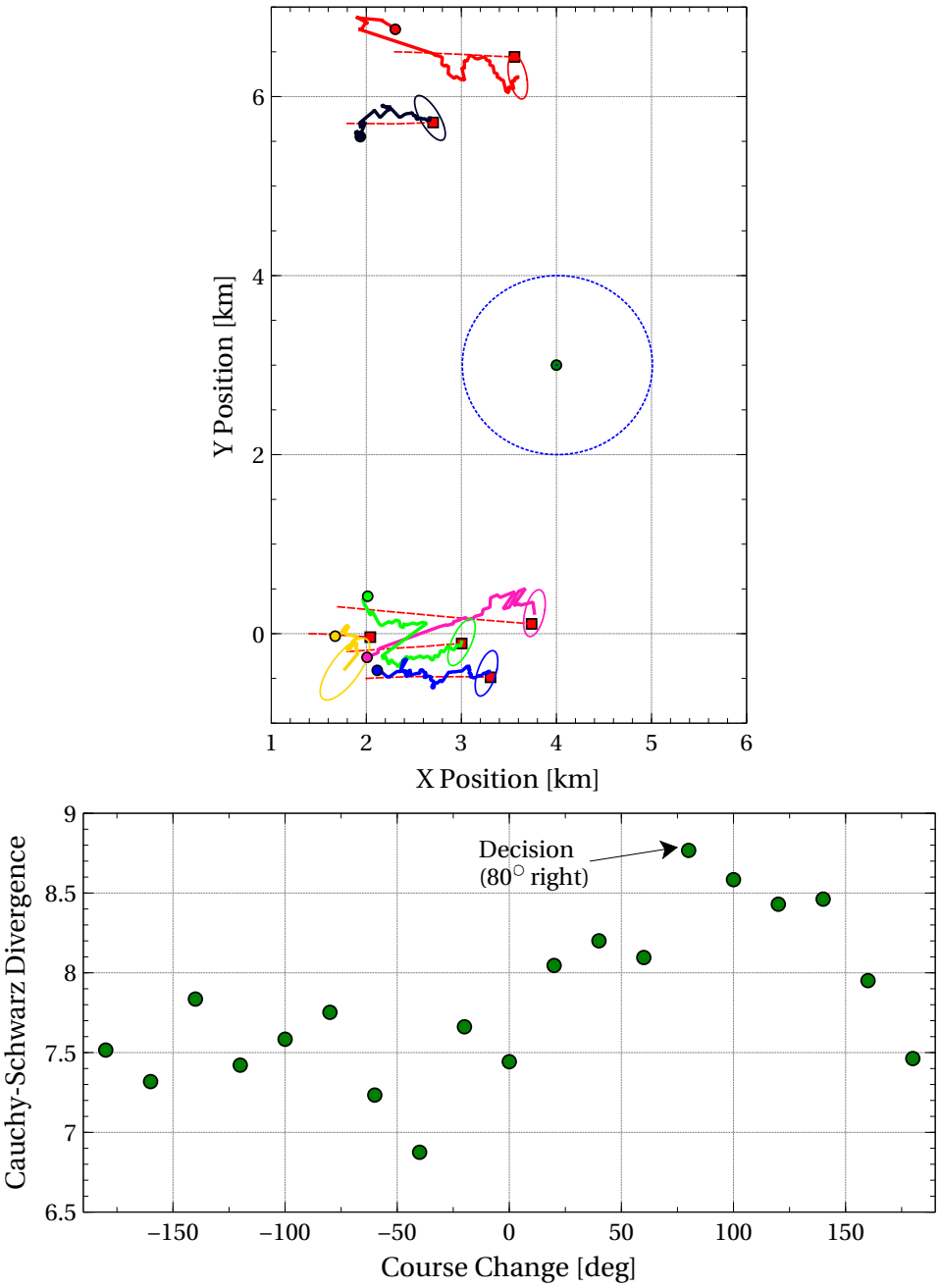


Figure 44: Control scenario 1, geometry and reward curve at the time of the first decision (400s). The sensor platform is stationary for the first 400s, and pointing towards the right. The first decision made by the control algorithm is to turn 80° to the right, which takes the sensor towards the group of four targets.

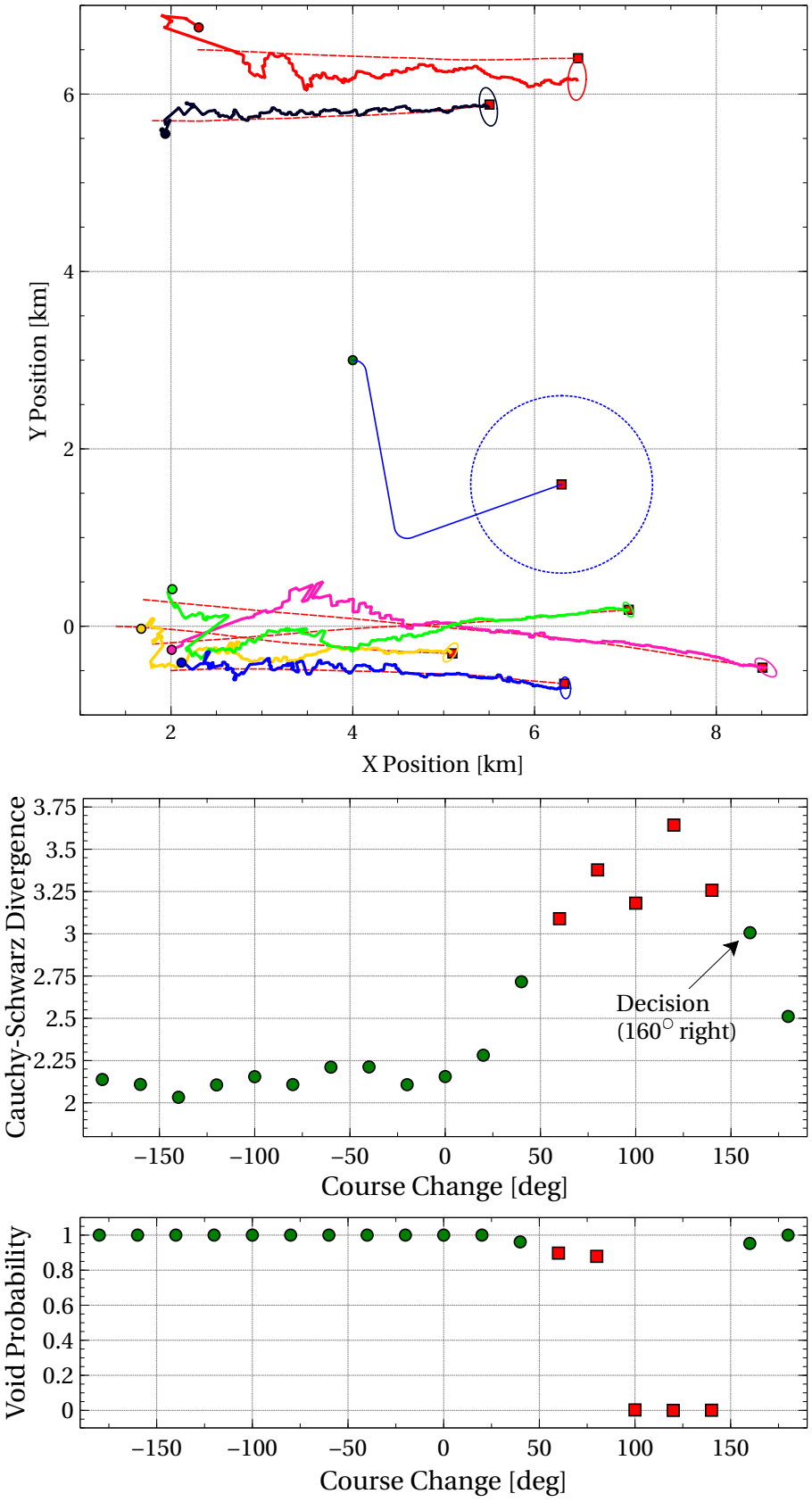


Figure 45: Control scenario 1, geometry, reward curve and void probability curve at the time of the third decision (1200s). Four of the manoeuvres do not satisfy the constraint, because they would result in high probability of targets getting too close to the sensor. Excluding these, the best remaining decision is to turn 160° to the right.

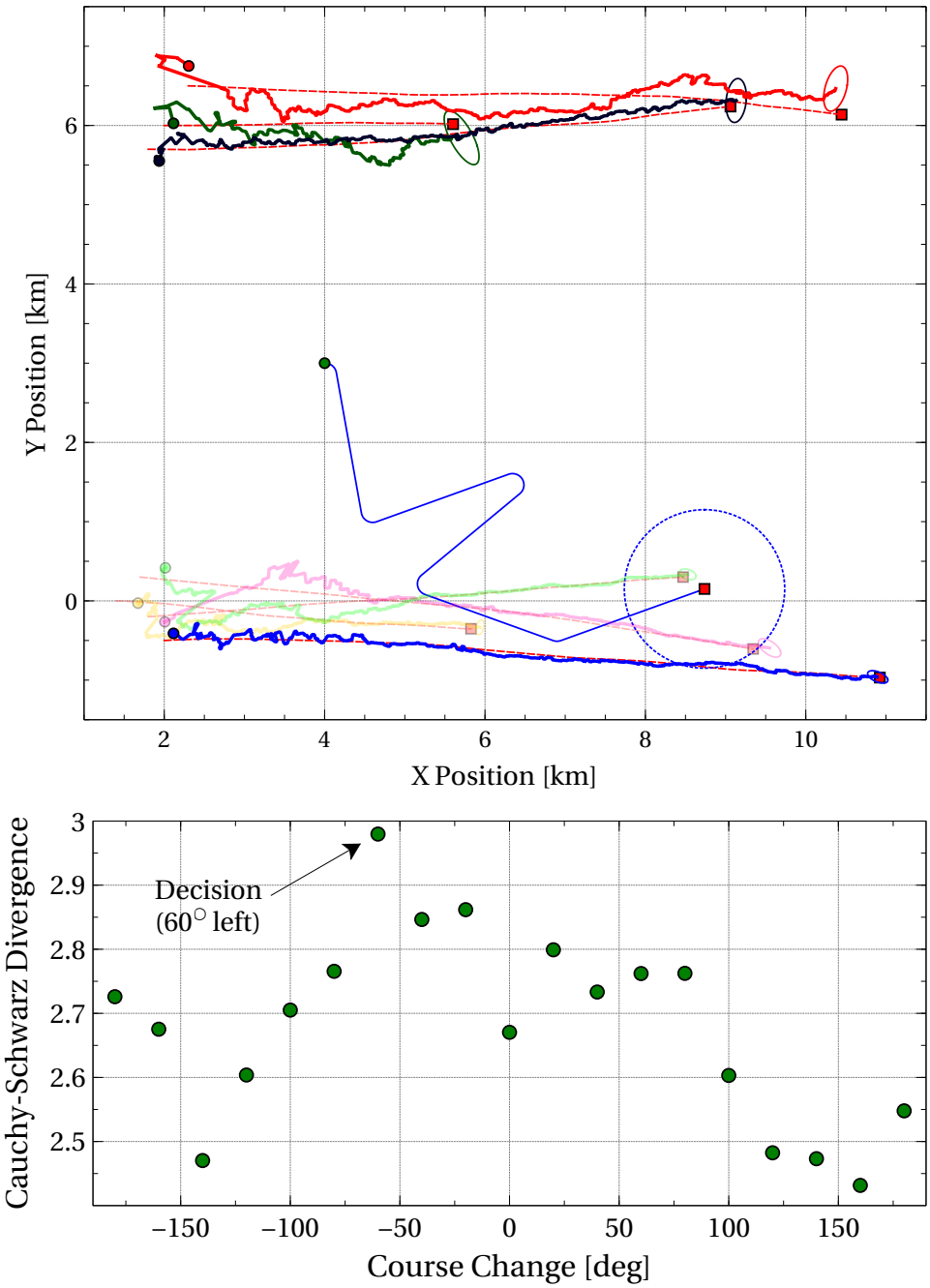


Figure 46: Control scenario 1, geometry and reward curve at the time of the sixth decision (2400s). Three of the targets at the bottom have terminated since time 1200s (as indicated by the faint tracks), and one additional target has appeared at the top. As a result, the algorithm decides to turn 60° left, taking it closer to the three targets at the top.

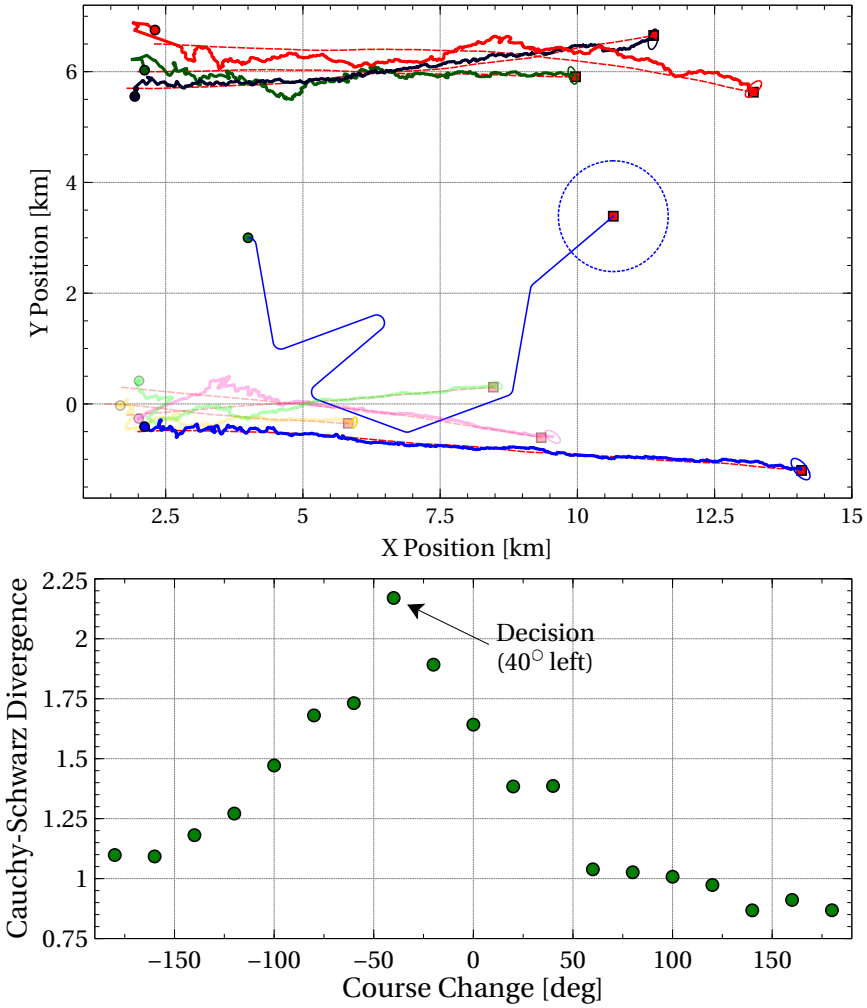


Figure 47: Control scenario 1, geometry and reward curve at the time of the eighth decision (3200s). The algorithm decides on a slight turn to the left, which takes it closer to the group of three targets.

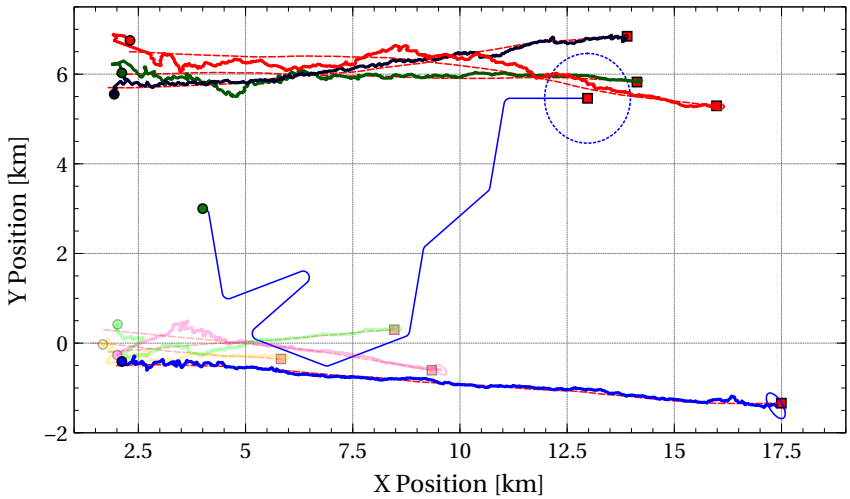


Figure 48: Control scenario 1, geometry at time 4000s. The sensor is now following the group of three targets at the top.

5.4.5.2 *Scenario 2*

This scenario consists of 8 targets at various locations and moving in different directions, and unlike the previous scenario, the best path for the sensor to take is not immediately obvious. The scenario geometry is shown in Figure 49, which also depicts one of the typical sensor trajectories obtained during the 100 Monte Carlo runs. The starting location for the sensor is fixed near the top right-hand corner for all runs, and in the particular case shown in Figure 49, it moves around the surveillance area, appearing to visit each pair of targets in sequence.

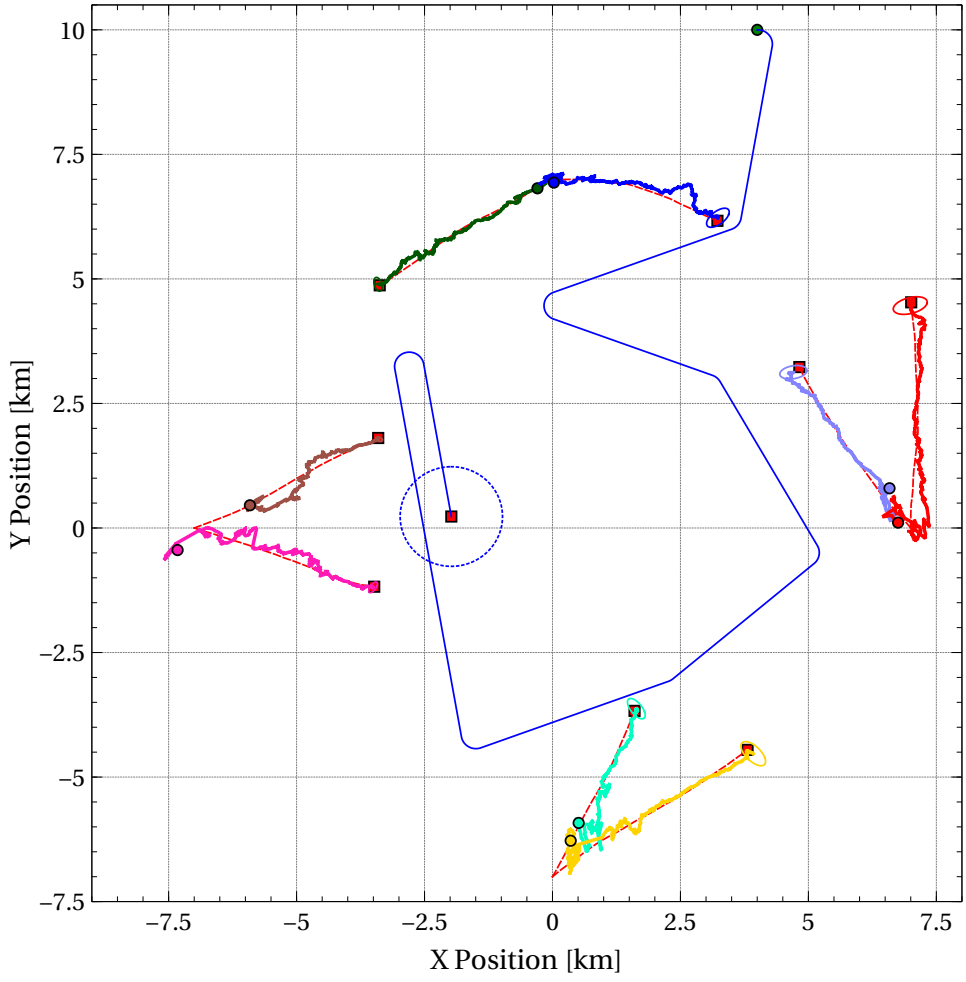


Figure 49: Control scenario 2, typical sensor trajectory under the proposed control scheme. The true and estimated target trajectories are also shown.

Figure 50 shows a comparison of the OSPA distance obtained for the cases of fixed sensor location, random actions, and with the proposed control strategy. The sensor with fixed position performs worst, because it has difficulty tracking the targets near the bottom of the surveillance region due to their large distance from the sensor. Moving with randomised actions improves the performance, because despite the

randomness of the chosen trajectories, the sensor still has the opportunity to move closer to the far-away targets. The proposed control strategy outperforms both the fixed and randomly moving sensors, as indicated by the lower OSPA distance. Due to the stochastic nature of the problem, the exact behaviour observed in Figure 50 is not necessarily replicated on every run. However the sensor generally moves around the centre of the surveillance region and attempts to visit each target in sequence. This can be observed in Figure 51, which shows that the sensor spends most of the time moving around between the targets.

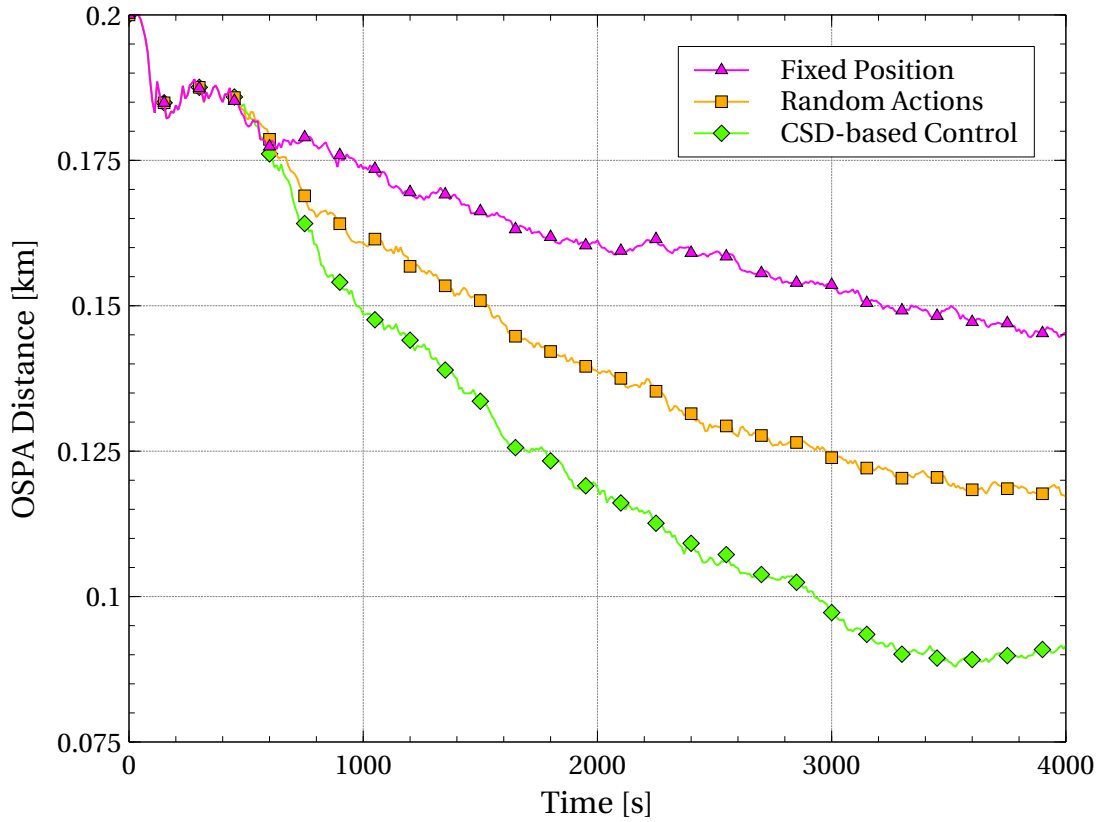


Figure 50: Control scenario 2, OSPA results. Comparison between fixed sensor location, randomly chosen actions, and actions chosen using the Cauchy-Schwarz divergence and void probability.

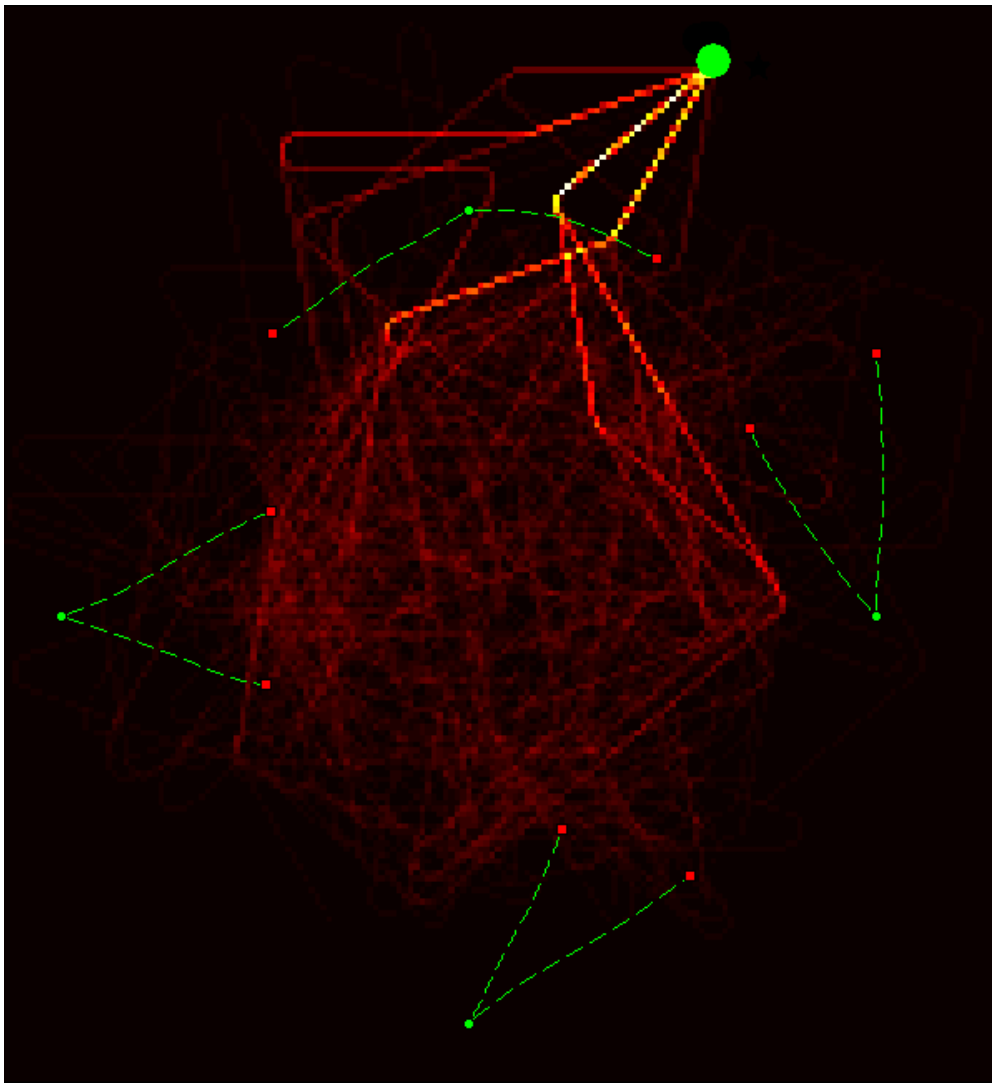


Figure 51: Control scenario 2, heatmap showing the control behaviour over 100 Monte Carlo runs.

5.4.6 *Pseudo-code for Multi-target Sensor Control Scheme*

The overall tracking and sensor control scheme is summarised by Algorithm 30. The function for deciding a single control action is given in Algorithm 31, which makes a decision based on the current GLMB density, by computing the reward and constraint values for each potential action and selecting the one with highest reward subject to the constraint being satisfied. The computation of the reward and constraint values is done according to Algorithms 32 and 33.

Algorithm 30: Multi-target tracking and control scheme.

Input: Measurement sampling times ($t_{1:N}$), decision times ($s_{1:M}$), space of control actions ($\mathbb{A}$)

Parameters: Control horizon length (H), sampling interval over control horizon ($\delta_{t,c}$), GLMB birth density (Φ_B)

Output: Track estimates (τ_x), track labels (τ_ℓ)

Function $[\tau_x, \tau_\ell] = \texttt{RunControl}\,(t_{1:N}, s_{1:M}, \mathbb{A})$ **is**
  $d = 1$;
  $\tau_x = \emptyset$;
  $\tau_\ell = \emptyset$;
  for $k = 1 : N$ **do**
    if $k = 1$ **then**
      $\Phi_+ = \Phi_B$;
    else
      $\Phi_{S,k} = \texttt{PredictSurvivalGLMB}\,(\Phi_{k-1}, t_k)$;
      $\Phi_+ = \texttt{MultiplyGLMB}\,(\Phi_{S,k}, \Phi_B)$;
    $Z = \texttt{GetMeasurements}\,(t_k)$;
    $\Phi_k = \texttt{UpdateGLMB}\,(\Phi_+, Z)$;
    $[\tau_x, \tau_\ell] = \texttt{ExtractEstimatesGLMB}\,(\Phi_k, \tau_x, \tau_\ell)$;
    $\Phi_k = \texttt{PruneGLMB}\,(\Phi_k)$;
    if $t_k = s_d$ **then**
      $u_k = \texttt{SensorState}\,(t_k)$;
      $t_c = [t_k + \delta_{t,c}, t_k + 2\delta_{t,c}, \ldots, t_k + H\delta_{t,c}]$;
      $\alpha = \texttt{ChooseAction}\,(\Phi_k, \Phi_B, u_k, \mathbb{A}, t_c)$;
      $\texttt{ExecuteAction}\,(\alpha)$;
      $d = d + 1$;

Algorithm 31: Choose control action.

Input: GLMB density (Φ), GLMB birth density (Φ_B), current sensor state (u), space of control actions ($\mathbb{A}$), sample times ($t_{1:H}$)

Parameters: Radius of exclusion zone (r_{void}), void probability threshold (Q_{thresh}), survival probability (P_S), detection probability (P_D), clutter intensity (λ), clutter region (R)

Output: Control action (α_{opt})

Function $\alpha_{opt} = \texttt{ChooseAction}\left(\Phi, \Phi_B, u, \mathbb{A}, t_{1:H}\right)$ **is**
  $Q_{min} = \texttt{ComputeConstraints}\left(\Phi, \Phi_B, u, \mathbb{A}, t_{1:H}\right)$;
  $D = \texttt{ComputeRewards}\left(\Phi, \Phi_B, u, \mathbb{A}, t_{1:H}\right)$;
  $D_{max} = 0$;
  for $\alpha \in \mathbb{A}$ **do**
    $\bar{D}^{(\alpha)} = \texttt{Mean}\left(D^{(:,\alpha)}\right)$;
    if $\left(Q_{min}^{(\alpha)} > Q_{thresh}\right)$ & $\left(\bar{D}^{(\alpha)} > D_{max}\right)$ **then**
      $\alpha_{opt} = \alpha$;
      $D_{max} = \bar{D}^{(\alpha)}$;

Algorithm 32: Compute rewards for control actions.

Input: GLMB density (Φ), GLMB birth density (Φ_B), current sensor state (u), space of control actions ($\mathbb{A}$), measurement sample times ($t_{1:H}$)

Parameters: Number of multi-target samples (N), survival probability (P_S), detection probability (P_D), clutter intensity (λ), clutter region (R)

Output: Vector of rewards (D)

Function $D = \texttt{ComputeRewards}\left(\Phi, \Phi_B, u, \mathbb{A}, t_{1:H}\right)$ **is**
  for $i = 1 : N$ **do**
    $X_0 = \texttt{DrawSampleGLMB}\left(\Phi\right)$;
    for $k = 1 : H$ **do**
      $X_S = \texttt{SampleSurvivals}\left(X_{k-1}, P_S\right)$;
      $X_B = \texttt{DrawSamplesGLMB}\left(\Phi_B, 1\right)$;
      $X_j = X_S \cup X_B$;
    for $\alpha \in \mathbb{A}$ **do**
      $\Psi_0 = \Phi$;
      for $k = 1 : H$ **do**
        $u_k = \texttt{SensorPosition}\left(\alpha, u, t_k\right)$;
        $\tilde{Z}_k = \texttt{SampleMeasurements}\left(X_k, u_k, P_D, \lambda, R\right)$;
        $\Psi_k = \texttt{PredictSurvivalGLMB}\left(\Psi_{k-1}, t_k\right)$;
        $\Psi_k = \texttt{MultiplyGLMB}\left(\Psi_k, \Phi_B\right)$;
        $\Psi_k = \texttt{UpdateGLMB}\left(\Psi_k, \tilde{Z}_k\right)$;
        $\Psi_k = \texttt{PruneGLMB}\left(\Psi_k\right)$;
      $D^{(i,\alpha)} = \texttt{CauchySchwarzGLMB}\left(\Omega_H, \Psi_H^{(\alpha)}\right)$;

Algorithm 33: Compute constraint values for control actions.

Input: GLMB density (Φ), GLMB birth density (Φ_B), current sensor state (u), space of control actions ($\mathbb{A}$), sample times ($t_{1:H}$)
Parameters: Survival probability (P_S), void radius (r_{void})
Output: Minimum void probabilities (Q_{min})

Function $Q_{min} = \texttt{ComputeConstraints}\left(\Phi, \Phi_B, u, \mathbb{A}, t_{1:H}\right)$ **is**
    $\Omega_0 = \Phi$;
    for $k = 1 : H$ **do**
        $\Omega_k = \texttt{PredictSurvivalGLMB}\left(\Omega_{k-1}, t_k\right)$;
        $\Omega_k = \texttt{MultiplyGLMB}\left(\Omega_k, \Phi_B\right)$;
        $\Omega_k = \texttt{PruneGLMB}\left(\Omega_k\right)$;
    for $\alpha \in \mathbb{A}$ **do**
        $Q_{min}^{(\alpha)} = 1$;
        for $k = 1 : H$ **do**
            $u_k = \texttt{SensorPosition}\left(\alpha, u, t_k\right)$;
            $S = \texttt{Circle}\left(u_k, r_{void}\right)$;
            $Q_k^{(\alpha)} = \texttt{VoidProbabilityGLMB}\left(\Omega_k, S\right)$;
            if $Q_k^{(\alpha)} < Q_{min}^{(\alpha)}$ **then**
                $Q_{min}^{(\alpha)} = Q_k^{(\alpha)}$;

6 CONCLUSIONS AND FUTURE WORK

6.1 CONCLUSIONS

This dissertation presents new methods for estimation and sensor control for tracking an unknown and time-varying number of targets, using the methodology of labelled random finite sets. Two new multi-target trackers are proposed, which are based on more realistic measurement likelihood models than the more traditional standard measurement model. The first of these uses a model that accommodates possible merging of target measurements, which is a common phenomenon when tracking multiple targets using a sensor with limited resolution. The second proposed technique accommodates multiple simultaneous detections per target, by modelling the physical target extent. This is an important consideration in cases where the targets are large and/or the sensor resolution is high, such that a single target may occupy more than one detection cell at a time. Lastly, the problem of sensor control is addressed with the use of newly derived analytical expressions for the Cauchy-Schwarz divergence between two GLMB densities, and the GLMB void probability functional.

For the case of merged measurements, an algorithm called GLMB-MP is proposed, which forms partitions of the target set in order to model unresolved target groups. This algorithm is computationally feasible only for a relatively low number of targets, since the number of partitions grows too large as the number of targets increases. As a cheaper alternative, an algorithm called GLMB-MR was also propopsed, which uses relaxed measurement assignment to form the unresolved groups. Simulations were carried out on a bearings-only tracking scenario, in which some of the target bearing observations became merged due to finite sensor resolution effects. The results

confirmed that the GLMB-MP outperforms the relaxed assignment approach as expected, and that both of the proposed methods offer significant improvement over the GLMB filter with the standard measurement likelihood model, since the latter has no mechanism for maintaining tracks in the presence of merged measurements. The results also demonstrate that the GLMB-MP algorithm is indeed more computationally demanding than the GLMB-MR, but still feasible for clusters involving up to four targets. Although less accurate, the GLMB-MR filter is capable of processing larger clusters of targets without the computational bottleneck.

For multiple extended target tracking, two algorithms were proposed, namely the GGIW-GLMB and GGIW-LMB filter. Both are based on modelling the problem using labelled random finite sets, and gamma Gaussian inverse Wishart mixtures. The proposed algorithms estimate the number of targets, and their kinematics, extensions and measurement rates. The major advantage these methods have over the previously developed GGIW-(C)PHD filters is the inclusion of target labels, allowing for continuous target tracks, which is not directly supported by the (C)PHD filters. Of the two proposed algorithms, the GGIW-GLMB filter is more accurate as it involves fewer approximations, but it is also more computationally demanding than the GGIW-LMB filter. Simulation results demonstrate that the algorithms outperform the GGIW-(C)PHD filters, especially in cases where the performance of the CPHD filter is degraded due to the spooky effect. It was also demonstrated that the filter performs well in a real-world application, in which laser range-finders are used to track pedestrians.

Finally, two useful properties of generalised labelled multi-Bernoulli models were derived; an analytical form for the Cauchy-Schwarz divergence between two GLMBs, and an analytical form for the void probability functional of a GLMB. These properties have clear applications in multi-target sensor control, which was demonstrated by their use in a simulated control scenario, where the objective was to plan a sensor trajectory that optimises the multi-target tracking error. The problem was formulated as a constrained POMDP, with a reward function based on the expected Cauchy-Schwarz divergence, and a constraint based on the void probability, to ensure adequate separation between the sensor and targets was maintained at all times. The results showed that this method was highly effective at reducing the multi-target estimation error, compared to cases where the sensor was stationary or undergoing random actions.

The work in this dissertation has demonstrated that labelled random finite sets have useful applications well beyond multi-target tracking under the standard measurement model. More realistic models incorporating phenomena such as merged measurements and extended targets are indeed feasible, as is the application of labelled random finite sets to multi-target sensor control problems.

6.2 FUTURE WORK

The following are some potential areas for future research in the area of multi-target estimation and control using the labelled random finite set formulation.

- Developing a labelled RFS-based tracker which can handle both merged measurements and extended targets simultaneously would be advantageous. This dissertation has presented independent methods for handling merged measurements and extended targets. In practice however, the decision on which formulation to use may not be clear-cut, as both phenomena may be present within the same scenario. A completely general solution to detection level multi-target tracking would involve associating groups of measurements with groups of targets. In essence, this could be considered as an alternative to the track-before-detect approach to multi-target tracking.

- Incorporating more realistic target extent models into the (G)LMB filter could be significantly beneficial. The work in this dissertation only considers the case of elliptical targets, which are modelled using gamma Gaussian inverse-Wishart distributions. Although the GGIW model is appropriate for some target types, for other types, this model does not adequately capture the spatial structure.

- Multi-target smoothing using the GLMB model has the potential to offer improved estimation performance over filtering alone. Smoothing would be particularly beneficial in low observability scenarios, since in these cases, the use of future information can go a long way towards resolving ambiguities in the observed data. The development of such a smoother would also allow for an equitable comparison to be carried out between the GLMB and MHT approaches to the multi-target tracking problem.

- Integrating model parameter estimation into the labelled RFS methods would enhance their generality and reduce their reliance on prior assumptions. For

example, on-line estimation of the clutter intensity function, the target detection probabilities, and the target birth/death model would all offer significant benefits when implementing these algorithms in operational systems.

- A multi-step sensor control strategy based on the Cauchy-Schwarz divergence and void probability functional for GLMBs would be a highly useful extension to the work presented in Chapter 5. Using these properties of the GLMB, along with a technique such as dynamic programming [26], could lead to the development of such a scheme. The Cauchy-Schwarz divergence could be particularly suitable for this type of control, because unlike other methods, it can naturally accommodate the formulation of an additive cost function.

- Fast implementations of the GLMB filter for high-fidelity models are necessary for their implementation in real-world systems. The GLMB filter has a structure which can be easily parallelised on several levels, which offers the potential for real-time implementation. A principled and fast method for dividing a GLMB density into independent target clusters would also be beneficial in speeding up the processing. The use of parallellisation and clustering together could pave the way for the application of labelled RFS methods in large scale tracking problems. A prime example of this is in the tracking of space debris, where there is a need to track a very large number of targets simultaneously.

- This dissertation only considers the case of centralised multi-target estimation and control, however, distributed systems are also becoming increasingly important. Work such as [191], [13] and [52] have studied multi-target distributed estimation under the standard point-detection model, which can be considered as multi-target generalisations of the covariance intersection method first proposed in [98]. It may be fruitful to investigate combining these techniques with the methods proposed in this dissertation, with the goal of developing distributed multi-target trackers for high-fidelity sensor models. Additionally, a centralised approach to multi-target simultaneous localisation and mapping (SLAM) for point-detections has been proposed in [46], and it may be interesting to investigate the incorporation of improved sensor models, and also performing distributed SLAM by generalising the work in [99] to the multi-target case.

BIBLIOGRAPHY

[1] B. D. O. Anderson and J. B. Moore. *Optimal Filtering*. Prentice-Hall, 1979. (Cited on pages 11 and 13.)

[2] C. Andrieu, A. Doucet, S. S. Singh, and V. B. Tadic. Particle methods for change detection, system identification, and control. *Proceedings of the IEEE*, 92(3):423–438, 2004. (Cited on page 18.)

[3] C. Andrieu, A. Doucet, and R. Holenstein. Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):269–342, 2010. (Cited on page 17.)

[4] D. Angelosante, E. Biglieri, and M. Lops. Multiuser detection in a dynamic environment. Part II: Joint user identification and parameter estimation. *IEEE Transactions on Information Theory*, 55(5):2365–2374, May 2009. (Cited on page 164.)

[5] I. Arasaratnam and S. Haykin. Cubature Kalman filters. *IEEE Transactions on Automatic Control*, 54(6):1254–1269, June 2009. (Cited on page 38.)

[6] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for on-line non-linear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, February 2002. (Cited on page 16.)

[7] F. Baccelli, M. Klein, M. Lebourges, and S. A. Zuyev. Stochastic geometry and architecture of communication networks. *Telecommunication Systems*, 7(1-3):209–227, 1997. (Cited on page 164.)

[8] A. J. Baddeley and M. N. M. Van Lieshout. Stochastic geometry models in high-level vision. *Journal of Applied Statistics*, 20(5-6):231–256, 1993. (Cited on page 164.)

[9] Y. Bar-Shalom and T. E. Fortmann. *Tracking and Data Association*. Academic Press, San Diego, CA, 1988. (Cited on page 3.)

[10] Y. Bar-Shalom and X. R. Li. *Multitarget-Multisensor Tracking: Principles and Techniques*. YBS Publishing, 1995. (Cited on page 44.)

[11] Y. Bar-Shalom and E. Tse. Tracking in a cluttered environment with probabilistic data association. *Automatica*, 11(5):451–460, September 1975. (Cited on pages 3 and 44.)

[12] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, New York, 2001. (Cited on page 13.)

[13] G. Battistelli, L. Chisci, C. Fantacci, A. Farina, and A. Graziano. Consensus CPHD filter for distributed multitarget tracking. *IEEE Journal of Selected Topics in Signal Processing*, 7(3):508–520, 2013. (Cited on page 200.)

[14] M. Baum and U. D. Hanebeck. Shape tracking of extended objects and group targets with star-convex RHMs. In *Proceedings of the 14th International Conference on Information Fusion*, Chicago, IL, USA, July 2011. (Cited on page 125.)

[15] M. Baum, B. Noack, and U. D. Hanebeck. Extended object and group tracking with elliptic random hypersurface models. In *Proceedings of the 13th International Conference on Information Fusion*, Edinburgh, UK, July 2010. (Cited on page 125.)

[16] M. Beard, B.-T. Vo, B.-N. Vo, and S. Arulampalam. Gaussian mixture PHD and CPHD filtering with partially uniform target birth. In *Proccedings of the 15th International Conference on Information Fusion*, Singapore, July 2012. (Cited on pages 8, 9, 29, and 30.)

[17] M. Beard, B.-T. Vo, and B.-N. Vo. Multitarget filtering with unknown clutter density using a bootstrap GMCPHD filter. *IEEE Signal Processing Letters*, 20(4): 323–326, 2013. (Cited on pages 8 and 9.)

[18] M. Beard, B.-T. Vo, B.-N. Vo, and S. Arulampalam. A partially uniform target birth model for Gaussian mixture PHD/CPHD filtering. *IEEE Transactions on Aerospace and Electronic Systems*, 49(4):2835–2844, October 2013. (Cited on pages 8, 9, 29, and 30.)

[19] M. Beard, B.-T. Vo, and B.-N. Vo. Multi-target tracking with merged measurements using labelled random finite sets. In *Proceedings of the 17th International Conference on Information Fusion*, Salamanca, Spain, July 2014. (Cited on pages 7 and 85.)

[20] M. Beard, S. Reuter, K. Granström, B.-T. Vo, B.-N. Vo, and A. Scheel. A generalised labelled multi-Bernoulli filter for extended multi-target tracking. In *Proceedings of 18th International Conference on Information Fusion*, Washington DC, USA, July 2015. (Cited on pages 7 and 123.)

[21] M. Beard, B.-T. Vo, and B.-N. Vo. Bayesian multi-target tracking with merged measurements using labelled random finite sets. *IEEE Transactions on Signal Processing*, 63(6):1433–1447, March 2015. (Cited on pages 7, 85, 90, 95, 124, 126, and 165.)

[22] M. Beard, B.-T. Vo, B.-N. Vo, and S. Arulampalam. Void probabilities and Cauchy-Schwarz divergence for generalized labeled multi-Bernoulli models. Submitted to IEEE Transactions on Information Theory, arXiv preprint arXiv:1510.05532, October 2015. (Cited on pages 8 and 163.)

[23] M. Beard, B.-T. Vo, B.-N. Vo, and S. Arulampalam. Sensor control for multi-target tracking using Cauchy-Schwarz divergence. In *Proceedings of the 18th International Conference on Information Fusion*, Washington DC, July 2015. (Cited on pages 8 and 163.)

[24] M. Beard, S. Reuter, K. Granström, B.-T. Vo, B.-N. Vo, and A. Scheel. Multiple extended target tracking with labelled random finite sets. April 2016. (Cited on pages 7 and 123.)

[25] B. Benfold and I. Reid. Stable multi-target tracking in real-time surveillance video. In *2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3457–3464. IEEE, 2011. (Cited on page 5.)

[26] D. P. Bertsakas. *Dynamic Programming and Optimal Control*. Athena Scientific, Belmont, MA, USA, 1995. (Cited on pages 177, 178, and 200.)

[27] D. P. Bertsekas. The auction algorithm: A distributed relaxation method for the assignment problem. *Annals of Operations Research*, 14(1):105–123, 1988. (Cited on page 106.)

[28] E. Biglieri and M. Lops. Multiuser detection in a dynamic environment. Part I: User identification and data detection. *IEEE Transactions on Information Theory*, 53(9):3158–3170, September 2007. (Cited on page 164.)

[29] S. Blackman and R. Popoli. *Design and Analysis of Modern Tracking Systems*. Artech House, 1999. (Cited on pages 5, 6, and 67.)

[30] H. A. P. Blom and E. A. Bloem. Bayesian tracking of two possibly unresolved maneuvering targets. *IEEE Transactions on Aerospace and Electronic Systems*, 43 (2):612–627, April 2007. (Cited on page 87.)

[31] M. Bolić, P. M. Djurić, and S. Hong. Resampling algorithms for particle filters: A computational complexity perspective. *EURASIP Journal on Applied Signal Processing*, 2004:2267–2277, 2004. (Cited on page 17.)

[32] M. Briers, A. Doucet, and S. Maskell. Smoothing algorithms for state–space models. *Annals of the Institute of Statistical Mathematics*, 62(1):61–89, 2010. (Cited on page 18.)

[33] J.E. Le Cadre and C. Jauffret. Discrete-time observability and estimability analysis for bearings-only target motion analysis. *IEEE Transactions on Aerospace and Electronic Systems*, 33(1):178–201, January 1997. (Cited on page 36.)

[34] G. Casella and C. P. Robert. Rao-Blackwellisation of sampling schemes. *Biometrika*, 83:81–94, March 1996. (Cited on page 179.)

[35] D. A. Castanon and L. Carin. Stochastic control theory for sensor management. In A. O. Hero, D. A. Castanon, D. Cochran, and K. Kastella, editors, *Foundations and Applications of Sensor Management*, chapter 2, pages 7–32. Springer, New York, 2008. (Cited on page 177.)

[36] K.-C. Chang and Y. Bar-Shalom. Joint probabilistic data association for multitarget tracking with possibly unresolved measurements and maneuvers. *IEEE Transactions on Automatic Control*, 29(7):585–594, July 1984. (Cited on pages 86 and 102.)

[37] H. Chen, K. Pattipati, T. Kirubarajan, and Y. Bar-Shalom. General data association with possibly unresolved measurements using linear programming. In *2003 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW '03)*, volume 9, June 2003. (Cited on page 87.)

[38] D. Clark and R. Mahler. Generalized PHD filters via a general chain rule. In *Proceedings of the 15th International Conference on Information Fusion*, Singapore, July 2012. (Cited on page 125.)

[39] D. Clark, B.-T. Vo, and B.-N. Vo. Gaussian particle implementations of probability hypothesis density filters. In *Proceedings of the 2007 IEEE Aerospace Conference*. IEEE, 2007. (Cited on page 26.)

[40] D. E. Clark and J. Bell. Convergence results for the particle PHD filter. *IEEE Transactions on Signal Processing*, 54(7):2652–2661, 2006. (Cited on page 26.)

[41] J. M. C. Clark, R. B. Vinter, and M. M. Yaqoob. Shifted Rayleigh filter: A new algorithm for bearings-only tracking. *IEEE Transactions on Aerospace and Electronic Systems*, 43(4):1373–1384, October 2007. (Cited on page 38.)

[42] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley-Interscience, New York, NY, USA, 1991. (Cited on page 164.)

[43] D. Daley and D. Vere-Jones. *An introduction to the theory of point processes*. Springer-Verlag, 1988. (Cited on pages 164, 166, 167, and 168.)

[44] F. E. Daum and R. J. Fitzgerald. The importance of resolution in multiple target tracking. In *SPIE 2235, Proc. SPIE Signal & Data Processing of Small Targets*, volume 329, 1994. (Cited on pages 6 and 86.)

[45] S. J. Davey. Tracking possibly unresolved targets with PMHT. In *Information, Decision and Control (IDC) 2007*, February 2007. (Cited on page 87.)

[46] H. Deusch, S. Reuter, and K. Dietmayer. The labeled multi-Bernoulli SLAM filter. *IEEE Signal Processing Letters*, 22(10):1561–1565, October 2015. (Cited on pages 164, 165, 171, and 200.)

[47] R. Douc and O. Cappé. Comparison of resampling schemes for particle filtering. In *Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis*, pages 64–69, Istanbul, Turkey, September 2005. IEEE. (Cited on page 17.)

[48] A. Doucet and V. B. Tadić. Parameter estimation in general state-space models using particle methods. *Annals of the Institute of Statistical Mathematics*, 55(2): 409–422, 2003. (Cited on page 18.)

[49] A. Doucet, S. Godsill, and C. Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3):197–208, 2000. (Cited on page 17.)

[50] A. Doucet, N. de Freitas, and N. Gordon. *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, 2001. (Cited on pages 2 and 16.)

[51] D. Dunne and T. Kirubarajan. Multiple model multi-Bernoulli filters for manoeuvering targets. *IEEE Transactions on Aerospace and Electronic Systems*, 49(4): 2679–2692, 2013. (Cited on page 29.)

[52] C. Fantacci, B.-N. Vo, B.-T. Vo, G. Battistelli, and L. Chisci. Consensus labeled random finite set filtering for distributed multi-object tracking. *arXiv preprint arXiv:1501.01579*, 2015. (Cited on page 200.)

[53] P. Fearnhead, D. Wyncoll, and J. Tawn. A sequential smoothing algorithm with linear computational cost. *Biometrika*, 97(2):447–464, 2010. (Cited on page 18.)

[54] M. Feldmann, D. Fränken, and J. W. Koch. Tracking of extended objects and group targets using random matrices. *IEEE Transactions on Signal Processing*, 59 (4):1409–1420, April 2011. (Cited on page 125.)

[55] R. A. Fisher. On the mathematical foundations of theoretical statistics. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character.*, pages 309–368, 1922. (Cited on page 1.)

[56] T. E. Fortmann, Y. Bar-Shalom, and M. Scheffe. Sonar tracking of multiple targets using joint probabilistic data association. *IEEE Journal of Oceanic Engineering*, 8(3):173–184, 1983. (Cited on page 3.)

[57] C. F. Gauss and C. H. Davis. *Theory of the Motion of the Heavenly Bodies Moving about the Sun in Conic Sections.* Courier Corporation, 2004. (Cited on page 1.)

[58] K. Gilholm and D. Salmond. Spatial distribution model for tracking extended objects. *IEE Proceedings (Radar, Sonar and Navigation)*, 152(5):364–371, October 2005. (Cited on page 124.)

[59] K. Gilholm, S. Godsill, S. Maskell, and D. Salmond. Poisson models for extended target and group tracking. In *Proceedings of SPIE Signal and Data Processing of Small Targets*, volume 5913, pages 230–241, San Diego, CA, USA, August 2005. (Cited on page 124.)

[60] S. Godsill and T. Clapp. Improvement strategies for Monte Carlo particle filters. In *Sequential Monte Carlo Methods in Practice*, pages 139–158. Springer, 2001. (Cited on page 18.)

[61] A. P. Goobic, M. E. Welser, S. T. Acton, and K. Ley. Biomedical application of target tracking in clutter. In *Signals, Systems and Computers, 2001. Conference Record of the Thirty-Fifth Asilomar Conference on*, volume 1, pages 88–92. IEEE, 2001. (Cited on page 5.)

[62] I. R. Goodman, R. P. S. Mahler, and H. T. Nguyen. *Mathematics of Data Fusion*. Kluwer Academic Publishers, 1997. (Cited on page 4.)

[63] N. Gordon, D. Salmond, and A. Smith. Novel approach to non-linear/non-Gaussian Bayesian state estimation. *IEE Proceedings F (Radar and Signal Processing)*, 140(2):107–113, 1993. (Cited on pages 2, 10, 16, and 17.)

[64] A. K. Gostar, R. Hoseinnezhad, and A. Bab-Hadiashar. Multi-Bernoulli sensor control for multi-target tracking. In *IEEE Eighth International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, pages 312–317. IEEE, 2013. (Cited on page 29.)

[65] A. K. Gostar, R. Hoseinnezhad, and A. Bab-Hadiashar. Sensor control for multi-object tracking using labeled multi-Bernoulli filter. In *Proceedings of the 17th International Conference on Information Fusion*. IEEE, 2014. (Cited on pages 6, 165, and 177.)

[66] K. Granström. *Extended Target Tracking Using PHD Filters*. PhD thesis, Department of Electrical Engineering, Linkoping University, 2012. (Cited on page 6.)

[67] K. Granström and U. Orguner. A PHD filter for tracking multiple extended targets using random matrices. *IEEE Transactions on Signal Processing*, 60(11):5657–5671, November 2012. (Cited on pages 125, 129, 133, 134, 141, 142, and 144.)

[68] K. Granström and U. Orguner. Estimation and maintenance of measurement rates for multiple extended target tracking. In *Proceedings of 15th International Conference on Information Fusion*, Singapore, July 2012. (Cited on page 140.)

[69] K. Granström, C. Lundquist, and U. Orguner. Extended target tracking using a Gaussian-mixture PHD filter. *IEEE Transactions on Aerospace and Electronic Systems*, 48(4):3268 – 3286, October 2012. (Cited on pages 125, 129, 131, 133, 140, 141, 142, and 144.)

[70] B. Grocholskya, A. Makarenko, and H. Durrant-Whyte. Information-theoretic coordinated control of multiple sensor platforms. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2003. (Cited on page 176.)

[71] A. Gupta and D. Nagar. *Matrix Variate Distributions*. Chapman & Hall, 2000. (Cited on page 129.)

[72] M. Haenggi. On distances in uniformly random networks. *IEEE Transactions on Information Theory*, 51(10):3584–3586, October 2005. (Cited on page 164.)

[73] M. Haenggi, J. Andrews, F. Baccelli, O. Dousse, and M. Franceschetti. Stochastic geometry and random graphs for the analysis and design of wireless networks. *IEEE Journal on Selected Areas in Communications*, 27(7):1029–1046, 2009. (Cited on page 164.)

[74] M. L. Hernandez, T. Kirubarajan, and Y. Bar-Shalom. Multisensor resource deployment using posterior Cramer-Rao bounds. *IEEE Transactions on Aerospace and Electronic Systems*, 40(2):399–416, 2004. (Cited on page 176.)

[75] A. O. Hero, D. Castanon, D. Cochran, and K. Kastella. *Foundations and applications of sensor management*. Springer Science & Business Media, 2007. (Cited on page 7.)

[76] Y. Ho and R. Lee. A Bayesian approach to problems in stochastic estimation and control. *IEEE Transactions on Automatic Control*, AC-9:333–339, 1964. (Cited on page 11.)

[77] H. Hoang, B.-N. Vo, B.-T. Vo, and R. Mahler. The Cauchy-Schwarz divergence for Poisson point processes. *IEEE Transactions on Information Theory*, 61(8):4475–4485, August 2015. (Cited on pages 164, 166, 167, and 168.)

[78] H. G. Hoang and B.-T. Vo. Sensor management for multi-target tracking via multi-Bernoulli filtering. *Automatica*, 50(4):1135–1142, 2014. (Cited on page 6.)

[79] J. D. Hol, T. B. Schon, and F. Gustafsson. On resampling algorithms for particle filters. In *IEEE Nonlinear Statistical Signal Processing Workshop*, pages 79–82. IEEE, 2006. (Cited on page 17.)

[80] R. Hoseinnezhad, B.-N. Vo, D. Suter, and B.-T. Vo. Multi-object filtering from image sequence without detection. In *2010 IEEE International Conference on*

Acoustics Speech and Signal Processing, pages 1154–1157. IEEE, 2010. (Cited on page 28.)

[81] R. Hoseinnezhad, B.-N. Vo, B.-T. Vo, and D. Suter. Bayesian integration of audio and visual information for multi-target tracking using a CB-MeMBer filter. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2300–2303. IEEE, 2011. (Cited on page 29.)

[82] R. Hoseinnezhad, B.-N. Vo, B.-T. Vo, and D. Suter. Visual tracking of numerous targets via multi-Bernoulli filtering of image data. *Pattern Recognition*, 45(10): 3625–3635, October 2012. (Cited on page 164.)

[83] J. Houssineau and D. Laneuville. PHD filter with diffuse spatial prior on the birth process with applications to GM-PHD filter. In *Proceedings of the 13th International Conference on Information Fusion*, Edinburgh, UK, July 2010. (Cited on pages 29 and 33.)

[84] J. D. Isom, S. P. Meyn, and R. D. Braatz. Piecewise linear dynamic programming for constrained POMDPs. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence*, 2008. (Cited on page 178.)

[85] C. Jauffret and D. Pillon. Observability in passive target motion analysis. *IEEE Transactions on Aerospace and Electronic Systems*, 32(4):1290–1300, October 1996. (Cited on page 36.)

[86] A. Jazwinski. *Stochastic Processes and Filtering Theory*. Academic Press, New York, 1970. (Cited on pages 2, 10, 11, and 13.)

[87] P. Jensfelt and S. Kristensen. Active global localization for a mobile robot using multiple hypothesis tracking. *IEEE Transactions on Robotics and Automation*, 17 (5):748–760, 2001. (Cited on page 5.)

[88] R. Jenssen, D. Erdogmus, K. E. Hild, J. C. Principe, and T. Eltoft. Optimizing the Cauchy-Schwarz PDF distance for information theoretic, non-parametric clustering. In *Proceedings of the 5th International Conference on Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 34–45, Berlin, Heidelberg, 2005. Springer-Verlag. (Cited on page 166.)

[89] R. Jenssen, J. C. Principe, D. Erdogmus, and T. Eltoft. The Cauchy-Schwarz divergence and Parzen windowing: Connections to graph theory and Mercer kernels. *Journal of the Franklin Institute*, 343(6):614–629, 2006. (Cited on page 166.)

[90] S. Jeong and J. K. Tugnait. Tracking of two targets in clutter with possibly unresolved measurements. *IEEE Transactions on Aerospace and Electronic Systems*, 44(2):748–765, April 2008. (Cited on page 87.)

[91] C. Ji, D. Merl, T. B. Kepler, and M. West. Spatial mixture modelling for unobserved point processes: Examples in immunofluorescence histology. *Bayesian Analysis*, 4:297–316, 2009. (Cited on page 164.)

[92] A. M. Johansen, S. Singh, A. Doucet, and B.-N. Vo. Convergence of the SMC implementation of the PHD filter. *Methodology and Computing in Applied Probability*, 8(2):265–291, 2006. (Cited on page 26.)

[93] B. A. Jones and B.-N. Vo. A labeled multi-Bernoulli filter for space object tracking. In *2014 AAS/AIAA Spaceflight Mechanics Meeting*, AAS 15-413, Williamsburg, VA, January 2014. (Cited on page 172.)

[94] B. A. Jones, D. S. Bryant, B.-T. Vo, and B.-N. Vo. Challenges of multi-target tracking for space situational awareness. In *Proceedings of the 18th International Conference on Information Fusion*, Washington DC, USA, July 2015. (Cited on page 172.)

[95] R. Jonker and A. Volgenant. A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing*, 38(4):325–340, 1987. (Cited on page 106.)

[96] S. Julier and J. Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422, 2004. (Cited on pages 10, 14, 15, and 38.)

[97] S. J. Julier and J. K. Uhlmann. New extension of the Kalman filter to nonlinear systems. In *AeroSense'97*, pages 182–193. International Society for Optics and Photonics, 1997. (Cited on pages 2 and 14.)

[98] S. J. Julier and J. K. Uhlmann. General decentralized data fusion with covariance intersection (CI). In D. L. Hall. and J. Llinas, editors, *Handbook of Multisensor Data Fusion*. CRC Press, 2001. (Cited on page 200.)

[99] S. J. Julier and J. K. Uhlmann. Using covariance intersection for SLAM. *Robotics and Autonomous Systems*, 55(1):3–20, 2007. (Cited on page 200.)

[100] R. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME - Journal of Basic Engineering*, 82(Series D):35–45, 1960. (Cited on pages 2, 10, and 11.)

[101] D. J. Kershaw and R. J. Evans. Optimal waveform selection for target tracking. *IEEE Transactions on Information Theory*, 40(5):1536–1550, September 1994. (Cited on pages 6 and 176.)

[102] Z. Khan, T. Balch, and F. Dellaert. Multitarget tracking with split and merged measurements. In *IEEE Conference Computer Vision and Pattern Recognition*, 2005. (Cited on page 87.)

[103] J. Kingman. *Poisson Processes*. Oxford studies in probability. Oxford University Press, 1993. (Cited on page 168.)

[104] T. Kirubarajan, Y. Bar-Shalom, and K. R. Pattipati. Multiassignment for tracking a large number of overlapping objects. *IEEE Transactions on Aerospace and Electronic Systems*, 37(1):2–21, January 2001. (Cited on page 87.)

[105] M. Klaas, M Briers, N. De Freitas, A. Doucet, S. Maskell, and D. Lang. Fast particle smoothing: If I had a million particles. In *Proceedings of the 23rd international conference on Machine learning*, pages 481–488. ACM, 2006. (Cited on page 18.)

[106] J. W. Koch. Bayesian approach to extended object and cluster tracking using random matrices. *IEEE Transactions on Aerospace and Electronic Systems*, 44(3): 1042–1059, July 2008. (Cited on pages 124 and 132.)

[107] J. W. Koch and M. Feldmann. Cluster tracking under kinematical constraints using random matrices. *Robotics and Autonomous Systems*, 57(3):296–309, March 2009. (Cited on page 125.)

[108] W. Koch and G. van Keuk. Multiple hypothesis track maintenance with possibly unresolved measurements. *IEEE Transactions on Aerospace and Electronic Systems*, 33(3):883–892, July 1997. (Cited on pages 87 and 102.)

[109] A. N. Kolmogorov. Interpolation and extrapolation of stationary random sequences. *Izv. Akad. Nauk SSSR Ser. Mat*, 5(1):3–14, 1941. Translated by W. L.

Doyle and Ivan Selin, Rept. RM-3090-PR, RAND Corp, Santa Monica, California, 1962. (Cited on page 2.)

[110] C. Kreucher, K. Kastella, and A. O. Hero. Multi-target sensor management using alpha-divergence measures. In *Information Processing in Sensor Networks*, pages 209–222. Springer, 2003. (Cited on page 6.)

[111] C. Kreucher, K. Kastella, and A. O. Hero III. Sensor management using an active sensing approach. *Signal Processing*, 85(3):607–624, March 2005. (Cited on page 176.)

[112] V. Krishnamurthy and R. J. Evans. Hidden Markov model multiarm bandits: A methodology for beam scheduling in multitarget tracking. *IEEE Transactions on Signal Processing*, 49(2):2893–2908, December 2001. (Cited on page 176.)

[113] H. W. Kuhn. The Hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. (Cited on page 106.)

[114] T. Kurien. *Multisensor-Multitarget Tracking: Advanced Applications*, volume 1, chapter Issues in the Design of Practical Multitarget Tracking Algorithms, pages 43–83. Artech House, 1990. (Cited on page 102.)

[115] A. Lee, C. Yau, M. B. Giles, A. Doucet, and C. C. Holmes. On the utility of graphics cards to perform massively parallel simulation of advanced Monte Carlo methods. *Journal of Computational and Graphical Statistics*, 19(4):769–789, 2010. (Cited on page 18.)

[116] C. S. Lee, S. Nagappa, N. Palomeras, D. E. Clark, and J. Salvi. SLAM with SC-PHD filters: An underwater vehicle application. *IEEE Robotics and Automation Magazine*, 21(2):38–45, June 2014. (Cited on page 164.)

[117] J. J. Leonard and H. F. Durrant-Whyte. Application of multi-target tracking to sonar-based mobile robot navigation. In *Proceedings of the 29th IEEE Conference on Decision and Control*, pages 3118–3123. IEEE, 1990. (Cited on page 5.)

[118] X. R. Li and N. Li. Integrated real-time estimation of clutter density for tracking. *IEEE Transactions on Signal Processing*, 48(10):2797–2805, October 2000. (Cited on page 44.)

[119] F. Lian, C. Li, C. Han, and H. Chen. Convergence analysis for the SMC-MeMBer and SMC-CBMeMBer filters. *Journal of Applied Mathematics*, 2012. (Cited on page 29.)

[120] A. J. Lipton, H. Fujiyoshi, and R. S. Patil. Moving target classification and tracking from real-time video. In *Proceedings of the 4th IEEE Workshop on Applications of Computer Vision*, pages 8–14. IEEE, 1998. (Cited on page 5.)

[121] J. S. Liu and R. Chen. Sequential Monte Carlo methods for dynamic systems. *Journal of the American Statistical Association*, 93(443):1032–1044, 1998. (Cited on page 17.)

[122] W. S. Lovejoy. A survey of algorithmic methods for partially observed Markov decision processes. *Annals of Operations Research*, 28:47–66, 1991. (Cited on page 177.)

[123] C. Lundquist, K. Granström, and U. Orguner. Estimating the shape of targets with a PHD filter. In *Proceedings of the 14th International Conference on Information Fusion*, Chicago, IL, USA, July 2011. (Cited on pages 125, 126, and 144.)

[124] 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, February 2013. (Cited on pages 125, 126, 131, 140, 143, and 144.)

[125] D. Macagnano and G. T. F. de Abreu. Multitarget tracking with the cubature Kalman probability hypothesis density filter. In *Conference Record of the Forty Fourth Asilomar Conference on Signals, Systems and Computers*, pages 1455–1459. IEEE, 2010. (Cited on page 26.)

[126] R. Mahler. Multitarget Bayes filtering via first-order multitarget moments. *IEEE Transactions on Aerospace and Electronic Systems*, 39(4):1152–1178, October 2003. (Cited on pages 4, 19, 25, 30, 87, 125, 164, 167, and 168.)

[127] R. Mahler. Objective functions for Bayesian control-theoretic sensor management, 1: Multitarget first-moment approximation. In *Proceeding of the 2003 IEEE Aerospace Conference*, volume 4, pages 1905–1923. IEEE, 2003. (Cited on pages 6 and 178.)

[128] R. Mahler. Multitarget sensor management of dispersed mobile sensors. In D. Grundel, R. Murphey, and P. Pardalos, editors, *Theory and Algorithms for Cooperative Systems*, chapter 12, pages 239–310. World Scientific Books, 2004. (Cited on page 164.)

[129] R. Mahler. PHD filters of higher order in target number. *IEEE Transactions on Aerospace and Electronic Systems*, 43(4):1523–1543, October 2007. (Cited on pages 4, 26, 34, 44, 125, and 127.)

[130] R. Mahler. PHD filters for nonstandard targets, II: Unresolved targets. In *Proceedings of the 12th International Conference on Information Fusion*, Seattle, Washington, USA, July 2009. (Cited on pages 87 and 88.)

[131] R. Mahler. PHD filters for nonstandard targets, I: Extended targets. In *Proceedings of the 12th International Conference on Information Fusion*, 2009. (Cited on page 125.)

[132] R. Mahler. On multitarget jump-Markov filters. In *Proceedings of the 15th International Conference on Information Fusion*, pages 149–156, Singapore, July 2012. IEEE. (Cited on page 27.)

[133] R. Mahler, B.-T. Vo, and B.-N. Vo. CPHD filtering with unknown clutter rate and detection profile. *IEEE Transactions on Signal Processing*, 59(8):3497–3513, August 2011. (Cited on pages 44 and 45.)

[134] R. P. S. Mahler. Global posterior densities for sensor management. In *Proceedings of SPIE*, volume 3365, pages 252–263, 1998. (Cited on page 164.)

[135] R. P. S. Mahler. *Statistical Multisource-Multitarget Information Fusion*. Artech House, 2007. (Cited on pages 4, 18, 26, 27, 62, 87, 88, and 128.)

[136] R. P. S. Mahler. *Advances in Statistical Multisource-Multitarget Information Fusion*. Artech House, 2014. (Cited on pages 19, 26, 164, 165, 167, 168, and 170.)

[137] V. Marmarelis and T. Berger. General methodology for nonlinear modeling of neural systems with Poisson point-process inputs. *Mathematical Biosciences*, 196 (1):1–13, 2005. (Cited on page 164.)

[138] G. Matheron. *Random Sets and Integral Geometry*. John Wiley & Sons, 1975. (Cited on pages 19, 164, 165, and 166.)

[139] E. Meijering, O. Dzyubachyk, I. Smal, and W. A. van Cappellen. Tracking in cell and developmental biology. In *Seminars in cell and developmental biology*, volume 20, pages 894–902. Elsevier, 2009. (Cited on page 5.)

[140] I. Molchanov. *Theory of Random Sets*. Springer Science & Business Media, 2006. (Cited on pages 19, 164, 165, and 166.)

[141] J. Moller and R. Waagepetersen. *Statistical Inference and Simulation for Spatial Point Processes*. Chapman & Hall CRC, 2004. (Cited on pages 164, 166, 167, and 168.)

[142] G. E. Monahan. A survey of partially observable Markov decision processes: Theory, models and algorithms. *Management Science*, 28(1):1–16, 1982. (Cited on page 177.)

[143] P. Del Moral, A. Doucet, and A. Jasra. Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3):411–436, 2006. (Cited on page 18.)

[144] P. Del Moral, A. Doucet, and A. Jasra. On adaptive resampling strategies for sequential Monte Carlo methods. *Bernoulli*, 18(1):252–278, 2012. (Cited on page 18.)

[145] J. Mullane, B.-N. Vo, M. Adams, and B.-T. Vo. A random-finite-set approach to Bayesian SLAM. *IEEE Transactions on Robotics*, 27(2):268–282, April 2011. (Cited on page 164.)

[146] L. M. Murray, A. Lee, and P. E Jacob. Parallel resampling in the particle filter. *arXiv preprint arXiv:1301.4019*, 2013. (Cited on page 18.)

[147] K. G. Murty. An algorithm for ranking all the assignments in increasing order of cost. *Operations Research*, 16(3):682–678, 1968. (Cited on pages 67, 103, and 106.)

[148] D. Musicki and R. Evans. Joint integrated probabilistic data association: JIPDA. *IEEE Transactions on Aerospace and Electronic Systems*, 40(3):1093–1099, July 2004. (Cited on pages 4 and 44.)

[149] D. Musicki and W. Koch. Multi scan target tracking with finite resolution sensors. In *Proceedings of the 11th International Conference on Information Fusion*, Cologne, Germany, July 2008. (Cited on page 87.)

[150] D. Musicki and M. Morelande. Finite resolution multitarget tracking. In *SPIE Signal and Data Processing of Small Targets*, volume 5913, 2005. (Cited on page 87.)

[151] D. Musicki, R. Evans, and S. Stankovic. Integrated probabilistic data association. *IEEE Transactions on Automatic Control*, 39(6):1237–1241, June 1994. (Cited on pages 3 and 44.)

[152] D. Musicki, B. F. La Scala, and R. J. Evans. Multi-target tracking in clutter without measurement assignment. In *43rd IEEE Conference on Decision and Control*, 2004. (Cited on page 4.)

[153] S. C. Nardone and V. J. Aidala. Observability criteria for bearings-only target motion analysis. *IEEE Transactions on Aerospace and Electronic Systems*, 17(2): 162–166, March 1981. (Cited on page 36.)

[154] R. Niu, P. Willett, and Y. Bar-Shalom. Tracking considerations in selection of radar waveform for range and range-rate measurements. *IEEE Transactions on Aerospace and Electronic Systems*, 38(2):467–487, April 2002. (Cited on page 176.)

[155] Y. Ogata. Seismicity analysis through point-process modeling: A review. *Pure and Applied Geophysics*, 55(2-4):471–507, 1999. (Cited on page 164.)

[156] Y. Oshman and P. Davidson. Optimization of observer trajectories for bearings-only target localization. *IEEE Transactions on Aerospace and Electronic Systems*, 35 (3):892–902, 1999. (Cited on pages 6 and 176.)

[157] F. Papi, B.-N. Vo, B.-T. Vo, C. Fantacci, and M. Beard. Generalized labeled multi-Bernoulli approximation of multi-object densities. *IEEE Transactions on Signal Processing*, 63(20):5487–5497, December 2015. (Cited on pages 7, 85, 90, 95, 165, and 170.)

[158] S. A. Pasha, B.-N. Vo, H. D. Tuan, and W.-K. Ma. A Gaussian mixture PHD filter for jump Markov system models. *IEEE Transactions on Aerospace and Electronic Systems*, 45(3):919–936, 2009. (Cited on page 26.)

[159] J.-M. Passerieux and D. Van Cappel. Optimal observer maneuver for bearings-only tracking. *IEEE Transactions on Aerospace and Electronic Systems*, 34(3):777–788, 1998. (Cited on page 6.)

[160] M. K. Pitt and N. Shephard. Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association*, 94(446):590–599, 1999. (Cited on page 17.)

[161] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA, USA, 2005. (Cited on page 174.)

[162] D. B. Reid. An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control*, AC-24:843–854, December 1979. (Cited on pages 4 and 102.)

[163] S. Reuter. *Multi-object Tracking Using Labelled Random Finite Sets*. PhD thesis, Faculty of Engineering and Computer Science, Ulm University, 2014. (Cited on pages 171 and 172.)

[164] S. Reuter, B.-T. Vo, B.-N. Vo, and K. Dietmayer. The labeled multi-Bernoulli filter. *IEEE Transactions on Signal Processing*, 62(12):3246–3260, June 2014. (Cited on pages 69, 126, 148, and 165.)

[165] B. Ristic and S. Arulampalam. Bernoulli particle filter with observer control for bearings-only tracking in clutter. *IEEE Transactions on Aerospace and Electronic Systems*, 48(3):2405–2415, 2012. (Cited on page 6.)

[166] B. Ristic and J. Sherrah. Bernoulli filter for joint detection and tracking of an extended object in clutter. *IET Radar, Sonar and Navigation*, 7(1):26–35, January 2013. (Cited on page 125.)

[167] B. Ristic and B.-N. Vo. Sensor control for multi-object state-space estimation using random finite sets. *Automatica*, 46(11):1812–1818, 2010. (Cited on page 6.)

[168] B. Ristic, S. Arulampalam, and N. Gordon. *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House, 2004. (Cited on pages 2, 14, and 16.)

[169] B. Ristic, D. Clark, and B.-N. Vo. Improved SMC implementation of the PHD filter. In *Proceedings of the 13th International Conference on Information Fusion*, Edinburgh, UK, July 2010. (Cited on pages 164, 177, 178, and 179.)

[170] B. Ristic, B.-N. Vo, and D. Clark. A note on the reward function for PHD filters with sensor control. *IEEE Transactions on Aerospace and Electronic Systems*, 47(2): 1521–1529, April 2011. (Cited on page 164.)

[171] B. Ristic, D. Clark, B.-N. Vo, and B.-T Vo. Adaptive target birth intensity in PHD and CPHD filters. *IEEE Transactions on Aerospace and Electronic Systems*, 48 (2):1656–1668, April 2012. (Cited on pages 29 and 32.)

[172] A. Scheel, K. Granström, D. Meissner, S. Reuter, and K. Dietmayer. Tracking and data segmentation using a GGIW filter with mixture clustering. In *Proceedings of 17th Internataional Conference on Information Fusion*, Salamanca, Spain, July 2014. (Cited on page 148.)

[173] D. Schuhmacher, B.-T. Vo, and B.-N. Vo. A consistent metric for performance evaluation of multi-object filters. *IEEE Transactions on Signal Processing*, 56(8): 3447–3457, August 2008. (Cited on pages 38, 48, 143, and 183.)

[174] H. Sidenbladh. Multi-target particle filtering for the probability hypothesis density. In *Proceedings of the Sixth International Conference of Information Fusion*, volume 2, pages 800–806, July 2003. (Cited on page 26.)

[175] S. S. Singh, N. Kantas, B.-N. Vo, A. Doucet, and R. J. Evans. Simulation-based optimal sensor scheduling with application to observer trajectory planning. *Automatica*, 43(5):817–830, 2007. (Cited on pages 6 and 176.)

[176] D. L. Snyder, L. J. Thomas, and M. M. Ter-Pogossian. A mathematical model for positron-emission tomography systems having time-of-flight measurements. *IEEE Transactions on Nuclear Science*, 28(3):3575–3583, June 1981. (Cited on page 164.)

[177] T. L. Song and D. Musicki. Adaptive clutter measurement density estimation for improved target tracking. *IEEE Transactions on Aerospace and Electronic Systems*, 467(2):1457–1466, April 2011. (Cited on page 44.)

[178] H. Sorenson, editor. *Kalman Filtering: Theory and Application*. Piscataway,NJ: IEEE, 1985. (Cited on page 11.)

[179] H. W. Sorenson. Least-squares estimation: From Gauss to Kalman. *IEEE Spectrum*, 7(7):63–68, 1970. (Cited on page 1.)

[180] D. Stoyan and A. Penttinen. Recent applications of point process methods in forestry statistics. *Statistical Science*, 15(1):61–78, 2000. (Cited on page 164.)

[181] D. Stoyan, D. Kendall, and J. Mecke. *Stochastic Geometry and its Applications*. John Wiley & Sons, 1995. (Cited on pages 18, 19, 164, 165, 166, 167, and 168.)

[182] E. J. Sutton, T. D. Henning, B. J. Pichler, C. Bremer, and H. E. Daldrup-Link. Cell tracking with optical imaging. *European radiology*, 18(10):2021–2032, 2008. (Cited on page 5.)

[183] D. Svensson, M. Ulmke, and L. Danielsson. Joint probabilistic data association filter for partially unresolved target groups. In *Proceedings of the 13th International Conference on Information Fusion*, Edinburgh, UK, July 2010. (Cited on page 86.)

[184] D. Svensson, M. Ulmke, and L. Hammarstrand. Multitarget sensor resolution model and joint probabilistic data association. *IEEE Transactions on Aerospace and Electronic Systems*, 48(4):3418–3434, October 2012. (Cited on pages 86, 89, and 102.)

[185] A. Swain and D. Clark. Extended object filtering using spatial independent cluster processes. In *Proceedings of the 13th International Conference on Information Fusion*, Edinburgh, UK, July 2010. (Cited on page 125.)

[186] A. Swain and D. Clark. The PHD filter for extended target tracking with estimable shape parameters of varying size. In *Proceedings of 15th International Conference on Information Fusion*, Singapore, July 2012. (Cited on page 125.)

[187] Z. Tang and U. Ozguner. Motion planning for multitarget surveillance with mobile sensor agents. *IEEE Transactions on Robotics*, 21(5):898–908, October 2005. (Cited on page 176.)

[188] R. Tharmarasa, T. Kirubarajan, M. L. Hernandez, and A. Sinha. PCRLB-based multisensor array management for multitarget tracking. *IEEE Transactions on Aerospace and Electronic Systems*, 43(2):539–555, April 2007. (Cited on page 176.)

[189] M. Ulmke, O. Erdinc, and P. Willett. Gaussian mixture cardinalized PHD filter for ground moving target tracking. In *Proceedings of the 10th International Conference on Information Fusion*. IEEE, 2007. (Cited on page 27.)

[190] A. Undurti and J. P. How. An online algorithm for constrained POMDPs. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3966–3973, 2010. (Cited on page 178.)

[191] M. Uney, D. E. Clark, and S. J. Julier. Distributed fusion of PHD filters via exponential mixture densities. *IEEE Journal of Selected Topics in Signal Processing*, 7(3):521–531, 2013. (Cited on page 200.)

[192] B.-N. Vo and W.-K. Ma. The Gaussian mixture probability hypothesis density filter. *IEEE Transactions on Signal Processing*, 54(11):4091–4104, November 2006. (Cited on pages 26 and 29.)

[193] B.-N. Vo, S. Singh, and A. Doucet. Sequential Monte Carlo implementation of the PHD filter for multi-target tracking. In *Proceedings of the 6th International Conference on Information Fusion*, pages 792–799, 2003. (Cited on page 26.)

[194] B.-N. Vo, S. Singh, and A. Doucet. Sequential Monte Carlo methods for Bayesian multi-target filtering with random finite sets. *IEEE Transactions on Aerospace and Electronic Systems*, 41(4):1224–1245, October 2005. (Cited on pages 4 and 20.)

[195] B.-N. Vo, B.-T. Vo, and D. Phung. Labeled random finite sets and the Bayes multi-target tracking filter. *IEEE Transactions on Signal Processing*, 62(29):6554–6567, 2014. (Cited on pages 87, 125, 165, 168, 169, and 171.)

[196] B.-T. Vo. *Random Finite Sets in Multi-object Filtering*. PhD thesis, School of Electrical, Electronic and Computer Engineering, University of Western Australia, 2008. (Cited on pages 26, 27, and 29.)

[197] B.-T. Vo and B.-N. Vo. The para-normal Bayes multi-target filter and the spooky effect. In *Proceedings of 15th International Conference on Information Fusion*, Singapore, July 2012. (Cited on pages 125, 145, and 151.)

[198] B.-T. Vo and B.-N. Vo. Labeled random finite sets and multi-object conjugate priors. *IEEE Transactions on Signal Processing*, 61(13):3460–3475, July 2013. (Cited on pages 5, 56, 58, 59, 60, 62, 63, 66, 87, 92, 96, 98, 103, 125, 137, 139, 165, 168, 169, 170, 171, and 173.)

[199] B.-T. Vo, B.-N. Vo, and A. Cantoni. Analytic implementations of the cardinalized probability hypothesis density filter. *IEEE Transactions on Signal Processing*, 55(7):3553–3567, July 2007. (Cited on pages 27, 29, 44, and 125.)

[200] B.-T. Vo, B.-N. Vo, and A. Cantoni. The cardinality balanced multi-target multi-Bernoulli filter and its implementations. *IEEE Transactions on Signal Processing*, 57(2):409–423, February 2009. (Cited on pages 5, 28, and 29.)

[201] E. Wan and R. Van Der Merwe. The unscented Kalman filter for nonlinear estimation. In *Proceedings of IEEE Symposium on Adaptive Systems for Signal Processing, Communications, and Control*, pages 153–158, Lake Louise, Alberta, Canada, 2000. IEEE. (Cited on page 14.)

[202] N. Whiteley and A. Lee. Twisted particle filters. *The Annals of Statistics*, 42(1): 115–141, 2014. (Cited on page 17.)

[203] W. Wieneke and J. W. Koch. Probabilistic tracking of multiple extended targets using random matrices. In *Proceedings of SPIE Signal and Data Processing of Small Targets*, Orlando, FL, USA, April 2010. (Cited on page 125.)

[204] N. Wiener. *Extrapolation, interpolation, and smoothing of stationary time series*, volume 2. MIT Press, Cambridge, MA, 1949. (Cited on page 2.)

[205] R. A. Wise and R. T. Rysdyk. UAV coordination for autonomous target tracking. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference, Keystone, CO, Aug*, pages 21–24, 2006. (Cited on page 5.)

[206] J. Y. Yen. Finding the K-shortest loopless paths in a network. *Management Science*, 17:712–716, 1971. (Cited on page 104.)

[207] T. Zajic and R. P. S. Mahler. Particle-systems implementation of the PHD multitarget-tracking filter. In *AeroSense 2003*, pages 291–299. International Society for Optics and Photonics, 2003. (Cited on page 26.)