Random Finite Set Methods for Multitarget Tracking

RANDOM FINITE SET METHODS FOR MULTITARGET TRACKING

BY

DARCY DUNNE

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

AND THE SCHOOL OF GRADUATE STUDIES

OF MCMASTER UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

Doctor of Philosophy (2013) McMaster University

(Electrical & Computer Engineering) Hamilton, Ontario, Canada

TITLE:	Random Finite Set Methods for Multitarget Tracking
AUTHOR:	Darcy Dunne
SUPERVISOR:	Dr. T. Kirubarajan
NUMBER OF PAGES:	xv, 133

Dedicated to my loving family

Abstract

The problem of multiple target tracking (MTT) is a major area that occurs in a variety of real world systems. The problem involves the detection and estimation of an unknown number of targets within a defined scenario space given a sequence of noisy, incomplete measurements. The classic approach to MTT performs data association between individual measurements and the estimated objects, then applying standard single target estimation methods on each set of track measurements. However, the measurement-to-track data association step is a computationally complex problem and commonly a main source of runtime performance issues in these algorithms.

More recently, a series of algorithms based on Random Finite Set (RFS) theory, that do not require data association, have been introduced. These algorithms recursively propagate a function that estimates the entire multitarget state. This function can be used to estimate both the number and values of the multitarget states. This thesis addresses some of the main issues involved with the RFS based methodology. It will also derive key extensions to improve various RFS based filters in order to enhance them for use in more diverse situations.

The first contribution is toward the well known track continuity issue in the Probability Hypothesis Density (PHD) filter. A solution of separating the PHD surface into

partitions that represent the individual state estimates both spatially and proportionally is described. The partitions are labeled and tracked over several time steps to form continuous track estimates. This theory is denoted as the Weight Partitioned PHD filter and multiple variants of the filter are presented.

Next, a key extension to the Multitarget Multi-Bernoulli (MeMBer) filter is derived. The extension allows for the tracking of manoeuvring targets with the MeMBer filter. A model likelihood vector is incorporated into the filter framework to estimate the probability of each motion model. Implementations in the standard linear and non-linear instances of the filter are computed.

Finally, a new linear variant of the multitarget intensity filter (iFilter) is presented. This new approach based on a Gaussian Mixture approximation provides a simpler, more efficient manner of performing multitarget tracking using the iFilter. Alternate extensions of the approach for non-linear systems are also indicated.

Each of the new algorithms presented are validated on simulated scenario data using a collection of standard multitarget tracking metrics. In each case, the methods contribute or improve on one or more of the aspects of applying multitarget tracking in a real world setting.

Acknowledgements

Above all, I would like to recognize my supervisor Dr. T. Kirubarajan for his guidance and experience. It has been both a privilege and an honour to work with one of the worlds most renowned and respected experts in the field of target tracking. Dr. Kiruba has been a constant source of support and encouragement throughout my degree. I am especially grateful for his patience with me as a part time student.

I would also like to acknowledge the Department of Electrical and Computer Engineering at McMaster University, especially the members of my supervisory committee, Dr. J. Reilly and Dr. A. Jeremic, for their guidance. I would also like to thank the students of the Estimation, Tracking and Fusion Laboratory for their knowledge and expertise. Also, to Cheryl Gies, for her extrordinary administrative support, especially towards my personal situation, as well as her vibrant enthusiasm.

Many thanks are given to Thomas Lang for initially introducing me to the world of target tracking and, more importantly, for his encouragement for me to originally pursue this degree.

Finally, I would like to thank my family for all of their love and support. My parents for teaching me the value of education and working towards your goals. To my wife, Jordan, for her tireless support, especially in correcting my papers, and to my son, Connor, my reason for seeing this through to the end.

Notation and Abbreviations

Abbreviations

CB	Cardinality Balanced
FISST	Finite Set Statistics
GM	Gaussian Mixture
iFilter	Intensity Filter
IMM	Interacting Multiple Model
JPDA	Joint Probabilistic Data Association
KF	Kalman Filter
LM	Linear Multitarget
MeMBer	Multitarget Multi-Bernoulli
MHT	Multiple Hypothesis Tracking
NN	Nearest Neighbour
OOSM	Out Of Sequence Measurements

PF	Particle Filter
pdf	probability density function
PHD	Probability Hypothesis Density
RFS	Random Finite Set
SIR	Sequence Important Resampling
SIS	Sequence Importance Sampling
SMC	Sequential Monte Carlo
TBD	Track Before Detect
WPPHD	Weight Partitioned Probability Hypothesis Density

Notation

$\mu_k^{(i,j)}$	Association factor between the i-th particle and j-th partition
$\mathcal{B}_k$	RFS of new (birth) elements at time k
$r_k^{(j)}$	Probability of existance of the j-th Bernoulli set at time k
ϕ	Clutter space
P_k	Estimate covariance matrix at time k
$\delta_\alpha(x)$	Dirac delta function

$\hat{x}_k$	Estimate vector at time k
$\mathcal{K}_k$	RFS of false measurements at time k
$\mathcal{N}(x; m, P)$	Multivariate Gaussian normal pdfover x with mean m and covariance P
$q_k^{(i)}(z_j)$	Measurement likelihood of z_j with the i-th component of a GM at time k
$\pi_k^{(i)}$	Model (regime) ID value of the i-th particle at time k
$\alpha_k^{(i,j)}$	Probability of transitioning from model i to model j at time k
$\Theta_k(x)$	RFS of the measurement on element x at time k
N_k^x	Estimated number of targets fomr the iFilter at time k
$x_k^{(i)}$	State of i-th particle at time k
$\Psi_k^{\phi\|\phi}$	Probability of remaining in the clutter space at time k
$\Psi_k^{X\|\phi}$	Probability of transitioning from clutter space to target space (birth) at time k
$D_k(x)$	First order moment (PHD) of multitarget pdf
$\Psi_k^{\phi\|X}$	Probability of transitioning from target space to clutter space (death) at time k
$\Psi_k^{X\|X}$	Probability of remaining in the target space (survival) at time k
x_k	State vector at time k

σ	Standard deviation
$\mathcal{S}_k(x)$	RFS of the survival of element x between time $k-1$ ands k
$\omega_k^{(i)}$	Weight of i-th particle at time k
X	Target space

Contents

List of Figures

Chapter 1

Introduction and Problem Statement

In this chapter, an introduction to the topic of target tracking will be described. As well, the outstanding issues associated with multitarget tracking that will be addressed in this thesis will be introduced. This will begin with a general overview of target tracking, starting with single target estimation. Next, some of the complexities involved in multiple target tracking, including the data association issue, will be examined. This will include a review of some of the classic, association based, approaches to the problem. Next, the theory of Random Finite Sets (RFS) as a solution for multiple target tracking, which does not require explicit enumeration of the measurement-to-track associations, will be introduced. Specifically, three RFS based target tracking algorithms, will be examined: the Probability Hypothesis Density (PHD) filter, the Multitarget Multi-Bernoulli (MeMBer) filter and the Intensity filter (iFilter). Finally, the contributions of this thesis, which is the enhancement of each of the aforementioned filters, will be described. The publications that have resulted

from this research will also be noted.

1.1 Target Tracking

The area of target tracking is a large and complex problem that is prevalent in a vast and diverse collection of real world systems (Blackman and Popoli, 1999). The problem involves the detection, estimation, prediction, classification, grouping, management and monitoring of one or more objects over time based on a collection of noisy, incomplete measurements (Bar-Shalom and Li, 1995). The area of target tracking has been a large and important area of study for several decades.

In its simplest form, target tracking can be represented as the single target estimation (filtering) case. The problem involves the estimation of a target state, within a dynamic system, based on a sequence of noisy measurements (Bar-Shalom *et al.*, 2002). Consider a sequence of object states $\{x_k\}$ where $k = 1, 2, \ldots$ indexes the sequence of time steps $t_0, t_1, \ldots, t_k, \ldots$. Typically, the state is a vector $x_k \in \mathbb{R}^{n_x}$ where n_x is the dimension of the state vector. The state x_k is composed of target state and dynamics parameters. For example, a state vector representing a target moving in a 2-dimensional space could be represented as follows:

$$x_k = \begin{bmatrix} \mathrm{x}_k & \dot{\mathrm{x}}_k & \mathrm{y}_k & \dot{\mathrm{y}}_k \end{bmatrix}^T \tag{1.1}$$

where $(\mathrm{x}_k, \mathrm{y}_k)$ represents the state location in the xy-plane and $(\dot{\mathrm{x}}_k, \dot{\mathrm{y}}_k)$ its velocity.

The system dynamics and measurement relationships can be described mathematically as follows:

$$x_k = f_k(x_{k-1}, \nu_k) \tag{1.2}$$

$$z_k = h_k(x_k, \omega_k) \tag{1.3}$$

where $f_k(\cdot)$ is the state motion dynamics at time k and ν_k is a random noise vector. Also, z_k is the measurement at time k and is typically a vector $z_k \in \mathbb{R}^{n_z}$ where $n_z \leq n_x$. The measurement is related to the state via the measurement transformation function $h_k(\cdot)$ where ω_k is a random noise vector.

The simplest case of the single target tracking problem described above is the linear Gaussian system. In this case, both the target motion and measurement transformation functions are linear functions F_k, H_k and both of the noise vectors ν_k, ω_k are zero mean Gaussian distributed. The linear Gaussian system is defined as follows:

$$x_k = F_k x_{k-1} + \nu_k \tag{1.4}$$

$$z_k = H_k x_k + \omega_k \tag{1.5}$$

$$\text{where } \nu_k \sim \mathcal{N}(0, Q_k)$$

$$\text{and } \omega_k \sim \mathcal{N}(0, R_k)$$

where Q_k and R_k are known as the transformation and measurement noise covariance matrices respectively.

1.1.1 Kalman Filter

When the tracking problem is described as in (1.4) and (1.5) above, an optimal estimate of the state value can be obtained (Bar-Shalom *et al.*, 2002). The recursive method that can obtain this estimate is the well-known Kalman Filter (KF). The KF computes the least squared optimal estimate of the state at each time step by representing the state probability density function (pdf) by a Gaussian with mean $\hat{x}_k$ and covariance P_k. It can recursively propagate this estimate forward using a 2 step procedure. The first step involves predicting the estimate forward using the following equations:

$$\hat{x}_{k|k-1} = F_k \hat{x}_{k-1} \tag{1.6}$$

$$P_{k|k-1} = F_k P_{k-1} {F_k}^T + Q_k \tag{1.7}$$

The second step involves updating the predicted state and covariance with the measurement z_k as follows:

$$\hat{x}_k = \hat{x}_{k|k-1} + K_k \left(z_k - H_k \hat{x}_{k|k-1} \right) \tag{1.8}$$

$$P_k = \left[I - K_k H_k \right] P_{k|k-1} \tag{1.9}$$

$$K_k = P_{k|k-1} {H_k}^T [S_k]^{-1} \tag{1.10}$$

$$S_k = H_k P_{k|k-1} {H_k}^T + R_k \tag{1.11}$$

1.1.2 Particle Filter

When the linear conditions necessary in order to use the KF are not met, other approaches, that are sub-optimal, can be applied. One such technique is known as the

particle filter (Gordon *et al.*, 1993). The particle filter (PF) is based on a Sequential Monte Carlo (SMC) approximation of the state pdf. The state pdf is approximated by a set of weighted states, known as particles, using a weighted sum of Dirac delta functions. The approximation is as follows:

$$p_k(x) \sim \left\{x_k^{(i)}, \omega_k^{(i)}\right\}_{i=1}^{L_k} \tag{1.12}$$

$$p_k(x) = \sum_{i=1}^{L_k} \omega_k^{(i)} \delta_{x_k^{(i)}}(x) \tag{1.13}$$

$$\text{where } \sum_{i=1}^{L_k} \omega_k^{(i)} = 1$$

This set of particles can be recursively propagated, similar to the KF. It is first predicted forward, then updated with a measurement in order to derive the estimate at each time step. First, for the prediction step, a new set of particles is sampled based on the previous set as follows:

$$x_{k|k-1}^{(i)} \sim q\left(x_k \middle| x_{k-1}^{(i)}\right) \tag{1.14}$$

$$\omega_{k|k-1}^{(i)} = \frac{p\left(x_{k|k-1}^{(i)} \middle| x_{k-1}^{(i)}\right)}{q\left(x_{k|k-1}^{(i)} \middle| x_{k-1}^{(i)}\right)} \omega_{k-1}^{(i)} \tag{1.15}$$

$$\text{for } i = 1, \ldots, L_{k-1}$$

where $q\left(\left|x_{k-1}^{(i)}\right.\right)$ is a known, proposal distribution and $p\left(x_{k|k-1}^{(i)} \middle| x_{k-1}^{(i)}\right)$ is the probability of a particle state at time k, given its value at time $k-1$. Next, the particle

set is updated based on a measurement z as follows:

$$x_k^{(i)} = x_{k-1}^{(i)} \tag{1.16}$$

$$\omega_k^{(i)^*} = p\left(z\middle|x_k^{(i)}\right)\omega_{k|k-1}^{(i)} \tag{1.17}$$

$$\omega_k^{(i)} = \omega_k^{(i)^*} \Big/ \sum_{i=1}^{L_k} \omega_k^{(i)^*} \tag{1.18}$$

$$\text{for } i = 1, \ldots, L_k$$

where $p\left(z\middle|x_k^{(i)}\right)$ is the measurement likelihood function. Note that the particle weights are normalized at each time step. This method is known as the Sequence Importance Sampling (SIS) particle filter (Ristic *et al.*, 2004).

After several iterations of a SIS particle filter, the majority of the weight in the particle set will typically lie in only a few particles. This problem is known as particle degradation (Ristic *et al.*, 2004). In this case, the particle set will not be a diverse representation of the state pdf since only a small number of the particles will be relevant in the approximation. In order to avoid this, the particle set can be resampled after a recursion of the filter. Resampling the particle set redistributes the weights more evenly. This technique is known as the Sequence Important Resampling (SIR) particle filter (Gordon *et al.*, 1993).

1.1.3 Multiple Model Tracking

In most real world scenarios, targets do not manoeuvre under a single motion model for the entire course of observation (Li and Jilkov, 2003; Li and Bar-Shalom, 1993). Most targets travel under a sequence of single motion patterns for short periods of time known as legs. In order to adapt to the possible change in motion pattern, single target tracking algorithms must be augmented in 2 ways. They must be able to operate with more than a single motion model. As well, they must interpret which of the models a target is currently operating under and when it might change between them.

Several methods have been derived to try to accomodate multiple models within the context of target tracking (Li and Jilkov, 2005). One approach for enabling multiple models is the Jump Markov (JM) model approach. In the JM approach a target is able to switch between models at each iteration based on a Markov Chain (on the current model) (Li *et al.*, 1999). A target is assigned a probability of operating under each of the possible motion models. At each time step these probabilities are updated based on the available parameters. In a linear JM approach, the probability of transitioning between models is constant and this results in a state transition matrix. The most well know of these methods is the Interacting Multiple Model (IMM) method (Blom, 1984).

1.2 Multitarget Tracking

The area of multiple target tracking is much more complex than the single target case. Multiple target tracking involves the estimation of a number of objects, typically unknown, in a given scenario space based on a collection of measurements (Bar-Shalom and Li, 1995). The targets are often free to enter and exit the scenario over the course of the observation period, making the number of targets a dynamic value. This gives rise to the scenario space having a probability of target birth and survival, $p_B(x)$ and $p_S(x)$, respectively. However, the real added difficulty in multitarget tracking lies within the measurement sets. First, the target measurements may also be absent from the measurement set, as a target probability of detection function $p_D(x)$ exists. As well, the target measurement value be affected by random noise, as in the single target case. Also, the measurement set may contain any number of clutter generated measurements known as false alarms. The number of false alarms is typically Poisson distributed with mean λ and the false measurements are spatially distributed based on a probability of false alarm $p_{FA}(z)$. The source of each of the individual measurements within the measurement set is not generally known.

The most complex aspect of multiple target tracking is the incompleteness of each measurement's source (Bar-Shalom and Fortmann, 1988). Once each of the measurements are associated with either individual targets or determined to be clutter, the individual target estimates can be generated by simply applying a single target tracking technique, as described in Section 1.1, to each sequence of measurements. However, the measurement source information is not typically available and must be computed manually. This is often the most complex and difficult portion of the

multiple target tracking algorithm. A wide variety of techniques have been compiled in an attempt to solve this issue, however, there is no one agreed upon best solution.

1.2.1 Classic Methods

The solution to the mutliple target tracking problem has been addressed using a variety of approaches. The classic technique is to perform measurement-to-track association to relate sequences of measurements to a target estimate, or track (Reid, 1979; Blackman, 1986). This has been done using a variety of techniques, the most common of which are described here. The simplest method is the Nearest Neighbor (NN) algorithm that associates each measurement with its closest track (Bar-Shalom and Li, 1995). Another methodology is the Joint Probabilistic Data Association (JPDA) (Fortmann *et al.*, 1980) that uses the probability of association between the measurement and a known number of targets to compute the target estimates. Finally, the Multiple Hypothesis Tracking (MHT) technique (Reid, 1979) builds hypotheses of concrete measurement-to-track association possibilities over time and determines which of the hypotheses is the most likely.

1.3 Random Finite Set Based Methods

More recently, a different approach to the multitarget tracking problem has been introduced, known as Random Finite Set (RFS) theory (Goodman *et al.*, 1997). In this theory sets are treated as having a random cardinality of finite size and the values within the set are also random variables. Using RFS theory, a framework of statistical paradigms known as Finite Set Statistics (FISST) can be derived that can be used to

perform multitarget estimation (Mahler, 2007b). This method treats both the multitarget state sets as well as the measurement sets as RFS within their respective spaces. These sets evolve in both cardinality as well as element values over time. By forming a relationship between these random sets based on the multitarget tracking model, FISST can be used to derive a full family of multitarget tracking filters (Mahler, 2001).

The multitarget framework is modeled under the RFS concept in two aspects, the random state sets and the random measurement sets. First, the random finite sequence of multitarget state sets, X_k, are recursively related as follows:

$$X_k = \left[\bigcup_{x \in X_{k-1}} \mathcal{S}_k(x) \right] \cup \mathcal{B}_k \tag{1.19}$$

where $\mathcal{S}_k(x)$ is the random set of the target element $x \in X_k$ that either survives between time $k-1$ and k and has a new random value or disappears from the set. The set $\mathcal{B}_k$ is an RFS of the new targets that appear between time steps. Next, the random finite measurement sets Z_k are related to the state sets as follows:

$$Z_k = \left[\bigcup_{x \in X_k} \Theta_k(x) \right] \cup \mathcal{K}_k \tag{1.20}$$

where $\Theta_k(x)$ is the random set of a noisy measurement of the target $x \in X_k$ if detected, or empty if the detection is missed. The set $\mathcal{K}_k$ is the RFS containing the false, clutter based measurements at time k.

A much different approach to multitarget tracking can be formed by applying FISST to the RFS configuration of the multitarget tracking problem. Instead of

making associations between measurement and tracks, then applying single target filter on each element, the paradigm of single target tracking is applied to an estimate of the full multitarget state as a whole. The entirety of the multitarget state, X_k, including all the elements within, is estimated using a single function. This function can then be predicted forward in time and updated using the full measurement set Z_k without individually associating between the individual elements within. Once an overall estimate of the multitarget state is produced, the number of elements along with their likely locations can be extracted from this multitarget state intensity function. A number of multitarget filtering algorithms that are based on RFS and FISST have been derived (Mahler, 2003; Vo *et al.*, 2009; Streit and Stone, 2008).

1.3.1 Probability Hypothesis Density Filter

PHD Theory

The PHD is an approximation the first order moment of the multitarget Bayes probability density function. The PHD function is defined on the state space and informally provides the likelihood of a given state being in the multitarget solution set. As such, it also gives the estimated mean density (size) of the multitarget solution set when integrated over any region of the state space. These two properties allow for the PHD function to estimate both the number of estimates as well as the most likely locations of these elements in the multitarget solution set. The first multitarget moment

(PHD) is defined as follows:

$$D_{\Xi}(x) = \int \delta_X(x) \cdot f_{\Xi}(X)\, \delta(X) \tag{1.21}$$
$$\text{where } \delta_X(x) = \sum_{v \in X} \delta_v(x)$$

where $f_{\Xi}(X)$ is the multitarget pdf over the finite solution set Ξ and δ is the Dirac delta function.

The PHD is not an actual probability density function as it does not necessarily integrate to 1, but at each point $x \in X$ it can be interpreted as the zero probability event that each state is an element of the solution set $Pr[x \in \Xi]$ (Mahler, 2003). Thus the integral over any given region of state space is the expected number of elements in that region. That is,

$$D_{\Xi}(x) = Pr[x \in \Xi] \tag{1.22}$$
$$\text{and } E\left[|S \cap \Xi|\right] = \int_S D_{\Xi}(x)\, dx \tag{1.23}$$

Similar to the Kalman Filter for single target filtering, the PHD can be recursively predicted forward in time and updated with a multitarget measurement set. The PHD recursive prediction equation is given in (Mahler, 2007b) as follows:

$$D_{k|k}\left(x|Z^{(k-1)}\right) = b_{k|k-1}(x) + \int F_{k|k-1}(x|y) \cdot D_{k-1|k-1}\left(y|Z^{(k-1)}\right) dy \tag{1.24}$$
$$\text{where } F_{k|k-1}(x|y) = p_S(y) \cdot f_{k|k-1}(x|y) \tag{1.25}$$

Here $b_{k+|k-1}(x)$ represents the target birth intensity, $f_{k|k-1}(x|y)$ is the single target transition likelihood and $p_S(y)$ is the target survival probability. Models for target spawning may also be included however they are not considered in the scope of this work.

The PHD update equation using the measurement set Z_k is given in (Mahler, 2007b) as follows:

$$D_{k|k}\left(x|Z^{(k)}\right) = \left(1 - p_D(x) + \sum_{z \in Z_k} \frac{p_D(x)\, L_z(x)}{\lambda c(z) + D_{k|k-1}\left[p_D L_z\right]}\right) D_{k|k-1}\left(x|Z^{(k)}\right) \tag{1.26}$$

$$\text{where } D_{k|k-1}\left[p_D L_z\right] = \int p_D(x)\, L_z(x)\, D_{k|k-1}\left(x|Z^{(k)}\right) dx$$

$$\text{and } L_z(x) = f_k(z|x)$$

Here $f_k(z|x)$ is the measurement likelihood function, $c(z)$ is the false alarm noise distribution function with a mean number of λ false alarms per scan and $p_D(x)$ is the probability of target detection.

The two most popular implementations of the PHD filter are the Sequential Monte Carlo PHD (SMC-PHD) filter (Vo *et al.*, 2005) and the Gaussian Mixture PHD (GM-PHD) filter (Vo and Ma, 2006). This thesis utilizes both variants in the implementation of Weight Partitioned PHD (WPPHD) filters thus the details of the both are outlined in the following sections.

1.3.2 SMC-PHD Filter

The SMC-PHD filter is a particle based, discrete approximation of the PHD surface. It approximates the PHD function $D_k\big(x|Z^k\big)$ using a set of weighted points $\left\{x_k^{(i)}, \omega_k^{(i)}\right\}_{i=1}^{L_k}$ similar to that used in a particle filter as follows:

$$D_k\big(x|Z^{(k)}\big) \cong \sum_{i=1}^{L_k} \omega_k^{(i)} \delta_{x_k^{(i)}}(x) \tag{1.27}$$

where L_k is the number of particles used in the approximation and the weights are scalar values, $\omega_k^{(i)} \in \mathbb{R}$.

The above equation (1.27) is only an approximation to the actual PHD equation, however it has been shown in (Clark and Bell, 2006) that the approximation converges to the actual function. The SMC-PHD allows for non-linear state transition functions and measurement models. It is typically considered to be more computationally intensive than the GM-PHD filter.

When the PHD function is approximated as in (1.27) the standard PHD prediction operation given in (1.24) can be performed as follows:

$$x_{k|k-1}^{(i)} \sim \begin{cases} q_k\Big(\cdot\Big|x_{k-1|k-1}^{(i)}, Z_k\Big) & \text{for } i = 1, \ldots, L_{k-1} \\ p_k\Big(\cdot\Big|Z_k\Big) & \text{for } i = L_{k-1}+1, \ldots, L_k \end{cases} \tag{1.28}$$

$$\omega_{k|k-1}^{(i)} = \begin{cases} \dfrac{f_{k|k-1}\Big(x_{k|k-1}^{(i)}|x_{k-1|k-1}^{(i)}\Big) p_S\Big(x_{k-1|k-1}^{(i)}\Big)}{q_k\Big(x_{k|k-1}^{(i)}, Z_k\Big)} \omega_{k-1|k}^{(i)} & \text{for } i = 1, \ldots, L_{k-1} \\ \dfrac{b_k\Big(x_{k|k-1}^{(i)}\Big)}{M_k p_k\Big(x_{k|k-1}^{(i)}|Z_k\Big)} & \text{for } i = L_{k-1}+1, \ldots, L_k \end{cases} \tag{1.29}$$

where q_k and p_k are proposal densities often chosen as the transition probability $q_k = f_k(x_k|x_{k-1})$ and normalized birth density $p_k = b_k(x)/\int b_k(x)$, respectively. The number of particles is $L_k = L_{k-1} + M_k$ with M_k as the number of new particles added at time k.

The PHD update recursion (1.26) can also be performed using the following equations:

$$x_{k|k}^{(i)} = x_{k|k-1}^{(i)} \tag{1.30}$$

$$\omega_{k|k}^{(i)} = \left[\left(1 - p_D\left(x_{k|k}^{(i)}\right)\right) + \sum_{z \in Z_k} \frac{p_D\left(x_{k|k}^{(i)}\right) f_k\left(z|x_{k|k}^{(i)}\right)}{\kappa_k(z) + C_k(z)} \right] \omega_{k|k-1}^{(i)} \tag{1.31}$$

$$\text{for } i = 1, \ldots, L_k$$

$$\text{where } C_k(z) = \sum_{i=1}^{L_k} p_D\left(x_{k|k}^{(i)}\right) f_k\left(z|x_{k|k}^{(i)}\right) \omega_{k|k-1}^{(i)}$$

$$\text{and } \kappa_k(z) = \lambda c_k(z)$$

1.3.3 GM-PHD Filter

The Gaussian Mixture PHD (GM-PHD) uses a Gaussian Mixture $\left\{m_k^{(i)}, P_k^{(i)}, \omega_k^{(i)}\right\}_{i=1}^{N}$ to approximate the first order multi-target moment of the multitarget Bayes filter. The approximation is a Gaussian summation defined as follows:

$$D_k\left(x|Z^k\right) \cong \sum_{i=1}^{J_k} \omega_k^{(i)} \cdot \mathcal{N}\left(x; m_k^{(i)}, P_k^{(i)}\right) \tag{1.32}$$

where J_k is the number of Gaussian components in the approximation. In (Clark and Bell, 2006), this approximation has been shown to converge to the actual PHD function as the number of Gaussian components goes to infinity.

Since the approximation is a Gaussian Mixture, the PHD prediction and update operations of (1.24) and (1.26) can be performed exactly on the GM components, provided the transition and measurement equations are linear (or linear approximations) as well, the probability of target survival and detection must both be constant.

Given a GM approximation as defined in (1.32) (but at time $k-1$) the PHD prediction operation described in (1.24) may be computed analytically using the following equations. First, the existing GM components are predicted forward using the following equations:

$$\omega_{k|k-1}^{(i)} = p_S \, \omega_{k-1}^{(i)} \tag{1.33}$$

$$m_{k|k-1}^{(i)} = F_{k-1} m_{k-1}^{(i)} \tag{1.34}$$

$$P_{k|k-1}^{(i)} = F_{k-1} P_{k-1}^{(i)} {F_{k-1}}^T + Q_{k-1} \tag{1.35}$$

$$\text{for } i = 1, \ldots, J_{k-1}$$

where p_S is the probability of survival, F_{k-1} is the linear state transition (approximation) and Q_{k-1} is the transitional process noise.

Then finally, the following GM components for the birth intensity are added:

$$\omega_{k|k-1}^{(J_{k-1}+j)} = \omega_{\Gamma}^{(j)} \tag{1.36}$$

$$m_{k|k-1}^{(J_{k-1}+j)} = m_{\Gamma}^{(j)} \tag{1.37}$$

$$P_{k|k-1}^{(J_{k-1}+j)} = P_{\Gamma}^{(j)} \tag{1.38}$$

$$\text{for } j = 1, \ldots, J_{\Gamma}$$

where $\left\{\omega_{\Gamma}^{(j)}, m_{\Gamma}^{(j)}, P_{\Gamma}^{(j)}\right\}_{j=1}^{J_{\Gamma}}$ is a GM set that represents the PHD birth intensity.

A predicted GM approximation can then be updated with a set of measurements Z_k exactly. The existing GM components are first treated under the assumption of a missed detection using the following equations:

$$\omega_{k|k}^{(i)} = (1 - p_D)\, \omega_{k|k-1}^{(i)} \tag{1.39}$$

$$m_{k|k}^{(i)} = m_{k|k-1}^{(i)} \tag{1.40}$$

$$P_{k|k}^{(i)} = P_{k|k-1}^{(i)} \tag{1.41}$$

$$\text{for } i = 1, \ldots, J_{k|k-1}$$

where p_D is the probability of target detection. As well, each GM component is individually updated with each measurement from the set Z_k using the following

equations:

$$\omega_{k|k}^{(i,z_j)^*} = p_D q_k^{(i)}(z_j)\,\omega_{k|k-1}^{(i)} \tag{1.42}$$

$$m_{k|k}^{(i,z_j)} = m_{k|k-1}^{(i)} + K_k^{(i)}\left(z_j - H_k^{(j)} m_{k|k-1}^{(i)}\right) \tag{1.43}$$

$$P_{k|k}^{(i,z_j)} = \left[I - K_k^{(i)} H_k^{(j)}\right] P_{k|k-1}^{(i)} \tag{1.44}$$

$$\omega_{k|k}^{(i,z_j)} = \frac{\omega_{k|k}^{(i,z_j)^*}}{\kappa(z_j) + \sum_{i=1}^{J_{k|k-1}} \omega_{k|k}^{(i,z_j)^*}} \tag{1.45}$$

$$\text{where } K_k^{(i)} = P_{k|k-1}^{(i)} H_k^{(j)^T}\left[S_k^{(i)}\right]^{-1} \tag{1.46}$$

$$S_k^{(i)} = H_k^{(j)} P_{k|k-1}^{(i)} H_k^{(j)^T} + R_k^{(j)} \tag{1.47}$$

$$\text{and } q_k^{(i)}(z_j) = \mathcal{N}\left(z_j; H_k^{(j)} m_{k|k-1}^{(i)}, S_k^{(i)}\right) \tag{1.48}$$

$$\text{for } i = 1,\ldots,J_{k|k-1} \text{ and } j = 1,\ldots,|Z_k|$$

where $H_k^{(j)}$ is the state to measurement transform matrix (approximation), $R_k^{(j)}$ is the measurement uncertainty covariance matrix and $\kappa(z_j)$ is the clutter intensity at the point of the jth measurement.

Following the above equations, the number of components in the GM will increase exponentially over time. This is computationally infeasible in most real time tracking systems. In order to reduce the computational load, component deletion and merging operations are used to reduce the number of components at each time step. Gaussian components whose weights fall below a deletion threshold are removed from the estimate completely and Gaussian components that are spatially similar can be merged (averaged) together and their weights combined.

Beyond the standard GM and SMC implementations of the PHD filter, several other implementations have been derived (Ristic *et al.*, 2010a). These include the Cardinalized PHD filter (Mahler, 2007a), which also propogates an estimate of the cardinality density along with the first order moment. As well, there have been physical space approaches taken to the PHD derivation (Erdinc *et al.*, 2006, 2009).

1.3.4 Multitarget Multi-Bernoulli Filter

MeMBer Filter Basics

A Bernoulli Random Finite Set (BRFS) is a random finite set that can be either a singleton $\{x\}$ with probability r or the empty set $\emptyset$ with probability $(1-r)$. If it is the singleton set $\{x\}$ then the value of x itself is distributed according to the pdf $p(x)$. Thus a BRFS can be represented by the pairing $\big(r, p(x)\big)$.

This definition can be extended to include random finite sets with cardinality greater than one by representing them as the union of a finite number of independent singleton BRFS. These unions are referred to as multi-Bernoulli Random Finite Sets (MBRFS):

$$X = \{x^{(1)}, \ldots, x^{(M)}\} = \bigcup_{i=1}^{M} \{x^{(j)}\}_{j=1}^{M} = \bigcup_{j=1}^{M} X^{(j)} \tag{1.49}$$

where each $X^{(j)}$ is represented by the Bernoulli set $\left(r^{(j)}, p^{(j)}(x)\right)$.

An MBRFS also describes a multitarget probability distribution f on X as follows:

$$f(X) \sim \left\{\left(r^{(j)}, p^{(j)}(x)\right)\right\}_{j=1}^{M} \tag{1.50}$$

$$\text{where } f(\emptyset) = \prod_{j=1}^{M}\left(1 - r^{(j)}\right) \tag{1.51}$$

$$\text{and } f(\{x_1, \ldots, x_n\}) = f(\emptyset)\sum_{j_i}\prod_{i=1}^{n}\frac{r^{(j_i)}p^{(j_i)}\left(x^{(i)}\right)}{1 - r^{(j_i)}} \tag{1.52}$$

$$\text{for } 1 \leq j_1 \neq \ldots \neq j_n \leq M$$

The expected cardinality of the multitarget solution set can be estimated using the following:

$$\hat{N}_{MEAN} = E\left[|X|\right] = \sum_{j=1}^{M} r^{(j)} \tag{1.53}$$

The most likely (mode) cardinality of the multitarget solution set can also be determined using the following:

$$\hat{N}_{MAX} = \arg\max_n\{p_n\} \tag{1.54}$$

$$\text{where } p_n = Pr[|X| = n]$$

$$= \begin{cases} \prod_{j=1}^{M}\left(1 - r^{(j)}\right) & n = 0 \\ \frac{1}{n}\sum_{i=1}^{n}(-1)^{i-1}p_{n-i}T(n) & n > 0 \end{cases} \tag{1.55}$$

$$\text{and } T(n) = \sum_{j=1}^{M}\left(\frac{r^{(j)}}{1 - r^{(j)}}\right)^n \tag{1.56}$$

MeMBer Filter Prediction

Assume we have the following MBRFS at time $k-1$:

$$X_{k-1} = \left\{ \left(r_{k-1}^{(j)}, p_{k-1}^{(j)}(x) \right) \right\}_{j=1}^{M_{k-1}} \tag{1.57}$$

The MBRFS can be predicted forward in time and forms a new MBRFS that is the union of the following two sets:

$$X_{k|k-1} = \left\{ \left(r_{k|k-1}^{(j)}, p_{k|k-1}^{(j)}(x) \right) \right\}_{j=1}^{M_{k-1}} \cup \left\{ \left(r_{\mathcal{B}_k}^{(j)}, p_{\mathcal{B}_k}^{(j)}(x) \right) \right\}_{j=1}^{M_{\mathcal{B}_k}} \tag{1.58}$$

Here the first set $\left\{ \left(r_{k|k-1}^{(j)}, p_{k|k-1}^{(j)}(x) \right) \right\}_{j=1}^{M_{k-1}}$ is formed from the predicting the BRFS in (1.57) ahead to time step k. The parameters of the predicted BRFS can be computed as follows:

$$r_{k|k-1}^{(j)} = r_{k-1}^{(j)} \left\langle p_{k-1}^{(j)}, p_S \right\rangle \tag{1.59}$$

$$p_{k|k-1}^{(j)}(x) = \frac{\left\langle f_{k|k-1}(x|\cdot), p_{k-1}^{(j)} p_S \right\rangle}{\left\langle p_{k-1}^{(j)}, p_S \right\rangle} \tag{1.60}$$

where p_S is the probability of survival and $f_{k|k-1}(\cdot|x)$ is the single target transition pdf between time steps $(k-1)$ and k. Also, here and throughout the thesis the notation $\langle \cdot, \cdot \rangle$ represents the integral of the product of the two functions over the state variable as follows $\langle v, h \rangle = \int v(y)\, h(y) dy$.

The second set $\left\{ \left(r_{\mathcal{B}_k}^{(j)}, p_{\mathcal{B}_k}^{(j)}(x) \right) \right\}_{j=1}^{M_{\mathcal{B}_k}}$ is an MBRFS approximation of the RFS multitarget births at time k and the values of $\left(r_{\mathcal{B}_k}^{(j)}, p_{\mathcal{B}_k}^{(j)}(x) \right)$ are typically determined

based on the parameters of the scenario birth model.

MeMBer Filter Update

Assume we have a predicted MBRFS at time $k|k-1$ in the form of (1.57). This MBRFS can be updated using a set of measurements Z_k to form a new MBRFS as the union of the following sets:

$$X_k = \left\{ \left(r_k^{(j)}, p_k^{(j)}(x) \right) \right\}_{j=1}^{M_{k|k-1}} \cup \left\{ (r_k(z), p_k(x|z)) \right\}_{z \in Z_k} \tag{1.61}$$

Here the first set $\left\{ \left(r_k^{(j)}, p_k^{(j)}(x) \right) \right\}_{j=1}^{M_{k|k-1}}$ is created from the set of undetected (legacy) MBRFS at time step $k|k-1$. They are formed using the following equations:

$$r_k^{(j)} = \frac{1 - \left\langle p_{k|k-1}^{(j)}, p_D \right\rangle}{1 - r_{k|k-1}^{(j)} \left\langle p_{k|k-1}^{(j)}, p_D \right\rangle} r_{k|k-1}^{(j)} \tag{1.62}$$

$$p_k^{(j)}(x) = \frac{1 - p_D}{1 - \left\langle p_{k|k-1}^{(j)}, p_D \right\rangle} p_{k|k-1}^{(j)}(x) \tag{1.63}$$

The second set $\left\{ (r_k(z), p_k(x|z)) \right\}_{z \in Z_k}$ is an MBRFS where each BRFS is formed by a measurement z from Z_k using the entire MBRFS at time step $k|k-1$. Each new

BRFS is created using the following equations:

$$r_k(z) = \frac{\sum_{j=1}^{M_{k|k-1}} \frac{r_{k|k-1}^{(j)} \left\langle p_{k|k-1}^{(j)}, \Psi_{k,z} \right\rangle}{1 - r_{k|k-1}^{(j)} \left\langle p_{k|k-1}^{(j)}, p_D \right\rangle}}{\kappa_k(z) + \sum_{j=1}^{M_{k|k-1}} \frac{r_{k|k-1}^{(j)} \left\langle p_{k|k-1}^{(j)}, \Psi_{k,z} \right\rangle}{1 - r_{k|k-1}^{(j)} \left\langle p_{k|k-1}^{(j)}, p_D \right\rangle}} \tag{1.64}$$

$$p_k(x|z) = \frac{\sum_{j=1}^{M_{k|k-1}} \frac{r_{k|k-1}^{(j)} p_{k|k-1}^{(j)}(x) \Psi_{k,z}(x)}{1 - r_{k|k-1}^{(j)} \left\langle p_{k|k-1}^{(j)}, p_D \right\rangle}}{\sum_{j=1}^{M_{k|k-1}} \frac{r_{k|k-1}^{(j)} \left\langle p_{k|k-1}^{(j)}, \Psi_{k,z} \right\rangle}{1 - r_{k|k-1}^{(j)} \left\langle p_{k|k-1}^{(j)}, p_D \right\rangle}} \tag{1.65}$$

$$\text{where } \Psi_{k,z}(x) = g_k(z|x)\, p_D(x)$$

Here Z_k is the set of measurements, $g_k(z|x)$ is the single target measurement likelihood function given state x, $p_D(x)$ is the probability of target detection and $\kappa_k(z)$ is the Poisson clutter intensity, all at time k.

1.3.5 Cardinality Balanced MeMBer Filter

The original MeMBer filter update equations described in (1.62)–(1.65) were derived in (Mahler, 2007b), however, it has since been shown in (Vo *et al.*, 2009) that the original update equations in (Mahler, 2007b) introduce a bias in the estimated number of targets. An alternative MeMBer recursion that reduces this bias was derived in (Vo *et al.*, 2009) using suggested replacements for the update equations (1.64) and (1.65). The replacement equations form what is referred to as the Cardinality Balanced MeMBer (CBMeMBer) filter that provides an unbiased estimate of the multitarget set's cardinality (Vo *et al.*, 2009). The replacement equations for (1.64) and (1.65)

are given in (1.66) and (1.67) below:

$$r_k(z) = \frac{\sum_{j=1}^{M_{k|k-1}} \frac{r_{k|k-1}^{(j)}\left(1-r_{k|k-1}^{(j)}\right)\left\langle p_{k|k-1}^{(j)}, \Psi_{k,z}\right\rangle}{\left(1-r_{k|k-1}^{(j)}\left\langle p_{k|k-1}^{(j)}, p_D\right\rangle\right)^2}}{\kappa_k(z) + \sum_{j=1}^{M_{k|k-1}} \frac{r_{k|k-1}^{(j)}\left\langle p_{k|k-1}^{(j)}, \Psi_{k,z}\right\rangle}{1-r_{k|k-1}^{(j)}\left\langle p_{k|k-1}^{(j)}, p_D\right\rangle}} \tag{1.66}$$

$$p_k(x|z) = \frac{\sum_{j=1}^{M_{k|k-1}} \frac{r_{k|k-1}^{(j)}}{1-r_{k|k-1}^{(j)}} p_{k|k-1}^{(j)}(x)\, \Psi_{k,z}(x)}{\sum_{j=1}^{M_{k|k-1}} \frac{r_{k|k-1}^{(j)}}{1-r_{k|k-1}^{(j)}} \left\langle p_{k|k-1}^{(j)}, \Psi_{k,z}\right\rangle} \tag{1.67}$$

CBMeMBer Filter Tracking

The CBMeMBer filter can be utilized as a full multitarget tracking solution with the inclusion of a unique track label for each of the BRFS in the MBRFS. This gives a track table that represents both the underlying existence likelihoods as well as their state pdfs. An MBRFS track table is defined as follows:

$$\mathcal{T}_k = \left\{\left(r_k^{(1)}, p_k^{(1)}(x), l_k^{(1)}\right), \ldots, \left(r_k^{(M_k)}, p_k^{(M_k)}(x), l_k^{(M_k)}\right)\right\} \tag{1.68}$$

where $l_k^{(j)}$ is the track label, $r_k^{(j)}$ is the probability of the track existence and $p_k^{(j)}(x)$ is the pdf of the track. Also, M_k is the total number of tracks at time k.

These track parameters are predicted forward and updated using the standard MBRFS equations. Labels are maintained over time to give track continuity. Typically, the legacy tracks (predicted and missed detections) retain their labels from the previous time step and new tracks (formed from birth or measurements) are assigned new, unique IDs.

Tracks may be confirmed and reported based on their likelihoods. The number of tracks to report can be determined using either (1.53) or (1.54) and the most likely tracks chosen. As well, track reporting can be done by reporting all those whose likelihoods exceed a given report threshold N_{report}. Tracks states are reported by the individual BRFS pdfs.

CBMeMBer Filter Maintenance

Based on the CBMeMBer filter prediction and update operations, the number of MBRFS (tracks) grows at each time step. This is impractical in most real time tracking systems, thus some mechanisms are used to keep the number of tracks in the track table at a reasonable level. It is suggested in (Mahler, 2007b) to use both pruning and merging to limit the total number of BRFS in the MBRFS estimate.

The first step of MBRFS maintenance is the method of pruning (or deletion). It simply consists of the removal of the BRFS that have a likelihood $r_k^{(j)}$ below a chosen deletion threshold N_{DEL}. As well, a threshold limiting the total number of BRFS in an MBRFS estimate $N_{BRFS_{max}}$ can be set where only the most likely BRFS are retained at each time step.

The second step of MBRFS maintenance, known as merging, is somewhat more complicated. As given in (Mahler, 2007b), the condition for two BRFS $\left(r_k^{(i)}, p_k^{(i)}(x)\right)$

and $\left(r_k^{(j)}, p_k^{(j)}(x)\right)$ to be merged are as follows:

$$r_k^{(i)} + r_k^{(j)} < 1 \tag{1.69}$$

$$N_{merge} < p_{i,j} = \int p_k^{(i)}(x) \cdot p_k^{(j)}(x) dx \leq 1 \tag{1.70}$$

where N_{merge} is a chosen merge threshold.

The merged BRFS $\left(r_k^{(l)}, p_k^{(l)}(x)\right)$ is then formed using the following equations:

$$r_k^{(l)} = r_k^{(i)} + r_k^{(j)} \tag{1.71}$$

$$p_k^{(l)}(x) = \frac{p_k^{(i)}(x) \cdot p_k^{(j)}(x)}{p_{i,j}} \tag{1.72}$$

where $p_{i,j}$ is the merging value described in (1.70).

1.3.6 SMC-CBMeMBer Filter

The Sequential Monte-Carlo CBMeMBer (SMC-CBMeMBer) filter is a discrete approximation of the multi-Bernoulli set using several sets of weighted states (or particles) (Vo *et al.*, 2009; Yin and Zhang, 2010). Each individual Bernoulli set is approximated with its own set of weighted particles. The multi-Bernoulli RFS given in

(1.50) would be represented as follows:

$$f_k(X) \sim \left\{ \left(r_k^{(j)}, p_k^{(j)}(x) \right) \right\}_{j=1}^{M_k} \tag{1.73}$$

$$\sim \left\{ r_k^{(j)}, \left\{ x_k^{(i,j)}, \omega_k^{(i,j)} \right\}_{i=1}^{L_k^{(j)}} \right\}_{j=1}^{M_k} \tag{1.74}$$

$$\text{where } \; p_k^{(j)} \sim \left\{ x_k^{(i,j)}, \omega_k^{(i,j)} \right\}_{i=1}^{L_k^{(j)}} \tag{1.75}$$

$$\Rightarrow p_k^{(j)}(x) = \sum_{i=1}^{L_k^{(j)}} \omega_k^{(i,j)} \delta_{x_k^{(i,j)}}(x) \tag{1.76}$$

This SMC estimate of the MBRFS can be predicted and updated following the equations of the CBMeMBer filter. The details of the SMC-CBMeMBer operations are omitted for brevity as similar details are given for the SMC-MM-CBMeMBer below. Details of the SMC-CBMeMBer filter derivation and steps can be found in (Vo *et al.*, 2009).

1.3.7 GM-CBMeMBer Filter

The Gaussian Mixture CBMeMBer filter (GM-CBMeMBer) is a closed form approximation solution to the CBMeMBer filter prediction and update recursion equations. The solution is based on a Gaussian Mixture approximation of the MBRFS representation of a multitarget pdf. That is,

$$f_k(X) \sim \left\{ \left(r_k^{(j)}, p_k^{(j)}(x) \right) \right\}_{j=1}^{M_k} \tag{1.77}$$

$$= \left\{ r_k^{(j)}, \left\{ \omega_k^{(i,j)}, m_k^{(i,j)}, P_k^{(i,j)} \right\}_{i=1}^{J_k^{(j)}} \right\}_{j=1}^{M_k} \tag{1.78}$$

Here, each BRFS pdf $p_k^{(j)}(x)$ is approximated by a Gaussian Mixture, $\left\{\omega_k^{(i,j)}, m_k^{(i,j)}, P_k^{(i,j)}\right\}_{i=1}^{J_k^{(j)}}$, as follows:

$$p_k^{(j)}(x) = \sum_{i=1}^{J_k^{(j)}} \omega_k^{(i,j)} \mathcal{N}\left(x; m_k^{(i,j)}, P_k^{(i,j)}\right) \tag{1.79}$$

The solution is limited to a class of linear Gaussian multitarget tracking models where the main components of the model, such as state transition and observation, are linear Gaussian. The target birth models BRFS must each be a Gaussian Mixture model. As well, target parameters for state survival and detection are each state independent.

The predicted and update equations of the CBMeMBer filter can be applied to a GM-MBRFS as defined above when under the stated conditions. The details of these GM-CBMeMBer operations are again not given here but similar details are given for the GM-MM-CBMeMBer below. Details of the GM-CBMeMBer filter derivation can be found in (Vo *et al.*, 2009).

The GM-CBMeMBer implementation can also be extended to implement non-linear motion and measurement models. The Extended Kalman Filter (EKF) or the Unscented Kalman Filter (UKF) can used as the mechanism for estimate propagation and measurement conversion in CBMeMBer filter prediction and update equations, respectively.

1.3.8 Intensity Filter

The intensity filter is a multiple target filtering solution based on the Poisson Point Processes (Streit and Stone, 2008). It has been shown to be a generalized form of the PHD filter (Streit, 2009), as the PHD can be interpreted as the intensity function of a PPP. Previously, a non-linear implementation of the iFilter has been implemented using a Sequential Monte Carlo (SMC) approximation of the intensity set (Schikora *et al.*, 2011). As well, a mulitple sensor version has been derived (Streit, 2008).

Poisson Point Processes

The theory of PPP, which has been used to solve a diverse variety of problems, forms the basis of the iFilter. Only the basics of PPP theory are needed to derive intensity filters. Further details of PPP theory can be found in (Kingman, 1993) and (Streit, 2010). A PPP is a stochastic process that can be represented by a non-negative intensity function $g(s)$ over the set S where:

$$\int_S g(s) < \infty \tag{1.80}$$

The PPP is said to be homogenous on S when the function is constant over all of S. Given a PPP $g(s)$ as above, a single realization consists of a finite number and locations of a set of points in S. These realizations can be sampled in a two step procedure. First, the number of points is sampled from the discrete Poisson

cardinality distribution.

$$Pr[n] = \exp\left(-\int_S g(s)\ ds\right)\frac{(g(s)\ ds)^n}{n!} \tag{1.81}$$

Note that the expected number of points is:

$$E(n) = \int_S g(s)\ ds \tag{1.82}$$

Next the points are independently sampled from a probability density function (pdf) defined as the normalized intensity function as follows:

$$p_g(s) = \frac{g(s)}{\int_S g(s)\ ds} \tag{1.83}$$

The total set of possible realizations of the PPP is as follows:

$$\mathcal{E}(S) = \{(0)\} \cup \{(n, \{x_1, \ldots, x_n\})\,; x_i \in S\}_{n=1}^{\infty} \tag{1.84}$$

The probability of each realization $\xi = (n, \{x_1, \ldots, x_n\}) \in \mathcal{E}(S)$ can be computed as follows:

$$p(\xi) = \frac{\exp\left(-\int_S g(s)\ ds\right)}{n!}\prod_{j=1}^{m} g(x_j) \qquad n > 0 \tag{1.85}$$

$$p(\xi) = \exp\left(-\int_S g(s)\ ds\right) \qquad n = 0 \tag{1.86}$$

Two PPP, g and h on S, are linearly superimposed if two independent realizations $(n, \{x_1, \ldots, x_n\})$ and $(m, \{y_1, \ldots, y_m\})$ are combined to form the single event $(n+m, \{x_1, \ldots, x_n, y_1, \ldots, y_m\})$. A PPP in which the same event is probabilistically

equivalent event can also be formed by superimposing the two indidivdual PPPs. This results in a PPP whose intensity is the sum of the two individual intensity functions.

iFilter Basics

Multitarget tracking can be modeled as the sequence of two PPP. Consider a set of discrete time steps $t_0, t_1, \ldots, t_k$. First, the sequence of multitarget states is approximated by a sequence of PPP $X_0, X_1, \ldots, X_k$. Under this approximation, the sequence of possible measurements also becomes a PPP. The realization of this PPP sequence is the sequence of received measurements $Z_1, \ldots, Z_k$. Also, the multitarget state space is an augmented space $S^+ = S \cup S_\phi$ where S_ϕ is the null or clutter target space and S is the state space of targets, typically $S \subseteq \Re^n$. This augmented state models both the targets states and the measurement clutter space simultaneously.

The basis for the iFilter is formulated using these two PPP. Consider the PPP at time $k-1$ denoted as X_{k-1}. This PPP can be represented by its intensity function $f_{k-1}(x)$. By definition the expected number of targets at this time step can be computed using (1.82) and the target locations estimated by sampling the pdf (1.83) where $g = f_{k-1}$ in both equations. As well, the intensity of the clutter space is represented by an intensity function $f_{k-1}(\phi)$. In many implementations this intensity function is defined as a homogenous PPP (Schikora *et al.*, 2011) and only the overall intensity volume is maintained. These intensities can then be predicted and updated recursively to form the iFilter. Each iteration of this recursion can be broken down into the several steps.

First, the state space filter intensity is predicted forward to time step k. This is done by applying the transformations of survival fitting and target motion to each realization of the PPP. Second, the predicted measurement intensities are computed for each of the two intensities. This includes the probability of detection and measurement transformation of each realization in each of the PPP. Finally, both intensities are updated using the set of measurements Z_k received at time k.

1.3.9 SMC-iFilter

The Sequential Monte Carlo (SMC) based implementation of the iFilter (SMC-iFilter) was derived in (Schikora *et al.*, 2011). The SMC-iFilter is formulated by approximating the target space intensity function by a set of weighted particles.

$$f_k(x) \sim \left\{ x_k^{(i)}, \omega_k^{(i)} \right\}_{i=1}^{L_k} \tag{1.87}$$

$$f_k(x) = \sum_{i=1}^{L_k} \omega_k^{(i)} \delta_{x_k^{(i)}}(x) \tag{1.88}$$

where $\delta_\alpha(x) = \delta(x - \alpha)$ is the Dirac delta function and the weights are scalar values, $\omega_k^{(i)} \in \mathbb{R}$. The number of targets can be computed as follows:

$$f_k(x) = \sum_{i=1}^{L_k} \omega_k^{(i)} \tag{1.89}$$

and the state estimates can be sampled by using a point clustering technique.

In (Schikora *et al.*, 2011), the intensity function of the clutter space is set as a homogenous PPP and only the total clutter intensity, $f(\phi)$, is measured over time. These

intensity functions are predicted and updated following the steps outlined above. This is done mainly using techniques similar to those in the particle filter prediction and update equations (Ristic *et al.*, 2004). The particle set is resampled for maintenance at each time step. The number of targets and the individual target state estimates can then be extracted using techniques similar to those used in SMC implementations of the PHD filter (Vo *et al.*, 2005). For brevity, the details of each of these steps of the SMC-iFilter are not given in this chapter but can be found in (Schikora *et al.*, 2011).

1.4 Contributions

1.4.1 Motivation

Most multitarget tracking systems in use today still use a classical, association based technique such as MTT, JDPA and even NN. One of the main reasons for this is the maturity of the association based algorithms. Since these techniques have been in place for several decades there has been ample amount of research performed in order to extend, enhance and optimize the association based methodology. Some of the important aspects to multiple target tracking which have been addressed in the classic methods are track smoothing, maneuvering targets, track continuity, computational optimization in linear conditions, multiple sensors, track before detect (TBD), extended (grouped) targets, dim targets, classification, out of sequence measurements (OOSM) and clutter estimation. The classical techniques have also been successfully applied to a wide variety of multitarget tracking cases involving various sensors suites and targets sets. Some of these scenarios include bearings only multitarget tracking,

multistatic tracking, tracking air targets, tracking multiple ground targets, low power radar tracking and many more.

The idea of using RFS based techniques for multitarget tracking is still a fairly, new concept. The RFS based methods described above have only recently been derived. Very few, if any, have been implemented in actual live, real world tracking systems. One of the main reason for this is that they lack the dynamic and robust enhancements which have been incorporated into the association based methods. As well, the RFS based methods have not yet been proven viable on a diverse set of scenarios in comparison to the classic methods. This is not because of a lack of capability or research but more due to their younger maturity levels and lack of exposure.

Its clear from the current research that there are several advantages to using RFS techniques to perform multiple target tracking. In order to facilitate their transition from simple theoretical solutions into full, robust target tracking algorithms capable of being used in actual tracking systems, there is a large amount of technical capabilities and enhancements which must be derived. Some of the enhancements have begun to be dervied for some of the RFS implmentations (Mahler *et al.*, 2012; Vo *et al.*, 2011; Punithakumar *et al.*, 2005; Nadarajah *et al.*, 2011). The RFS based method have begun to be validated in a variety of multitarget tracking scenarios (Vo *et al.*, 2010; Tobias and Lanterman, 2004; Tobias, 2006). These new improvements will make the RFS family of algorithms a more suitable choice for implementation in a real world multitarget tracking system.

1.4.2 Problem Statement

In this chapter, three different RFS based multitarget tracking techniques have been presented. While each has been shown to be a viable multitarget tracking filter under some basic conditions, each filter is individually lacking some capabilities which are required in most real world tracking systems. The PHD filter, in its most basic form, only gives an estimate of the current multitarget state. Its does not relate the individual element over multiple scans. This is known as the track continuity issue. The MeMBer filter has previous only been derived to handle targets which travel under a single motion model throughout the surveillance time and do not manoeurve. The iFilter has only been derived using a Monte Carlo approximation, which is typically computationally intensive and only required under highly non-linear conditions. Each of these issues and deficiencies restrict the usage and flexibility of these methods as viable algorithms for many multitarget tracking systems and they will be addressed in this thesis in order to enhance each filter.

1.4.3 Contributions

This thesis advances the research and capabilities of several RFS based tracking techniques. Specifically, it contributes three distinct improvements to the three individual RFS based filtering algorithms described in Section 1.3. Each contribution improves an important aspect of multitarget tracking and makes the filter more suitable for use in more complex tracking problems. The first contribution is the improvement of the track continuity issue in the PHD filter via the introduction of the Weight Partitioned PHD filter. The second contribution is the extension of the MeMBer filter for

tracking maneuvering targets. The third contribution is the derivation of a Gaussian Mixture based implementation of the iFilter.

Weight Partitioned PHD Filter

In Chapter 2, the track continuity issue of the PHD filter will be addressed by introducing the Weight Partitioned PHD filter. The WPPHD filter decomposes the original PHD surface into subsurfaces (or partitions) that represent the PHD of the individual singleton values of the multitarget state estimate. Each partition is created using both spatial and weight based criteria so that it is representative of a single target. These partitions are typically formed by assigning a portion of the weight of the components supporting the PHD surface to each of the partitions. These portions are held in a vector with each component and can transferred as part of the PHD recursion. This allows the partitions and their associated tracks to be continually measured over time. Each partition is also assigned a unique label associating it with a track. Two distinct implementations of the PHD filter are derived. The first is an SMC based method that uses the CLEAN algorithm to distribute particle weights amongst partitions in the SMC-PHD filter. The second uses the Linear Multitarget (LM) procedure to assign Gaussian components to each partition in the GM-PHD filter. Both of the methods are validated with a series of standard multitarget tracking metrics using a simulated data scenario.

MeMBer filter for Manouevring Targets

In Chapter 3, a new MeMBer filter recursion for tracking multiple targets, each traveling under multiple motion models, will be introduced. The multiple model CB-MeMBer (MM-CBMeMBer) filter presented here uses Jump Markov Models (JMM) to extend the standard CBMeMBer recursion to allow each target to transition under multiple motion models. This extension is implemented using both the SMC and GM based CBMeMBer approximations. The recursive prediction and update equations are presented for both implementations. Also, due to their design, each MM implmentation is also able to estimate the likelihood of each motion model a target is currently travelling under. Each multiple model implementation is validated against its respective standard CBMeMBer implementation, as well as against each other This validation is done using a simulated scenario containing multiple manoeuvering targets. A variety of metrics, including estimate accuracy, model detection capability and algorithm computational efficiency are used for performance evaluation. The new method is shown to improve results in several metrics with only a minor increase in computational complexity.

Gaussian Mixture Intensity Filter

In Chapter 4, a new Gaussian Mixture based implementation of the iFilter (GM-iFilter) will be presented. It is derived similar to the SMC-iFilter, but instead uses a weighted GM to approximate the target intensity function. The performance of the new filter is demonstrated against a standard GM-PHD filter using a simulated tracking scenario. Metrics used for comparison include cardinality estimates, overall tracking accuracy as well as computational complexity. The results show that the

new GM-iFilter gives a better cardinality estimate and a better overall filter accuracy without any increase in computational requirements.

1.4.4 Publications

The following publications have resulted from the research done during this thesis.

Journal Papers

Published

- Gadsden, S., Habibi, S., Dunne, D., and Kirubarajan, T. (2012). Nonlinear estimation techniques applied on target tracking problems. *Journal of Dynamic Systems, Measurement, and Control*, **134**(5), 054501

Conditionally Accepted

- Dunne, D., Chen, X., and Kirubarajan, T. (2012). Weight partitioned probability hypothesis density filters for multitarget tracking. Under 2nd Review with *Signal Processing*
- Dunne, D. and Kirubarajan, T. (2012b). Multiple model multi-bernoulli filters for manoeuvering targets. Accepted in *IEEE Transactions on Aerospace and Electronic Systems*

In Preparation

- Dunne, D. and Kirubarajan, T. (2013). Gaussian mixture intensity filter. To be submitted to *IEEE Transactions on Aerospace and Electronic Systems*

Conferences Papers

- Dunne, D., Ratnasingham, T., Lang, T., and Kirubarajan, T. (2009). SMC-PHD-based multi-target tracking with reduced peak extraction. In *Proc. SPIE 7445-Signal Processing, Sensor Fusion, and Target Recognition XX*
- Dunne, D. and Kirubarajan, T. (2012a). MeMBer filter for manoeuvring targets. In *Proceedings of SPIE Vol. 8392: Signal Processing, Sensor Fusion, and Target Recognition XXI*
- Dunne, D. and Kirubarajan, T. (2011). Weight partitioned probability hypothesis density filters. In *Proceedings of the 14th International Conference on Information Fusion (FUSION) 2011*, pages 1–8
- Gadsden, S., Dunne, D., Habibi, S., and Kirubarajan, T. (2011b). Combined particle and smooth variable structure filtering for nonlinear estimation problems. In *Information Fusion (FUSION), 2011 Proceedings of the 14th International Conference on*, pages 1 –8
- Gadsden, S., Dunne, D., Tharmarasa, R., Habibi, S., and Kirubarajan, T. (2011a). Application of the smooth variable structure filter to a multi-target tracking problem. In *Proceedings of SPIE Vol. 8050: Signal Processing, Sensor Fusion, and Target Recognition XX*
- Gadsden, S., Dunne, D., Habibi, S., and Kirubarajan, T. (2009). Comparison of extended and unscented Kalman, particle, and smooth variable structure filters on a bearing-only target tracking problem. In *Proceedings of SPIE Vol. 7445: Signal and Data Processing of Small Targets*
- McDonald, M., Dunne, D., Damini, A., and Kirubarajan, T. (2009). Event-based characterization and simulation of sea clutter. In *Proceedings of SPIE*

7445: Signal and Data Processing of Small Targets

Chapter 2

Weight Partitioned Probability Hypothesis Density Filter

2.1 Introduction

The PHD filter provides an estimate of the number of elements present as well as a multimodal surface with the most likely locations of those elements. The states are typically estimated based on spatial qualities (choosing higher or spaced peaks) of the PHD surface and do not necessarily consider the weight of the elements being extracted (Clark and Bell, 2007)(Panta *et al.*, 2006)(Erdinc *et al.*, 2005). If the peaks extracted from the PHD surface are estimates of individual targets, the total weight (volume) of each estimate should be representative of an individual target as well.

Several solutions exist for determining track continuity in PHD filters. One solution is to determine the track continuity external to the PHD framework. This is most typically accomplished by treating elements of the multitarget estimates from

the PHD filter as measurement inputs into another multitarget filter mechanism, such as MHT or JPDA (Lin *et al.*, 2004)(Panta *et al.*, 2007)(Wang *et al.*, 2008). Here, the PHD filter essentially acts as a decluttering tool for the external filter. The data association problem, which is so graciously circumvented in the PHD filter, is simply moved outside the framework into an external algorithm, albeit with reduced complexity.

Another method that is used to provide track continuity with the PHD filter is commonly referred to as track labeling. This method applies discrete track labels to various portions of the PHD surface to identify them with an individual state element. These labels are propagated and maintained over time to continually monitor estimates. Labeling is accomplished in the two implementations of the PHD filter described by assigning the labels to individual particles in the SMC-PHD (Clark and Bell, 2005) and the Gaussian components in the GM-PHD filters (Panta *et al.*, 2006) respectively. A variety of ad-hoc track labeling maintenance schemes are used in each to determine how the labels are propagated, updated and extracted from the filter estimates (Clark and Bell, 2007)(Wood *et al.*, 2010).

This chapter introduces a new variant of the PHD labeling technique. It partitions the PHD surface based on both weight and spatial characteristics into portions that represent the individual singleton elements making up the estimate of the multitarget solution set. These partitions can be assigned labels and propagated over time to obtain a continuous target estimate through consecutive PHD steps. This provides a solution to the PHD track continuity problem. This method is applied

and demonstrated using two unique implementations. The new techniques are compared to currently known PHD track labeling solutions. The method aims to reduce track fragmentation and increase track continuity versus previous labeling methods used with the PHD filter. The partitioning in the SMC implementation is computed using a new extension to the image decluttering algorithm known as CLEAN, which will also be described in this chapter. The partitioning for the GM implementation will be handled using Linear Multitarget (LM) assignment to update partition label likelihoods. This chapter further expands upon the ideas and implementations given in (Dunne and Kirubarajan, 2011).

This chapter is outlined as follows. First the WPPHD concept and theory are introduced in Section 2.2. Next, the SMC implementation (including the CLEAN algorithm) is given in Section 2.3. The GM implementation is then introduced in Section 2.4. In Section 2.5 both implementations are demonstrated on a simulated multitarget tracking scenario.

2.2 Weight Partitioned PHD Filter

The PHD filter provides a single function that represents the density of all targets collectively within the solution set. These targets must be individually extracted from this function, however they are predicted and updated collectively, an entire state intensity estimate using the PHD equations (1.24) and (1.26). The proposed Weight Partitioned PHD filter extracts individual partitions representing targets based on both weight and spatial characteristics. The estimates can easily generate an estimate of an individual target's probability density function. The partitions are also

uniquely labeled so they may be identified and maintained as the PHD estimate is propagated over time.

2.2.1 WPPHD Basics

The multitarget solution or estimate set can be written as the union of the individual singleton elements in that set. That is,

$$\hat{X} = \{\hat{x}_1, \ldots, \hat{x}_n\} = \bigcup_{j=1}^{n} \{\hat{x}_j\} \tag{2.1}$$

The proposed Weight Partitioned PHD filter attempts to partition the PHD surface such that each partition represents the PHD of each of the singletons $\{\hat{x}_i\}$ in the multitarget set estimate. The PHD of a union of independent sets is the sum of the PHD's of the individual sets. In the WPPHD filter, the PHD surface is decomposed into the sum of the PHD functions of the singleton sets of individual elements of the multitarget estimate. That is,

$$D(x) = \sum_{j=1}^{\hat{n}} D^{(j)}(x) \tag{2.2}$$

where here $D^{(j)}(x)$ is the PHD partition for the singleton $X^{(j)} = \{x_j\}$.

Since each partition $D^{(j)}(x)$ represents the PHD of a singleton set and is itself a PHD surface, the expected volume of each individual partition is expected to be

unity. Then,

$$\hat{N}^{(j)} = \int D^{(j)}(x) \approx 1 \tag{2.3}$$

$$\text{where } \hat{N} = \int D\,(x) = \int \sum_{j=1}^{\hat{n}} D^{(j)}(x)\; dx \tag{2.4}$$

$$= \sum_{j=1}^{\hat{n}} \int D^{(j)}(x)\; dx = \sum_{j=1}^{\hat{n}} \hat{N}^{(j)} \tag{2.5}$$

Also, since the partition of the PHD represents a singleton, the probability distribution function of the individual set element can be estimated from the partition PHD function as follows:

$$D^{(j)}(x) \approx Pr\big[X^{(j)} = x\big] \tag{2.6}$$

$$Pr\big[X^{(j)} = x\big] \approx p^{(j)}(x) = D^{(j)}(x) \Big/ \int D^{(j)}(x)\; dx \tag{2.7}$$

Each of the partitions represents an individual element (singleton) of the multistate estimate set. In order to maintain an estimate of an individual object over time, each partition is assigned a unique ID that is propagated along with the PHD surface over time. This enables the changes in each partition to be monitored over time and easily extracted at the following time step. New partitions may be added to the WPPHD in both the predictions and update stages.

Partition Maintenance

Labeling is a popular method of distinguishing individual objects within the entire PHD estimate (Clark and Bell, 2007)(Panta *et al.*, 2006). Labels are applied to various portions of the PHD surface, which essentially partitions the surface into smaller individual PHD functions. Since the PHD surface is the sum of the individual partitions, propagating the entire surface forward in time while keeping the labels intact will result in a partitioned PHD surface where each partition is its predicted and updated predecessor.

In most common implementations (Clark and Bell, 2007)(Lin *et al.*, 2006)(Zhu *et al.*, 2011), the labels are added and maintained based on information such as their spatial proximity to other labels (clustering in the SMC-PHD filter and merging in the GM-PHD filter) (Clark and Bell, 2007)(Panta *et al.*, 2006). However, it is also important to consider the total weight each individual partition forms. Most importantly, if the purpose of each partition is to identify individual targets then the total weight of each partition should also be representative of such a target. For instance, if a labeled partition has a total weight close to 2 then it is likely representative of two actual targets and should be split into smaller partitions with weights closer to 1. As well, extremely low weighted partitions (near 0) do not contain enough weight to accurately represent a single target and should not be reported as a track or even removed from the estimation process.

In order to ensure each partition correctly identifies an individual target, it must be maintained after each measurement update step. In order to accomplish this,

partition maintenance operations are introduced to remove low weighted estimates as well as break down larger partitions that are representative of multiple estimates. First, low weighted partitions whose total expected number of elements is below a certain deletion threshold $\hat{N}^{(j)} \leq N_{del} \sim 0$ are removed and their weights redistributed amongst the other surviving partitions. Next, any partitions that have an estimated number of targets greater than 1 are split into the appropriate number of sub-partitions. The partition splitting rule is implemented using a threshold N_{split}. If $\hat{N}^{(j)} \geq N_{\text{split}}$ then the partition j is split into $\tilde{N}^{(j)} = \text{round}(\hat{N}^{(j)})$ separate partitions. Finally only partitions whose weight represents an individual target are reported in the multitarget estimate set. This is accomplished by only reporting the tracks from partitions whose weight exceed a reporting threshold $\left(\hat{N}^{(j)} \geq N_{rep}\right)$.

2.3 SMC-WPPHD

2.3.1 SMC-WPPHD Basics

The WPPHD filter can be implemented using an SMC-PHD filter. One method of doing so is by extending each weighted particle to include a vector μ, which describes the probability that a particle (state) is mapped to each current partition. That is,

$$\left\{x_k^{(i)}, \omega_k^{(i)}, \mu_k^{(i)}\right\}_{i=1}^{L_k} \tag{2.8}$$

Here, $\mu_k^{(i)}$ is a vector of size T_k where T_k is the number of current partitions. Each element $\mu_k^{(i,j)}$ of $\mu_k^{(i)}$ represents the probability the individual estimate indexed by j

exists at the point $x_k^{(i)}$. It is the proportion of particle i's weight which is allocated to partition j. That is,

$$\mu_k^{(i,j)} = Pr\Big[X_k^{(j)} = x_k^{(i)} | x_k^{(i)} \in X\Big] \tag{2.9}$$

where here $X_k^{(j)}$ is the j-th partition representing the singleton set $\left\{x_k^{(j)}\right\}$ and $x_k^{(j)}$ is a random variable in the state space X. Since the entire weight of each particle weight must map to one of the current partitions, we have

$$\sum_{j=1}^{T_k} \mu_k^{(i,j)} = 1 \quad \text{for } i = 1, \ldots, L_k \tag{2.10}$$

Thus the set of vectors $\left\{\mu_k^{(i)}\right\}_{i=1}^{L_k}$ partitions the SMC-PHD estimate $\left\{x_k^{(i)}, \omega_k^{(i)}\right\}_{i=1}^{L_k}$ into individual SMC-PHDs of the singleton sets $\left\{x_k^{(j)}\right\}$ that are determined as follows:

$$D_k^{(j)}(x) \sim \sum_{i=1}^{L_k} \mu_k^{(i,j)} \omega_k^{(i)} \delta_{x_k^{(i)}}(x) \quad j = 1, \ldots, T_k \tag{2.11}$$

Accordingly, the total weight of an individual partition $N_k^{(j)}$ can be determined using the following:

$$N_k^{(j)} = \sum_{i=1}^{L_k} \mu_k^{(i,j)} \omega_k^{(i)} \quad j = 1, \ldots, T_k \tag{2.12}$$

To predict the elements of the SMC-WPPHD forward, the particles states and weights use the same equations (1.28) and (1.29) as the standard SMC-PHD filter. This includes new particles that are added for targets birth. These new particles (and weights) form a new, uniquely labeled partition of the PHD surface. New association vectors are formed for the new particles with no association to previous partitions and

complete association to the new birth partition. The association vectors of the existing particles are passed forward with their respective particle states and weights. They are then extended in size to include the new partition but are given no association weight to the new partition. This allows each particle to retain its association with each of the partitions from the previous timestep.

$$\mu_{k|k-1}^{(i)} = \begin{cases} \left[\mu_{k-1|k-1}^{(i,1)}, , \mu_{k-1|k-1}^{(i,T_{k-1})} \; 0\right] & \text{for } i = 1, \ldots, L_{k-1} \\ \left[\underbrace{0, \; \ldots \; , 0}_{T_{k-1} \text{ times}} \; 1\right] & \text{for } i = L_{k-1}+1, \ldots, L_k \end{cases} \tag{2.13}$$

Similarly, the update step of SMC-WPPHD also uses the same update equations (1.30) and (1.31) as the SMC-PHD filter. The association vectors are simply kept the same as after the prediction step. They are instead updated during the partition maintenance step:

$$\mu_{k|k}^{(i)} = \mu_{k|k-1}^{(i)} \;\; \text{for } i = 1, \ldots, L_k \tag{2.14}$$

After prediction and measurement updates have been performed on the extended particle set, the weights of the existing partitions are computed as in (2.12) above. Once the new partition weights are computed it is important to maintain the partitions so to assure they are each still representative of singleton sets. This process follows the logic of partition deletion and splitting as outlined above in section 2.2.1. Partitions with total weight below the delete threshold N_{del} are removed and their weights redistributed amongst the surviving partitions. Then any partitions whose total weight is above the splitting threshold N_{split} are each repartitioned into the appropriate number of sub-partitions whose individual weights better approximate a singleton set. Any new partitions are given new unique labels to identify them as

separate elements in the multitarget estimate. The steps for the maintenance of the SMC-WPPHD are given in Algorithm 1.

Algorithm 1 SMC-WPPHD Maintenance

for $j = 1 \dots T_k$ **do**
  $N_k^{(j)} = \sum_{i=1}^{L_k} \mu_k^{(i,j)} \omega_k^{(i)} \quad j = 1, \dots, T_k$ {Compute the weight of each partition}
  if $N_k^{(j)} < N_{del}$ **then**
    $\mu_k^{(i,j)} = 0 \; \forall i$ {Delete low weighted partitions}
  end if
  if $N_k^{(j)} > N_{split}$ **then**
    $REPARTITION\left(\left\{x_k^{(i)}, \mu_k^{(i,j)} \cdot \omega_k^{(i)}\right\}, round(N_k^{(j)})\right)$ {Split any high weighted partitions}
  end if
end for

There are several methods that are traditionally used to extract states from a PHD filter, or in the case of the WPPHD, a partition. For most methods the first step is determining the number of states to extract using the PHD estimate of the number elements contained in the surface. Once the appropriate number of elements is determined the peaks or states extracted based on the spatial characteristics of the PHD surface. Some techniques that are used in the SMC-PHD filter context are Gaussian Mixture approximation via Expectation Maximization (Clark and Bell, 2007) or various means of clustering (Dunne *et al.*, 2009)(Punithakumar *et al.*, 2008). The majority of these methods choose the peaks and estimates based on spatial characteristics of the PHD surface but do not necessarily consider the total size or weight of the peaks such that they accurately represents an individual target. One interesting technique of PHD state extraction that has been touched upon often the literature is the CLEAN method.

2.3.2 CLEAN Algorithm

The CLEAN algorithm (Bose *et al.*, 2002)(Högbom, 1974) is an image decluttering algorithm that can be used to extract individual estimates from a multi-state estimate surface. Originating in the domain of astronomy, CLEAN algorithms are a class of deconvolution algorithms that extract multiple objects from noisy images. The attraction of CLEAN algorithms is that both the spatial characteristics as well as the size of the peaks are taken into consideration. The pseudo code for the basic CLEAN algorithm is described in Algorithm 2.

Algorithm 2 Basic CLEAN

$k \leftarrow 0$ {Let $s_0(x)$ represent the original noisy image}
while Targets remain in the noisy image $s_k(x)$ **do**
  $x_k \leftarrow \max_x s_k(x)$ {Identify brightest point in image}
  $p_k(x) \leftarrow p_0(x - x_k)$ {Create typical image at max point}
  $s_{k+1}(x) \leftarrow s_k(x) - p_k(x)$ {Remove target from noisy image}
end while
$P_k \leftarrow \{p_k(x)\}$ {Denotes the set of target peaks}

Recently, the class of CLEAN algorithms has generated interest for extracting peaks from SMC-PHD filters (Tobias, 2006)(Tobias and Lanterman, 2008)(Tang and Wei, 2010). The particular sub-class of CLEAN algorithms that had been applied to SMC-PHD state extraction is the sequential CLEAN algorithm where peaks are individually chosen and their weights removed (subtracted) from the PHD surface in sequence (Tobias and Lanterman, 2004). Each subsequent peak is then chosen from the cluttered image with the previous peak removed until all peaks are extracted.(Bose

et al., 2002)

2.3.3 CLEAN-SMC-WPPHD

The proposed partitioning procedure in the WPPHD aims to break down the PHD surface into pieces that represent individual targets. As such, the CLEAN algorithm is the perfect candidate to repartition sets into smaller portions in the SMC-WPPHD filter. The procedure is similar to the CLEAN algorithm for the SMC-PHD filter, however the portions of particle weights that are extracted for each peak are reported in order to track the amount of weight each particle contributes to each partition. The CLEAN algorithm used in this chapter for partitioning in the SMC-WPPHD filter is described in full detail in Algorithm 3. The resulting filter is denoted the CLEAN-SMC-WPPHD.

The corresponding equations for pdfs of SMC-WPPHD partition elements $p_k^{(j)}(x)$ along with their Gaussian approximations with mean $m_k^{(j)}$ and covariance $P_k^{(j)}$ are computed as follows:

$$p_k^{(j)}(x) \cong \frac{\sum_{i=1}^{L_k} \delta_{x_k^{(i)}}(x)}{\sum_{i=1}^{L_k} \mu_k^{(i,j)} \omega_k^{(i)}} \tag{2.15}$$

$$m_k^{(j)} = \frac{\sum_{i=1}^{L_k} \mu_k^{(i,j)} \omega_k^{(i)} x_k^{(i)}}{\sum_{i=1}^{L_k} \mu_k^{(i,j)} \omega_k^{(i)}} \tag{2.16}$$

$$P_k^{(j)} = \frac{\sum_{i=1}^{L_k} \sum_{i=1}^{L_k} \left(x_k^{(i)} - m_k^{(j)} \right) \left(x_k^{(i)} - m_k^{(j)} \right)'}{\sum_{i=1}^{L_k} \mu_k^{(i,j)} \omega_k^{(i)}} \tag{2.17}$$

$$for\ j = 1, \ldots, T_k$$

Algorithm 3 CLEAN-SMC-WPPHD Partitioning

$P_0 = diag\left[\sigma_{x_0}^2, \sigma_{\dot{x}_0}^2, \sigma_{y_0}^2, \sigma_{\dot{y}_0}^2, \right]$ {Set neighborhood to initial target neighborhood covariance}

$N_k = round\left(\sum_{i=1}^{L_k} \omega_k^{(i)}\right)$ {Compute number of targets}

$W_k = \left(\sum_{i=1}^{L_k} \omega_k^{(i)}\right) \Big/ N_k$ {Compute weight of each of target}

for $j = 1$ **to** N_k **do**

$x_{i_{MAX}}$ where $i_{MAX} = \arg_i \max \left\{\omega_k^{(i)}\right\}$ {Find largest weighted particle}

$\sigma^{(i,j)} = \left(x_k^{(i)} - x_{i_{MAX}}\right)^T {P_0}^{-1} \left(x_k^{(i)} - x_{i_{MAX}}\right)$ {Compute Mahalanobis between each particle and initial target estimate}

for $l = 0$ **to** $MaxIterations$ **do**

$\Delta_l = \left\{x_k^{(i)} \middle| \sigma^{(i,j)} \leq l\right\}$ {Find particles in neighborhood of max}

$W_{Nhood} \leftarrow \sum_{i \in \Delta_l} \omega_k^{(i)}$ {Compute weight of neighborhood}

if $W_{Nhood} \geq W_k$ **then**

$\mathcal{P}_j = \left\{x_k^{(i)}, \omega_k^{(i,j)}\right\}$ where $\omega_k^{(i,j)} = \omega_k^{(i)} \cdot \frac{W_k}{W_{Nhood}}$ {Peak j to be returned}

$\omega_k^{(i)} = \omega_k^{(i)} - \omega_k^{(i,j)}, \quad i \in \Delta_l$ {Extract peak from particle set}

end if

end for

end for

$\mu_k^{(i,j)} = \omega_k^{(i,j)} / \sum_{j=1}^{N_k} \omega_k^{(i,j)}$ {Compute the weight proportion contributed to each partition}

$\mu_k^{(i)} = \left[\mu_k^{(i,j)}\right]_{j=1}^{N_k}$

2.4 LM-GM-WPPHD

In order to show the versatility of implementations of the WPPHD framework, a completely different approach is taken for the implementation of the WPPHD with the GM-PHD filter. In the proposed GM method, the GMPHD function is also decomposed into several partitions where each partition still represents a possible track. Here, a single Gaussian component in the GM-PHD filter cannot be interpreted as a single point in the state space because a Gaussian component already has a mean and covariance. Thus, each Gaussian component can only belong to a single partition, however, each partition can be composed of several Gaussian components. Besides the usual GMPHD filter update, the probability that the ith partition represents a true currently existing target is also recursively updated, using an algorithm based on the Linear Multitarget processing approximation proposed in (Musicki and La Scala, 2008). In LM processing approximation, when processing target i, those possible detections generated by targets other than i are treated as additional clutter measurements. The clutter intensity value used in the LM processing is increased accordingly, so the enumeration of the joint measurement–to–track assignment becomes unnecessary.

In the proposed method, denoted the LM-GM-WPPHD filter, the PHD function has been decomposed into N partitions $\{S_1, \cdots, S_N\}$ where N is the estimated number of targets, as in the WPPHD filter. The random event that the ith partition represents a currently existing target is represented by χ^i. The probability that partition S_i represents a currently existing target is $P(\chi^{(i)})$. For partition S_i, it contains n_i Gaussian components $\{\mathcal{N}_1^{(i)}, \cdots, \mathcal{N}_{n_i}^{(i)}\}$ with corresponding weights $\{w_1^{(i)}, \cdots, w_{n_i}^{(i)}\}$.

For Gaussian components $\mathcal{N}_j^{(i)}$, it has mean vector $\mu_j^{(i)}$ and covariance matrix $\Sigma_j^{(i)}$. The PHD function has the form

$$D(x) = \sum_{i=1}^{N} \sum_{j=1}^{n_i} w_j^{(i)} \mathcal{N}_j^{(i)}(x; \mu_j^{(i)}, \Sigma_j^{(i)}) \tag{2.18}$$

Assume at time k, after the PHD prediction step, the predicted PHD function is

$$D_{k|k-1}(x) = \sum_{i=1}^{N} \sum_{j=1}^{n_i} w_j^{(i)}(k|k-1) \mathcal{N}_j^{(i)}(x; \mu_j^{(i)}(k|k-1), \Sigma_j^{(i)}(k|k-1)) \tag{2.19}$$

Also at time k, after the Markov chain one propagation of the probability of target existence (Musicki and La Scala, 2008), the probability that partition S_i represents a true target existing at time k is $P_{k|k-1}(\chi^{(i)})$, $i = 1, \cdots, N$. At time k, the measurement set is $Z_k = \{z_1, z_2, \cdots, z_{N_k}\}$. In the following, the output from the GM-PHD filter and $P_{k|k-1}(\chi_i)$, $i = 1, \cdots, N$ will be used to calculate the posterior probability $P_{k|k}(\chi^{(i)})$, $i = 1, \cdots, N$.

From the measurement update step of the GM-PHD filter, $q_j^{(i)}(z)$, the predicted measurement probability density function (pdf) for Gaussian component $\mathcal{N}_j^{(i)}(x; \mu_j^{(i)}(k|k-1), \Sigma_j^{(i)}(k|k-1))$ can be obtained as in (Vo and Ma, 2006). Then, in this chapter, the following approximation is used to calculate the predicted measurement pdf of partition S_i, which is denoted as $f^{(i)}(z)$.

$$f^{(i)}(z) = \frac{\sum_{j=1}^{n_i} w_j^{(i)} q_j^{(i)}(z)}{\sum_{j=1}^{n_i} w_j^{(i)}} \tag{2.20}$$

Note, the numerator in (2.20) has already been calculated in the GM-PHD update

step (see (1.48)), so it is possible to reuse the results in the GM-PHD filter. Furthermore, it is easy to prove that $f^{(i)}(z)$ is a proper measurement pdf, i.e., $f^{(i)}(z) \geq 0$ and $\int f^{(i)}(z)dz = 1$.

Define $\rho^{(i)}(z)$ as the predicted intensity function of measurements that originated from clutter or Gaussian components belonging to partitions other than S_i. The intensity function $\rho^{(i)}(z)$ is given by

$$\rho^{(i)}(z) = \lambda(z) + \sum_{e=1,e\neq i}^{N} \sum_{j=1}^{n_e} w_j^{(e)} q_j^{(e)}(z) \tag{2.21}$$

where $\lambda(z)$ is the clutter intensity function. Note, that (2.21) can also be obtained by reusing the intermediate result in the GM-PHD filter update step.

Using (2.21), (2.20) and the Linear Multitarget processing approximation (Musicki and La Scala, 2008), the prior probability that measurement $z_j \in Z_k$ is associated with the target represented by partition S_i can be calculated as

$$P_{k|k-1}(\chi_j^{(i)}) \approx P_D^{(i)} \frac{\frac{f^{(i)}(z_j)}{\rho^{(i)}(z_j)}}{\sum_{r=1}^{N_k} \frac{f^{(i)}(z_r)}{\rho^{(i)}(z_r)}} \tag{2.22}$$

where $P_D^{(i)}$ is the detection probability for the possible target represented by partition S_i. From (2.20), (2.22) and using the LM processing approximation again, the value of the priori scatter measurement intensity for measurement z_j, which will be used to

update the probability of χ_i, is then given by

$$\tilde{\rho}^{(i)}(z_j) = \lambda(z_j) + \sum_{\tau=1,\tau\neq i}^{N} f^{(\tau)}(z_j)\frac{P_{k|k-1}(\chi^{(\tau)})}{1 - P_{k|k-1}(\chi^{(\tau)})} \tag{2.23}$$

According to LM approximations, the above equation gives an approximated value of the intensity function at measurement point z_j, conditional on z_j being a clutter or generated by some target other than the one represented by partition S_i. It is this term that provides the coupling between the updates for $P_{k|k}(\chi_i)$, $i = 1, \cdots, N$. Also it should be noted that (2.21) and (2.23) are hugely different. Equation (2.21) counters the fact that besides being a clutter, z can be a measurement caused by Gaussian components other than those associated with partition S_i. On the other hand, equation (2.23) handles the possibility that measurement z_j can be a clutter or a measurement generated by targets other than the one represented by S_i. Equations (2.21) is implicitly based on the Poisson point process assumption for the prior target random finite set, while (2.23) is based on the LM processing approximation.

From (2.23), (2.20) and the LM processing approximation used to derive (28) in (Musicki and La Scala, 2008), the posterior probability that the Gaussian component $\mathcal{N}_j^{(i)}(x; \mu_j^{(i)}(k|k-1)(0), \Sigma_j^{(i)}(k|k-1)(0))$, which is the Gaussian component obtained from $\mathcal{N}_j^{(i)}(x; \mu_j^{(i)}(k|k-1), \Sigma_j^{(i)}(k|k-1))$ by assuming that there is a missed detection, represents a track existing at time k is given by

$$P_{k|k}(\chi^{(i)}, \chi_j^{(i)}(0)) = \epsilon^{(i)} \cdot \frac{1 - P_D^{(i)}}{1 - \delta^{(i)} P_{k|k-1}(\chi^{(i)})} w_j^{(i)}(k|k-1) \tag{2.24}$$

From (2.23), (2.20) and the LM processing approximation used to derive (29) in

(Musicki and La Scala, 2008), the posterior probability that the Gaussian component $\mathcal{N}_j^{(i)}(x;\mu_j^{(i)}(k|k-1)(z_r),\Sigma_j^{(i)}(k|k-1)(z_r))$ $(z_r \in Z_k)$, which is the Gaussian component obtained by the GM-PHD filter using $\mathcal{N}_j^{(i)}(x;\mu_j^{(i)}(k|k-1),\Sigma_j^{(i)}(k|k-1))$ and z_r , represents a track existing at time k is given by

$$P_{k|k}(\chi^{(i)},\chi_j^{(i)}(z_r)) = \epsilon^{(i)} \cdot \frac{P_D^{(i)}}{1-\delta^{(i)}P_{k|k-1}(\chi^{(i)})} \cdot \frac{w_j^{(i)}(k|k-1)q_j^{(i)}(z_r)}{\tilde{\rho}^{(i)}(z_r)} \tag{2.25}$$

Because $j = 1, 2, \ldots, n_i$ the events that Gaussian component $\mathcal{N}_j^{(i)}(x;\mu_j^{(i)}(k|k-1)(z),\Sigma_j^{(i)}(k|k-1)(z))$ represents a track at time k are mutually exclusive, there is

$$P_{k|k}(\chi^{(i)}) = \sum_{j=1}^{n_i}\left(P_{k|k}(\chi^{(i)},\chi_j^{(i)}(0)) + \sum_{z_r\in Z_k} P_{k|k}(\chi^{(i)},\chi_j^{(i)}(z_r))\right) \tag{2.26}$$

The posterior probability that the track represented by partition S_i terminates at time k is then

$$P_{k|k}(\bar{\chi}^{(i)}) = \frac{1-P_{k|k-1}(\chi^{(i)})}{1-\delta^{(i)}P_{k|k-1}(\chi^{(i)})} \tag{2.27}$$

In (2.24), (2.25) and (2.27) the terms $\epsilon^{(i)}$ and $\delta^{(i)}$ are computed as follows:

$$\epsilon^{(i)} = \frac{P_{k|k-1}(\chi^{(i)})}{\sum_{j=1}^{n_i} w_j^{(i)}(k|k-1)} \tag{2.28}$$

$$\delta^{(i)} = P_D^{(i)}\left(1-\sum_{z_r\in Z_k}\frac{f^{(i)}(z_r)}{\tilde{\rho}^{(i)}(z_r)}\right) \tag{2.29}$$

The term $\epsilon^{(i)}$ is used to balance the cardinality. From the definition of the PHD function, the prior expected number of targets associated with S_i is $\sum_{j=1}^{n_i} w_j^{(i)}(k|k-1)$. On the other hand, for the target represented by partition S_i with prior probability

$P_{k|k-1}(\chi^{(i)})$, its expected number of target is $0 \cdot (1 - P_{k|k-1}(\chi^{(i)})) + 1 \cdot P_{k|k-1}(\chi^{(i)}) = P_{k|k-1}(\chi^{(i)})$. To compensate for the difference between the above two expectations, $\epsilon^{(i)}$ is introduced. The term $\delta^{(i)}$ and its equation are the same as the one given by (14) in (Musicki and La Scala, 2008).

In the following we will show that (2.26) and (2.27) define a probability measure, as long as $P_{k|k-1}(\chi^{(i)}) \in [0, 1]$.

Proof. First, it is obvious that as long as $P_{k|k-1}(\chi^{(i)}) \in [0, 1]$, $P_{k|k}(\chi^{(i)})$ and $P_{k|k}(\bar{\chi}^{(i)})$, calculated from (2.26) and (2.27), are no smaller than 0.

Now we only need to prove that $P_{k|k}(\chi^{(i)}) + P_{k|k}(\bar{\chi}^{(i)}) = 1$.

$$
\begin{aligned}
&P_{k|k}(\chi^{(i)}) + P_{k|k}(\bar{\chi}^{(i)}) \\
&= \sum_{j=1}^{n_i} \epsilon^{(i)} \cdot \frac{1 - P_D^{(i)}}{1 - \delta^{(i)} P_{k|k-1}(\chi^{(i)})} w_j^{(i)}(k|k-1) \\
&+ \sum_{j=1}^{n_i} \sum_{z_r \in Z_k} \epsilon^{(i)} \cdot \frac{P_D^{(i)}}{1 - \delta^{(i)} P_{k|k-1}(\chi^{(i)})} \cdot \frac{w_j^{(i)}(k|k-1) q_j^{(i)}(z_r)}{\tilde{\rho}^{(i)}(z_r)} + \frac{1 - P_{k|k-1}(\chi^{(i)})}{1 - \delta^{(i)} P_{k|k-1}(\chi^{(i)})} \\
&= \sum_{j=1}^{n_i} \frac{P_{k|k-1}(\chi^{(i)})}{\sum_{t=1}^{n_i} w_t^{(i)}(k|k-1)} \cdot \frac{1 - P_D^{(i)}}{1 - \delta^{(i)} P_{k|k-1}(\chi^{(i)})} w_j^{(i)}(k|k-1) \\
&+ \sum_{z_r \in Z_k} \sum_{j=1}^{n_i} \frac{P_{k|k-1}(\chi^{(i)})}{\sum_{t=1}^{n_i} w_t^{(i)}(k|k-1)} \cdot \frac{P_D^{(i)}}{1 - \delta^{(i)} P_{k|k-1}(\chi^{(i)})} \cdot \frac{w_j^{(i)}(k|k-1) q_j^{(i)}(z_r)}{\tilde{\rho}^{(i)}(z_r)} \\
&+ \frac{1 - P_{k|k-1}(\chi^{(i)})}{1 - \delta^{(i)} P_{k|k-1}(\chi^{(i)})} \\
&= \frac{P_{k|k-1}(\chi^{(i)})(1 - P_D^{(i)})}{1 - \delta^{(i)} P_{k|k-1}(\chi^{(i)})} + \frac{P_{k|k-1}(\chi^{(i)}) P_D^{(i)}}{1 - \delta^{(i)} P_{k|k-1}(\chi^{(i)})} \sum_{z_r \in Z_k} \frac{\sum_{j=1}^{n_i} w_j^{(i)}(k|k-1) q_j^{(i)}(z_r)}{\sum_{t=1}^{n_i} w_t^{(i)}(k|k-1)} \frac{1}{\tilde{\rho}^{(i)}(z_r)} \\
&+ \frac{1 - P_{k|k-1}(\chi^{(i)})}{1 - \delta^{(i)} P_{k|k-1}(\chi^{(i)})}
\end{aligned}
\tag{2.30}
$$

From (2.20), the above becomes

$$
\begin{aligned}
&P_{k|k}(\chi^{(i)}) + P_{k|k}(\bar{\chi}^{(i)}) \\
&= \frac{P_{k|k-1}(\chi^{(i)})(1 - P_D^{(i)})}{1 - \delta^{(i)} P_{k|k-1}(\chi^{(i)})} + \frac{P_{k|k-1}(\chi^{(i)}) P_D^{(i)}}{1 - \delta^{(i)} P_{k|k-1}(\chi^{(i)})} \cdots \\
&\cdots \sum_{z_r \in Z_k} f^{(i)}(z_r) \frac{1}{\tilde{\rho}^{(i)}(z_r)} + \frac{1 - P_{k|k-1}(\chi^{(i)})}{1 - \delta^{(i)} P_{k|k-1}(\chi^{(i)})}
\end{aligned}
$$

Then, we use (2.29) to replace the term $\sum_{z_r \in Z_k} f^{(i)}(z_r) \frac{1}{\tilde{\rho}^{(i)}(z_r)}$ in the above equation and get

$$
\begin{aligned}
&= \frac{P_{k|k-1}(\chi^{(i)})(1 - P_D^{(i)})}{1 - \delta^{(i)} P_{k|k-1}(\chi^{(i)})} + \frac{P_{k|k-1}(\chi^{(i)}) P_D^{(i)}}{1 - \delta^{(i)} P_{k|k-1}(\chi^{(i)})} \left(1 - \frac{\delta^{(i)}}{P_D^{(i)}}\right) + \frac{1 - P_{k|k-1}(\chi^{(i)})}{1 - \delta^{(i)} P_{k|k-1}(\chi^{(i)})} \\
&= \frac{P_{k|k-1}(\chi^{(i)})(1 - P_D^{(i)}) + P_{k|k-1}(\chi^{(i)}) P_D^{(i)} - P_{k|k-1}(\chi^{(i)}) \delta^{(i)} + 1 - P_{k|k-1}(\chi^{(i)})}{1 - \delta^{(i)} P_{k|k-1}(\chi^{(i)})} \\
&= 1
\end{aligned}
\tag{2.31}
$$

□

2.5 Simulations

2.5.1 Scenario

The performance of the each of the WPPHD based implementations is demonstrated using the following simulated example. Each implementation is compared to a standard PHD filter counterpart using a known track continuity solution. Consider the two dimensional multitarget scenario consisting of four targets as shown in Figure 2.1. Each target appears and vanishes in the scenario space at different points in time

throughout the simulation. Initial target values are given in Table 2.1. The measurements consist of the target xy-coordinate values with additive Gaussian white noise. The noise has a standard deviation of $\sigma_x = 1$m which is independent in each dimension. The sensor generates a measurement for each target with a constant probability of $p_D = 0.98$. It also generates a Poisson random number of false measurements at each time step with mean $\lambda = 1$. The spatial distribution, $c(z)$, of these false alarms is uniform throughout the measurement space. The scenario consists of 35 measurement scans, each at a constant $T = 1$s time interval. The scenario contains several difficult aspects of multitarget tracking such as crossing targets as well as nearby target births.

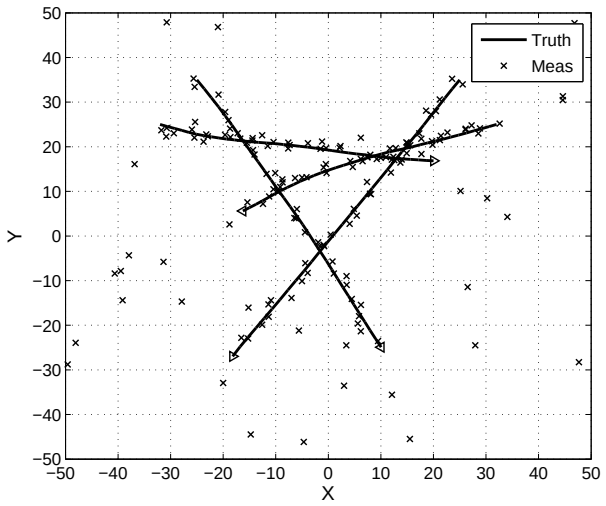

Figure 2.1: Four Target 2-D Scenario

Each WPPHD implementation (CLEAN-SMC-WPPHD and LM-GM-WPPHD) was run against their respective, standard SMC-PHD and GM-PHD filter counterparts. Each used a simple track continuity mechanism similar to those found in literature. The standard SMC-PHD implementation used a simple particle labeling

Table 2.1: Initial States of Targets in Simulation

	Position (m)	Velocity (m/s)	Starting Time (scan)
Target 1	$(-32,\ 25)$	$(1.4,\ -0.5)$	2
Target 2	$(32,\ 25)$	$(-1.4,\ -0.5)$	2
Target 3	$(-25,\ 35)$	$(1.4,\ -2)$	7
Target 4	$(25,\ 35)$	$(-1.4,\ -2)$	7

and k-means clustering mechanism (Clark and Bell, 2007). The standard GM-PHD implementation used a Gaussian component labeling and 1D update tree mechanism (Pasha *et al.*, 2006). All four filters were configured and run against the above described 2D scenario for 100 Monte Carlo runs.

The common scenario parameters (p_D, λ, etc.) were assumed to be known and set to their actual values in all four filters. The target birth density was set to be a Gaussian mixture of five Gaussian components. The components were located in each quadrant at $[\pm 25, 0, \pm, 25, 0]$ as well as one at the origin. Each Gaussian component had a covariance of $diag\left[\sigma_{x_0}^2, \sigma_{\dot{x}_0}^2, \sigma_{x_0}^2, \sigma_{\dot{x}_0}^2\right]$ where $\sigma_{x_0} = 10\text{m}$ and $\sigma_{\dot{x}_0} = 1.5\text{m/s}$ respectively. The birth density weight for each component was set to be 0.03 for a total birth weight of 0.15. The probability of target survival throughout the scenario was $p_S = 0.95$ and targets were modeled with nearly constant velocity linear motion with process noise factor $0.001\text{m}^2/\text{s}^3$. The number of particles used per target in the SMC implementations for both newborn objects and in resampling was 1000. The partition threshold values for the CLEAN-SMC-WPPHD filter were set at $N_{del} = 0.2, N_{split} = 1.5$ and $N_{rep} = 0.5$, respectively.

2.5.2 Metrics

A series of metrics were collected during the simulation in order to evaluate the performance of the WPPHD implementations against their standard PHD filter counterparts, as well as each other. For overall state estimation accuracy, there are a variety of multitarget tracking metrics that could be used such as the Hausdorff metric (El-Fallah *et al.*, 2001) or the Wasserstein Multitarget Miss Distance (Hoffman and Mahler, 2004). Recently, the Optimal Subpattern Assignment (OSPA) metric was introduced and demonstrated as a consistent metric for multitarget tracker performance evaluation (Schuhmacher *et al.*, 2008). As with the previous metrics mentioned, it considers both the distance between the individual elements as well as the difference in set cardinalities. The OSPA metric for use as a multitarget tracker evaluator is described in detail in (Ristic *et al.*, 2010b), however the basic equations are defined as follows:

$$\bar{d}_p^{(c)}(X,Y) = \left(\frac{1}{n} \left(\min_{\pi \in \Pi_k} \sum_{i=1}^{m} d^{(c)}\left(x_i, y_{\pi(i)}\right)^p + c^p \left(n-m\right) \right) \right)^{\frac{1}{p}} \tag{2.32}$$

where $d^{(c)}(x,y) = \min\{c, \|x-y\|\}$ and Π_k is the set of permutations of the set $\{1,\ldots,k\}$. Also note that in general $p \geq 1$, $c > 0$, $|X| = m$ and $|Y| = n$.

The track continuity can be measured using a variety of metrics. Both metrics used here are chosen from a collection proposed by the Multistatic Tracking Working Group (MSTWG) in (Coraluppi *et al.*, 2006). The metrics are used by the working group to benchmark multistatic tracking algorithms, however, the metrics are technically applicable as metrics for any multitarget tracking algorithm. A simple metric

that appeals to measuring continuity is the mean track fragmentation rate (mTFR). It can be computed simply as the ratio between the number of true targets N_{TT} and the total number of tracks generated N_T. Then,

$$TFR = \frac{N_{TT}}{N_T} \tag{2.33}$$

The second MSTWG metric used is the track probability of detection, also known as the overall track completeness metric. It gives the percentage of time which the true targets have an associated track estimate. It can be computed as the ratio of the total duration of tracks associated with true targets and the total duration of all true targets. The computation is as follows:

$$TP_D = \frac{\sum_{i=1}^{N_T} T_i}{\sum_{j=1}^{N_{TT}} TT_j} \tag{2.34}$$

where T_i is the total duration which track i is associated with a true target and TT_j is the total duration of true target j in the scenario.

2.5.3 Results

Track results generated by the CLEAN-SMC-WPPHD during one of the 100 Monte Carlo runs are shown in Figure 2.2. The tracks are overlayed against the true target trajectories for the entire trial. The Figure shows that the filter produces estimates that accurately estimate individual truth objects for the majority of scans. The number of false tracks remains low and the amount of track breakage is minimal with the exception of a few areas.

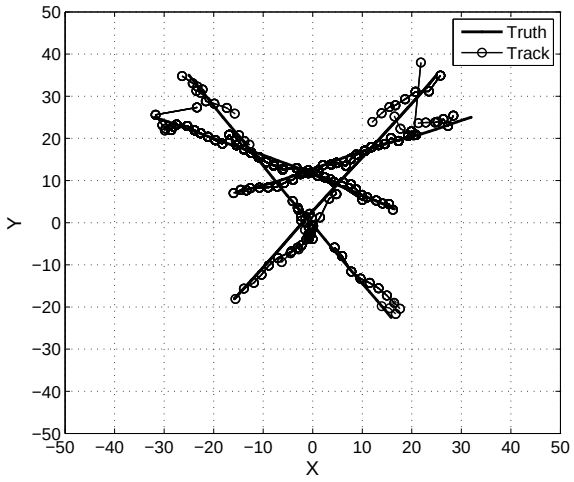

Figure 2.2: Sample CLEAN-SMC-WPPHD Track Results

The track results generated by the GM-WPPHD filter during one of the Monte Carlo runs of are shown in Figure 2.3. It shows highly accurate track results throughout the entire scenario. The tracks generated by the GM-WPPHD appear near the true target locations and consistently follow the actual target trajectories.

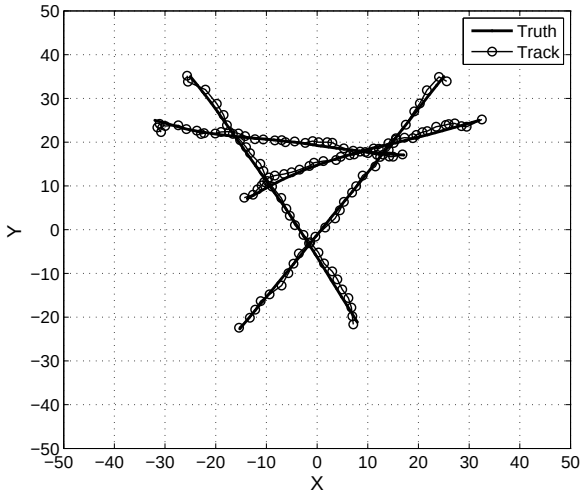

Figure 2.3: Sample LM-GM-WPPHD Track Results

The mean values over the full set of Monte Carlo runs for the OSPA distances at each scan are plotted for all filters in Figure 2.4. The metric values were generated using the OSPA parameters $p = 2$ and $c = 10$. The mean OSPA value over the entire scenario is also given for each filter in last row of Table 2.2. From the results it can be seen that the new WPPHD methods are effective in accurately measuring the cardinality and states of the multitarget scenario. The CLEAN-SMC-WPPHD is more accurate than its standard counterpart near track initialization while the tracks are closely separated. However, it is less accurate later in the scenario when the tracks diverge. On average, over the entire scenario, the standard SMC method is only slightly more accurate than the WPPHD based method. The LM-GM-WPPHD is nearly always more accurate than its standard GM-PHD counterpart throughout the scenario. The overall average OSPA value over the course of the entire scenario is significantly lower than the standard GM-PHD.

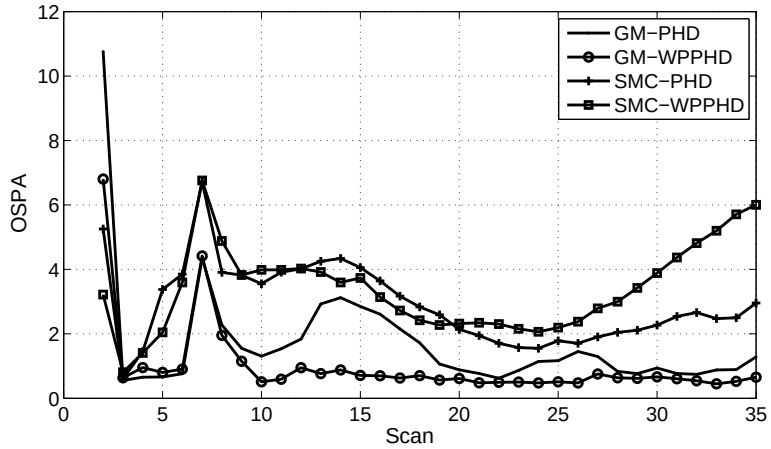

Figure 2.4: Mean OSPA Distance

The mean track fragmentation rate and track completeness metric values for all of

the filters are shown in Table 2.2. The track fragmentation rate of the CLEAN-SMC-WPPHD filter is slightly improved compared to the standard SMC-PHD implementation. However, the LM-GM-WPPHD filter track continuity is much improved over the standard GM-PHD counterpart. The track probability of detection is slightly improved in both WPPHD methods when compared to their respective standard PHD counterparts.

Table 2.2: Track Results Comparison

	Ideal Values	GM-WPPHD	GM-PHD	SMC-WPPHD	SMC-PHD
TP_D	1 (MAX)	0.844	0.799	0.637	0.635
TFR	1 (MIN)	3.79	6.00	5.25	7.96
OSPA	0 (MIN)	0.989	1.71	3.39	2.92

The computational load of each algorithm was also monitored throughout each simulation. The mean processing time for a single scan is given in seconds for each of the filters in Table 2.3. The mean run time of the SMC based methods are much higher than the GM based filters, which is typical. The more important observation is that both of the standard PHD filters operate more efficiently than their WPPHD based counterpart. However, this increase is not very significant and may be an acceptable cost in order to gain the improvements provided by the WPPHD in other metrics.

Table 2.3: Mean Filter Iteration Times

Filter	Mean (sec)
SMC-PHD	2.77
SMC-WPPHD	3.58
GM-PHD	0.104
GM-WPPHD	0.204

Chapter 3

Multiple Model Multi-Bernoulli Filter

3.1 Introduction

This chapter will derive and demonstrate an RFS based solution for tracking manoeuvring targets using the CBMeMBer filter. This is accomplished by using the methods used in classical multiple model filtering methods mainly by extending the CBMeMBer filter framework derived in (Vo *et al.*, 2009) to allow for multiple Jump Markov Models similar to the approach used in (Pasha *et al.*, 2006). The target state is extended to include a finite-valued random state variable representing the motion state model that the target is currently traveling under. This forms a multiple model Bernoulli singleton set and, by extension, a full multiple model multi-Bernoulli set that approximates a full multiple model, multitarget probability distribution function (pdf). This new MM parameter is incorporated into the RFS framework to form general multiple model prediction and update recursions of the generic CBMeMBer

equations. This is similar to the incorporation of the MM state in other RFS methods such as the PHD filter (Punithakumar *et al.*, 2008; Pasha *et al.*, 2009).

Once a general MM-CBMeMBer framework is established, derivations of the GM and SMC CBMeMBer implementations are presented. First, an SMC based approximation of the MM-CBMeMBer filter is derived by extending each weighted particle state to include the motion model state parameter. Using this extension a full set of SMC based prediction and update equations are derived with details. Next, an exact derivation of the MM-CBMeMBer filter prediction and recursion operations is given using a GM approximation to the multiple model multi-Bernoulli set. This is done by extending each Gaussian component with a model state parameter.

Simulations are used to demonstrate that both of these implementations produce more accurate estimates than their single model CBMeMBer counterparts in terms of both cardinality and overall multitarget tracking estimate. The simulations also show that these improvements are accomplished with only a marginal increase in computational time for the MM extended versions. Finally, and perhaps more importantly, the simulations are used to show that each of MM-CBMeMBer implementations can be successfully used to accurately estimate the motion model which a target is currently traveling under. This is a capability that is not available in single model CBMeMBer implementations.

In this chapter, the multiple model implementation of the CBMeMBer filter is derived in Section 3.2 including details for both the SMC and GM approximations.

These new filters are then validated against their basic counterparts, as well as each other, using a variety of multitarget tracking metrics in Section 3.3.

3.2 MM-CBMeMBer Filtering

3.2.1 Multiple Model Basics

A variety of approaches have been proposed to handle modeling manoeuvring targets (Bar-Shalom *et al.*, 2002; Blackman and Popoli, 1999). However, one approach that has been proven effective is the linear Jump Markov (JM) model (Bar-Shalom *et al.*, 2002; Pasha *et al.*, 2006; Punithakumar *et al.*, 2008). Under a JM modeled system, target motion models switch under a Markov Chain (based solely on its prior state). Consider the discrete, finite set of motion models $\{1, \ldots, N_\pi\}$. In a linear JM system the probability of transitioning between each of these models is constant. This forms a matrix $\Pi_k = \left[\alpha_k^{(m,n)}\right]_{1 \leq m,n \leq N_\pi}$ known as the regime transition matrix, where each element of the matrix represents the probability of transitioning between two models. That is,

$$\alpha_k^{(m,n)} = f_k(\pi_k = n | \pi_{k-1} = m) \tag{3.1}$$

To implement multiple model tracking using a JM approach, the random state variable is then extended to include the discrete state variable corresponding to the motion model. That is, $\tilde{x} = (x, \pi)$. The pdf of the extended random state is connected to the basic state pdf by parameterizing over the discrete model variable. Thus,

$$p(x) = \int_\Pi p(x, \pi)\, d\pi = \sum_{l=1}^{N_\pi} p(\pi = l)\, p\Big(x \Big| \pi = l\Big) \tag{3.2}$$

The JM method of extending the random state to include the target motion model alters the standard target stochastic model as well. The basic state transition and measurement likelihood models become dependent on the motion model π value. Thus,

$$f\Big(x_k, \pi_k \Big| x_{k-1}, \pi_{k-1}\Big) = p\Big(\pi_k \Big| \pi_{k-1}\Big)\, f_{\pi_k}(x_k|x_{k-1}, \pi_{k-1}) \tag{3.3}$$

$$g\Big(z \Big| x_k, \pi_k\Big) = g_{\pi_k}(z|x_k) \tag{3.4}$$

Note that while the measurement likelihood function can, in general, be dependent on the jump Markov variable, it is typically independent of the targets motion model state. Thus, for simplicity, the motion independent version $g_k(z|x_k)$ is used throughout this chapter. State birth intensities are also augmented to include initial state model likelihoods as

$$p_{\mathcal{B}}(x, \pi) = p_{\pi}(x)\, p_{\mathcal{B}}(x) \tag{3.5}$$

The JMS technique for performing multiple model target tracking has previously been applied to Random Finite Set based filters (Pasha *et al.*, 2006; Punithakumar *et al.*, 2008). These techniques were specifically applied for the use with the Probability Hypothesis Density (PHD) filter.

3.2.2 MM-CBMeMBer Filter

Using techniques similar to those used in (Pasha *et al.*, 2006; Punithakumar *et al.*, 2008), the CBMeMBer filter can be extended to implement multiple model tracking. This is accomplished by extending the traditional Bernoulli RFS state variable to include the discrete random variable representing the motion model the singleton is

operating under. Then,

$$(X,\Pi) = \left\{(x,\pi)^{(1)},\ldots,(x,\pi)^{(M)}\right\} \tag{3.6}$$

$$= \bigcup_{j=1}^{M}\left\{(x,\pi)^{(j)}\right\} = \bigcup_{i=1}^{M}(X,\Pi)^{(j)} \tag{3.7}$$

Here each singleton $(X,\Pi)^{(j)}$ is represented by the Bernoulli set $\left(r^{(j)}, p^{(j)}(x,\pi)\right)$. We refer to this as a multiple model BRFS (MM-BRFS) and the collection as an MM-MBRFS.

Thus, we have the following MM-MBRFS representation of the multitarget pdf at time k:

$$f_k(X,\Pi) \sim \left\{\left(r_k^{(j)}, p_k^{(j)}(x,\pi)\right)\right\}_{j=1}^{M_k} \tag{3.8}$$

Similar to the standard CBMeMBer filter equations, an MM-MBRFS can be predicted and updated over several time steps to form a recursive filter. This filter can be appropriately named the Multiple Model CBMeMBer (MM-CBMeMBer) filter.

MM-CBMeMBer Filter Prediction

The prediction operation of the MM-CBMeMBer filter is very similar to the standard CBMeMBer prediction operations. A predicted MM-MBRFS given in (3.8) at time $k-1$ can be predicted forward to approximate the multitarget pdf $f_{k|k-1}(X,\Pi)$ as

the union of two MM-MBRFS sets as follows:

$$f_{k|k-1}(X,\Pi) \sim \left\{ \left(r^{(j)}_{k|k-1}, p^{(j)}_{k|k-1}(x,\pi) \right) \right\}_{j=1}^{M_{k-1}} \cup \cdots \\ \cdots \left\{ \left(r^{(j)}_{\mathcal{B}_k}, p^{(j)}_{\mathcal{B}_k}(x,\pi) \right) \right\}_{j=1}^{M_{\mathcal{B}_k}} \tag{3.9}$$

The predicted (legacy) MM-BRFS can be computed using equations similar to the basic CBMeMBer filter prediction equations described in (1.59) and (1.60) but extended to include the MM variable. The single model stochastic equations are replaced with the model specific transition equations outlined in Section 3.2.1. The new MM-CBMeMBer prediction equations can be defined as follows:

$$r^{(j)}_{k|k-1} = r^{(j)}_{k-1} \left\langle p^{(j)}_{k-1}, p_S \right\rangle \tag{3.10}$$

$$p^{(j)}_{k|k-1}(x,\pi) = \frac{\left\langle f_{k|k-1}(x,\pi|\cdot)\, , p^{(j)}_{k-1} p_S \right\rangle}{\left\langle p^{(j)}_{k-1}, p_S \right\rangle} \tag{3.11}$$

Here it is important to recall that the notation $\langle \cdot, \cdot \rangle$ refers to the integral over the full state variable, which here includes the summation over the model state variable as follows:

$$\langle v, h \rangle = \int_{(X,\Pi)} v(x,\pi)\, h(x,\pi)\ \, dx d\pi \tag{3.12}$$

$$= \int_X \int_\Pi v(x,\pi)\, h(x,\pi)\, d\pi\ dx \tag{3.13}$$

$$= \sum_\pi \int_X v(x|\pi)\, h(x|\pi)\ \, dx \tag{3.14}$$

The birth MM-MBRFS $\left\{ \left(r^{(j)}_{\mathcal{B}_k}, p^{(j)}_{\mathcal{B}_k}(x,\pi) \right) \right\}_{j=1}^{M_{\mathcal{B}_k}}$ is similar to that described in

(1.58), however, it also considers the initial model state likelihoods.

MM-CBMeMBer Filter Update

The update operation of the MM-CBMeMBer filter is also quite similar to the standard CBMeMBer update operation. An MM-MBRFS given in (3.8) at time $k|k-1$ has the posterior multitarget pdf $f_k(X, \Pi)$ defined as the union of two MM-MBRFS sets as follows:

$$f_k(X, \Pi) \sim \left\{ \left(r_k^{(j)}, p_k^{(j)}(x, \pi) \right) \right\}_{j=1}^{M_{k|k-1}} \cup \left\{ (r_k(z), p_k(x, \pi|z)) \right\}_{z \in Z_k} \tag{3.15}$$

The equations used to describe the update MM-BRFS sets described in (3.15) are similar to the standard CBMeMBer update equations described in (1.62)–(1.65) but are extended to take into account the multiple model parameters. First, the missed detection update equations are defined similar to those outlined in (1.62) and (1.63):

$$r_k^{(j)} = \frac{1 - \left\langle p_{k|k-1}^{(j)}, p_D \right\rangle}{1 - r_{k|k-1}^{(j)} \left\langle p_{k|k-1}^{(j)}, p_D \right\rangle} r_{k|k-1}^{(j)} \tag{3.16}$$

$$p_k^{(j)}(x, \pi) = \frac{1 - p_D(x, \pi)}{1 - \left\langle p_{k|k-1}^{(j)}, p_D \right\rangle} p_{k|k-1}^{(j)}(x, \pi) \tag{3.17}$$

Finally, the equations for creating the new measurement based MM-BRFS in (3.15) are also similar to those described in (1.66) and (1.67). However, here the single target measurement likelihood function $g_k(z|x)$ can be replaced with a model

specific measurement likelihood function $g_k(z|x,\pi)$ (if it differs for each model):

$$\pi_k(z) = \frac{\sum_{j=1}^{M_{k|k-1}} \frac{r_{k|k-1}^{(j)} \left\langle p_{k|k-1}^{(j)}, \Psi_{k,z} \right\rangle}{1 - r_{k|k-1}^{(j)} \left\langle p_{k|k-1}^{(j)}, p_D \right\rangle}}{\kappa_k(z) + \sum_{j=1}^{M_{k|k-1}} \frac{r_{k|k-1}^{(j)} \left\langle p_{k|k-1}^{(j)}, \Psi_{k,z} \right\rangle}{1 - r_{k|k-1}^{(j)} \left\langle p_{k|k-1}^{(j)}, p_D \right\rangle}} \tag{3.18}$$

$$p_k(x,\pi|z) = \frac{\sum_{j=1}^{M_{k|k-1}} \frac{r_{k|k-1}^{(j)} p_{k|k-1}^{(j)}(x,\pi) \Psi_{k,z}(x,\pi)}{1 - r_{k|k-1}^{(j)} \left\langle p_{k|k-1}^{(j)}, p_D \right\rangle}}{\sum_{j=1}^{M_{k|k-1}} \frac{r_{k|k-1}^{(j)} \left\langle p_{k|k-1}^{(j)}, \Psi_{k,z} \right\rangle}{1 - r_{k|k-1}^{(j)} \left\langle p_{k|k-1}^{(j)}, p_D \right\rangle}} \tag{3.19}$$

$$\text{where } \Psi_{k,z}(x,\pi) = g_k(z|x,\pi)\, p_D(x,\pi)$$

The above definitions of the MM-CBMeMBer equations are based on the standard CBMeMBer filter's update equations. The key enhancement to the standard CBMeMBer filter is the addition of the motion state variable. By distributing over and applying specific motion transition and measurement update equations in the prediction and update steps respectively the uncertainly of the targets motion is captured and estimated. The base derivation, as well as any implementation specific derivations that may proceed, follow similarly for the CBMeMBer filter's equation. Using these new MM update equations forms the MM-CBMeMBer filter and its respective implementations.

3.2.3 SMC-MM-CBMeMBer Filter

The MM-CBMeMBer filter described above can be implemented using a particle based approximation with an extension to the SMC-CBMeMBer approximation of the standard CBMeMBer filter. The extension involves the addition of a model identification

variable along with each particles state-weight pairing. This model parameter is a discrete representation of the current particle states model. The extended particle set is described as follows:

$$(X_k, \Pi_k) = \left\{ \left(r_k^{(j)}, p_k^{(j)}(x, \pi) \right) \right\}_{j=1}^{M_k} \tag{3.20}$$

$$\sim \left\{ \left(r_k^{(j)}, \left\{ \left(x_k^{(i,j)}, \pi_k^{(i,j)}, \omega_k^{(i,j)} \right) \right\}_{i=1}^{L_k^{(j)}} \right) \right\}_{j=1}^{M_k} \tag{3.21}$$

This forms an approximation to the MM extended BRFS pdf in (3.8) that is similar to the SMC approximation given in (1.76) as follows:

$$p_k^{(j)}(x, \pi) = \sum_{i=1}^{L_k^{(j)}} \omega_k^{(i,j)} \delta_{x_k^{(i,j)}, \pi_k^{(i,j)}}(x, \pi) \tag{3.22}$$

SMC-MM-CBMeMBer Filter Predict

The regime extended particle set described in (3.21) can be predicted forward using the MM-CBMeMBer prediction equations stated in section (3.2.2) under an SMC implementation. These equations are similar to the standard SMC-CBMeMBer equations described in (Vo *et al.*, 2009), however, taking into account the additional regime conditioned particle state using equations similar to those first outlined in (Ristic *et al.*, 2004). The derivation of the key elements of the standard MM-MeMBer predict equations can be found in Appendix B.1.

First, the predicted MMBRFS can be described as follows:

$$r_{k|k-1}^{(j)} = r_{k-1}^{(j)} \sum_{i=1}^{L_k^{(j)}} p_S\left(x_{k-1}^{(i,j)}, \pi_{k-1}^{(i,j)}\right) \omega_{k-1}^{(i,j)} \tag{3.23}$$

$$p_{k|k-1}^{(j)}(x,\pi) = \sum_{i=1}^{L_k^{(j)}} \omega_{k|k-1}^{(i,j)} \delta_{x_k^{(i,j)},\pi_k^{(i,j)}}(x,\pi) \tag{3.24}$$

where

$$\pi_k^{(i,j)} \sim q_\Pi\left(\cdot \middle| \pi_{k-1}^{(i,j)}\right) \tag{3.25}$$

$$x_{k|k-1}^{(i,j)} \sim q_X\left(\cdot \middle| x_{k-1}^{(i,j)}, \pi_k^{(i,j)}\right)$$

$$\tilde{\omega}_{k|k-1}^{(i,j)} = \frac{p\left(\pi_k^{(i,j)} \middle| \pi_{k-1}^{(i,j)}\right) f\left(x_k^{(i,j)} \middle| x_{k-1}^{(i,j)}, \pi_k^{(i,j)}\right) p_S\left(x_{k-1}^{(i,j)}, \pi_{k-1}^{(i,j)}\right)}{q_\Pi\left(\pi_k^{(i,j)} \middle| \pi_{k-1}^{(i,j)}\right) q_X\left(x_k^{(i,j)} \middle| x_{k-1}^{(i,j)}, \pi_{k-1}^{(i,j)}, \pi_k^{(i,j)}\right)} \omega_{k-1}^{(i,j)} \tag{3.26}$$

$$\omega_{k|k-1}^{(i,j)} = \tilde{\omega}_{k|k-1}^{(i,j)} \Big/ \sum_{i=1}^{L_{k-1}^{(j)}} \tilde{\omega}_{k|k-1}^{(i,j)} \tag{3.27}$$

$$\text{for } i = 1, \ldots, L_{k-1}^{(j)} \text{ and } j = 1, \ldots, M_{k-1}$$

where q_X, q_Π are the state and motion model importance sampling distributions respectively. The key difference in these equations to those in the standard CBMeMBer filter found in (Vo *et al.*, 2009) is the sampling of the motion model in (3.25) and its application in the computation of the weight in (3.26). Next, a multiple model SMC

approximation to the MM-MBRFS birth set is generated as follows:

$$r_{\mathcal{B}_k}^{(j)} = \int_{X,\Pi} b_k(x,\pi)\ dx\, d\pi \tag{3.28}$$

$$p_{\mathcal{B}_k}^{(j)}(x,\pi) = \sum_{i=1}^{L_{\mathcal{B}_k}^{(j)}} \omega_{\mathcal{B}_k}^{(i,j)} \delta_{x_{\mathcal{B}_k}^{(i,j)},\pi_{\mathcal{B}_k}^{(i,j)}}(x,\pi) \tag{3.29}$$

$$\text{where } \pi_{\mathcal{B}_k}^{(i,j)} \sim b_k(\cdot) \tag{3.30}$$

$$x_{\mathcal{B}_k}^{(i,j)} \sim b_k\left(\cdot \middle| Z_k\right) \tag{3.31}$$

$$\tilde{\omega}_{\mathcal{B}_k}^{(i,j)} = \frac{p_k^{\mathcal{B}}\left(x_{\mathcal{B}_k}^{(i,j)}, \pi_{\mathcal{B}_k}^{(i,j)}\right)}{b\left(x_{\mathcal{B}_k}^{(i,j)}, \pi_{\mathcal{B}_k}^{(i,j)} \middle| Z_k\right)} \tag{3.32}$$

$$\omega_{\mathcal{B}_k}^{(i,j)} = \tilde{\omega}_{\mathcal{B}_k}^{(i,j)} \Big/ \sum_{i=1}^{L_{\mathcal{B}_k}^{(j)}} \tilde{\omega}_{\mathcal{B}_k}^{(i,j)} \tag{3.33}$$

$$\text{for } i = 1,\ldots,L_{\mathcal{B}_k}^{(j)} \text{ and } j = 1,\ldots,M_{\mathcal{B}_k}$$

where $b(\cdot)$ is the birth distribution and is used for sampling both the state space (3.31) and the motion model (3.30).

SMC-MM-CBMeMBer Filter Update

To update an SMC-MM-CBMeMBer filter approximation, the equations given in Section 3.2.2 are implemented in an SMC context, similar to the process for the standard SMC-CBMeMBer implementation update equations given in (Vo *et al.*, 2009). The single target measurement likelihood function $g_k(z|x)$ can be replaced with a model-specific measurement likelihood function if it differs for each model. The operations for the update step are similar to those for a basic regime conditioned particle filter

as in (Ristic *et al.*, 2004), however, under an RFS context, similar to that found in (Punithakumar *et al.*, 2008). Note, that there is no extra update step for the regime (model) state of the MBRFS. More details regarding the derivation of some key elements of the standard MM-MeMBer update equations in the SMC context can be found in Appendix B.1.

Consider an SMC-MM-MBRFS approximation as given in (3.21). The updated MM-MBRFS in (3.15) can be described as follows. First, the legacy (missed detection) MM-MBRFS are determined as follows:

$$r_k^{(j)} = r_{k|k-1}^{(j)} \frac{1-\rho_k^{(j)}}{1-r_{k|k-1}^{(j)}\rho_k^{(j)}} \tag{3.34}$$

$$p_k^{(j)}(x,\pi) = \sum_{i=1}^{L_{k|k-1}^{(j)}} \omega_k^{(i,j)} \delta_{x_k^{(i,j)},\pi_k^{(i,j)}}(x,\pi) \tag{3.35}$$

$$\text{where } \rho_k^{(j)} = \sum_{i=1}^{L_{k|k-1}^{(j)}} \omega_{k|k-1}^{(i,j)} p_D\left(x_{k|k-1}^{(i,j)}, \pi_{k|k-1}^{(i,j)}\right) \tag{3.36}$$

$$\tilde{\omega}_k^{(i,j)} = \omega_{k|k-1}^{(i,j)} \left(1 - p_D\left(x_{k|k-1}^{(i,j)}, \pi_{k|k-1}^{(i,j)}\right)\right) \tag{3.37}$$

$$\omega_k^{(i,j)} = \tilde{\omega}_k^{(i,j)} \Big/ \sum_{i=1}^{L_{k|k-1}^{(j)}} \tilde{\omega}_k^{(i,j)} \tag{3.38}$$

$$\text{for } i = 1, \ldots, L_{k|k-1}^{(j)} \text{ and } j = 1, \ldots, M_{k|k-1}$$

Here the key difference versus the standard GM-CBMeMBer described in (Vo *et al.*, 2009) is the application of both the state space and motion model state to a particles

probability of detection. Next, each measurement $z \in Z_k$ generates and SMC-MM-MBRFS as follows:

$$r_k(z) = \frac{\sum_{j=1}^{M_{k|k-1}} \frac{r_{k|k-1}^{(j)}\left(1-r_{k|k-1}^{(j)}\right)\rho_k^{(j)}(z)}{\left(1-r_{k|k-1}^{(j)}\rho_k^{(j)}(z)\right)^2}}{\kappa_k(z) + \sum_{j=1}^{M_{k|k-1}} \frac{r_{k|k-1}^{(j)}\rho_k^{(j)}(z)}{1-r_{k|k-1}^{(j)}\rho_k^{(j)}(z)}} \tag{3.39}$$

$$p_k(x,\pi|z) = \sum_{j=1}^{M_{k|k-1}} \sum_{i=1}^{L_{k|k-1}^{(j)}} \omega_k^{(i,j)} \delta_{x_{k|k-1}^{(i,j)},\pi_{k|k-1}^{(i,j)}}(x,\pi) \tag{3.40}$$

$$\text{where } \rho_k^{(j)}(z) = \sum_{i=1}^{L_{k|k-1}^{(j)}} \omega_{k|k-1}^{(i,j)} \Psi_{k,z}\left(x_{k|k-1}^{(i,j)}, \pi_{k|k-1}^{(i,j)}\right) \tag{3.41}$$

$$\tilde{\omega}_k^{(i,j)}(z) = \omega_{k|k-1}^{(i,j)} \frac{r_{k|k-1}^{(j)}}{1-r_{k|k-1}^{(j)}} \Psi_{k,z}\left(x_{k|k-1}^{(i,j)}, \pi_{k|k-1}^{(i,j)}\right) \tag{3.42}$$

$$\omega_k^{(i,j)}(z) = \tilde{\omega}_k^{(i,j)}(z) \Big/ \sum_{j=1}^{M_{k|k-1}} \sum_{i=1}^{L_{k|k-1}^{(j)}} \tilde{\omega}_k^{(i,j)}(z) \tag{3.43}$$

$$\text{for } i = 1, \ldots, L_{k|k-1}^{(j)} \text{ and } z \in Z_k$$

Here the MM-CBMeMBer differs from the standard implementation in the usage of a model based target likelihood function $\Psi_z(x,\pi)$ in the updated weight computations in (3.41) and (3.42).

3.2.4 GM-MM-CBMeMBer Filter

The GM-CBMeMBer filter may be extended to perform multiple model filtering. Essentially, each GM component is extended to include a model parameter that estimates the model which that component is currently operating under. A Jump Markov

(JM) model (Bar-Shalom *et al.*, 2002) may then be applied to allow for model switching parameters to be modeled at each prediction step. This is similar to the manner in which the GM-PHD filter is extended to allow for linear JM model (LJM) in (Pasha *et al.*, 2006). The Gaussian Mixture Multiple Model MBRFS (GM-MM-MBRFS) is described as follows:

$$(X_k, \Pi_k) = \left\{ \left(r_k^{(j)}, p_k^{(j)}(x, \pi) \right) \right\}_{j=1}^{M_k} \tag{3.44}$$

$$\sim \left\{ r_k^{(j)}, \left\{ \left(m_k^{(i,j)}, P_k^{(i,j)}, \pi_k^{(i,j)}, \omega_k^{(i,j)} \right) \right\}_{i=1}^{L_k^{(j)}} \right\}_{j=1}^{M_k} \tag{3.45}$$

$$p_k^{(j)}(x, \pi) = \sum_{i=1}^{L_k^{(j)}} \omega_k^{(i)}(\pi) \mathcal{N}\left(x; m_k^{(i,j)}(\pi), P_k^{(i,j)}(\pi) \right) \tag{3.46}$$

GM-MM-CBMeMBer Filter Prediction

The GM-MM-CBMeMBer prediction operations are performed by following the MM specific update operations described in (3.10) and (3.11). The GM-MM-BRFS as described in (3.45) at time $k-1$ can be predicted forward to time k. First, the predicted GM-MM-BRFS described in (3.9) is computed as follows:

$$r_{k|k-1}^{(j)} = r_{k-1}^{(j)} \sum_{\pi_{k-1}=1}^{N_\pi} \sum_{i=1}^{M_k^{(j)}} p_S(\pi) \omega_{k|k-1}^{(i)}(\pi_{k-1}) \tag{3.47}$$

$$p_{k|k-1}^{(j)}(x, \pi_k) = \sum_{\pi_{k-1}=1}^{N_\pi} \sum_{i=1}^{M_k^{(j)}} \omega_{k|k-1}^{(i)}(\pi_k | \pi_{k-1}) \cdots$$

$$\cdots \mathcal{N}\left(x; m_{k|k-1}^{(i,j)}(\pi_k | \pi_{k-1}), P_{k|k-1}^{(i,j)}(\pi_k | \pi_{k-1}) \right) \tag{3.48}$$

$$\text{where } \omega_{k|k-1}^{(i)}(\pi_k|\pi_{k-1}) = p\Big(\pi_k\Big|\pi_{k-1}\Big)\, p_S(\pi_{k-1})\, \omega_{k|k-1}^{(i)} \tag{3.49}$$

$$m_{k|k-1}^{(i)}(\pi_k|\pi_{k-1}) = F_{k-1}(\pi_{k-1}) m_{k-1}^{(j)}(\pi_{k-1}) \tag{3.50}$$

$$P_{k|k-1}^{(i)}(\pi_k|\pi_{k-1}) = F_{k-1}(\pi_{k-1}) P_{k-1}^{(j)}(\pi_{k-1}) F_{k-1}(\pi_{k-1})^T \cdots$$

$$\cdots + Q_{k-1}(\pi) \tag{3.51}$$

$$\text{for } i = 1, \ldots, L_{k-1}^{(j)} \text{ and } j = 1, \ldots, M_{k-1}^{(j)}$$

where $F_{k-1}(\pi)$ is the linear state transition matrix under motion model π and similarly $Q_{k-1}(\pi)$ is the process noise matrix for the same motion model. Also, $p_S(\pi)$ denotes the regime specific probability of survival. However, in most cases, this parameter is regime independent as well (i.e. $p_S(\pi) = p_S$). The key extension here versus the standard CBMeMBer described in (Vo *et al.*, 2009) is the inclusion of the model parameter in the each target motion parameters, especially the usage of the motion transition probability $p\Big(\pi_k\Big|\pi_{k-1}\Big)$ added in (3.49). Further, details of the derivation of key elements of the base MM-MeMBer prediction equations can be found in Appendix B.2.

The set of new birth BRFS $\left\{\left(r_{\mathcal{B}_k}^{(j)}, p_{\mathcal{B}_k}^{(j)}(x,\pi)\right)\right\}_{j=1}^{M_{\mathcal{B}_k}}$ is formed using multiple model GM approximations to the birth set pdfs as follows:

$$p_{\mathcal{B}_k}^{(j)}(x,\pi) = \sum_{i=1}^{J_{\mathcal{B}_k}^{(j)}} \omega_{\mathcal{B}_k}^{(i,j)}(\pi) \mathcal{N}\left(x; m_{\mathcal{B}_k}^{(i,j)}(\pi), P_{\mathcal{B}_k}^{(i,j)}(\pi)\right) \tag{3.52}$$

where $m_{\mathcal{B}_k}^{(i,j)}(\pi), P_{\mathcal{B}_k}^{(i,j)}(\pi)$ and $\omega_{\mathcal{B}_k}^{(i,j)}$ represent the GM approximation to the pdfs of the multiple model birth RFS.

GM-MM-CBMeMBer Filter Update

Again, the GM-MM-CBMeMBer filter version of the equations given in section 3.2.2 are derived similar to the basic GM-CBMeMBer update equations outlined in (Vo *et al.*, 2009). First, the legacy (missed detection) GM-MM-MBRFS are updated with a missed detection in the same manner as the standard GM-CBMeMBer:

$$r_k^{(j)} = \frac{1 - p_D(\pi)}{1 - r_{k|k-1}^{(j)} p_D(\pi)} r_{k|k-1}^{(j)} \tag{3.53}$$

$$p_k^{(j)}(x, \pi) = p_{k|k-1}^{(j)}(x, \pi) \tag{3.54}$$

$$\text{for } j = 1, \ldots, M_{k|k-1}^{(j)}$$

Here the key improvement on the standard CBMeMBer filter is the inclusion of the motion model parameter in the computation of the targets probability of detection.

Finally, each measurement $z \in Z_k$ generates a new GM-MM-MBRFS using the following equations:

$$r_k(z) = \frac{\sum_{j=1}^{M_{k|k-1}} \frac{r_{k|k-1}^{(j)}\left(1-r_{k|k-1}^{(j)}\right)\upsilon^{(j)}(z)}{\left(1-r_{k|k-1}^{(j)}p_D(\pi)\right)^2}}{\kappa_k(z) + \sum_{j=1}^{M_{k|k-1}} \frac{r_{k|k-1}^{(j)}\upsilon^{j}(z)}{1-r_{k|k-1}^{(j)}p_D(\pi)}} \tag{3.55}$$

$$p_k^{(j)}(x,\pi;z) = \frac{\sum_{j=1}^{M_{k|k-1}}\sum_{i=1}^{J_{k|k-1}^{(j)}} \omega_k^{(i,j)}(z)\,\mathcal{N}\left(x; m_k^{(i,j)}, P_k^{(i,j)}\right)}{\sum_{j=1}^{M_{k|k-1}}\sum_{i=1}^{J_{k|k-1}^{(j)}} \omega_k^{(i,j)}(z)} \tag{3.56}$$

$$\text{where } \upsilon^{(i)}(z) = p_D(\pi)\sum_{j=1}^{J_{k|k-1}^{(j)}} \omega_{k|k-1}^{(i,j)}\mathcal{N}\left(z; Hm_{k|k-1}^{(i,j)}, S_k^{(i,j)}\right)$$

$$\omega_k^{(i,j)}(z) = p_D(\pi)\omega_{k|k-1}^{(i,j)}\mathcal{N}\left(z; Hm_{k|k-1}^{(i,j)}, S_k^{(i,j)}\right)\frac{r_{k|k-1}^{(i)}}{1-r_{k|k-1}^{(i)}} \tag{3.57}$$

$$m_k^{(i,j)} = m_{k|k-1}^{(i,j)} + K_k^{(i,j)}\left(z - Hm_{k|k-1}^{(i,j)}\right) \tag{3.58}$$

$$P_k^{(i,j)} = \left[I - K_k^{(i,j)}H\right]P_{k|k-1}^{(i,j)} \tag{3.59}$$

$$K_k^{(i,j)} = P_{k|k-1}^{(i,j)}H^T\left[S_k^{(i,j)}\right]^{-1} \tag{3.60}$$

$$S_k^{(i,j)} = HP_{k|k-1}^{(i,j)}H^T + R \tag{3.61}$$

$$\text{for } i = 1,\ldots,L_{k|k-1}^{(j)} \text{ and } j = 1,\ldots,M_{k|k-1}^{(j)}$$

The main difference between GM-MM-CBMeMBer update equations and those of the standard GM-CBMeMBer filter is the use of model specific probability of detection parameter $p_D(\pi)$. However, in most tracking cases the detection probability is independent of the model (i.e. $p_D(\pi) = p_D$), making the equations the same throughout. Also, the measurement likelihood model can be extended to include the model specific a parameters H_π, R_π. More detailed derivations of the MM-MeMBer update

equation in the GM context can be found in Appendix B.2.

3.3 Simulation

3.3.1 Scenario

In order to test the viability and performance of the algorithms described above, each MM-CBMeMBer implementation is demonstrated using a simulated scenario. Consider a two dimensional scenario space of size 2000m in each dimension. The scenario consists of four manoeuvring targets as shown in Figure 3.1. The targets undergo a variety of motion patterns throughout the scenario so as to thoroughly test the ability of the techniques to track and detect all types of manoeuvres. Linear positional measurements are generated using a single sensor that monitors the entire scenario space at a time interval of $T = 1$s. The target measurements have additive Gaussian noise with standard deviation $\sigma_x^2 = 10\text{m}^2$ independent in each direction. Targets were detected based on a uniform and model independent probability of $p_D = 0.95$. The sensor also generated a number of false measurements with density $\lambda_c = 2.5\times10^{-6}\text{m}^{-2}$ and a uniform spatial distribution over the entire scenario space. The entire scenario consists of 60 time intervals. Included in the scenario were several elements considered to be difficult multi-target tracking aspects such as multiple targets appearing within sequential time intervals as well as crossing (closely spaced) targets.

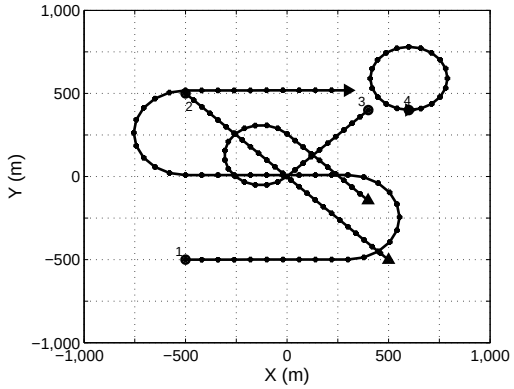

Figure 3.1: 2-D scenario containing four targets

3.3.2 Filter Configuration

Each MM-CBMeMBer filter was configured to operate using as many common parameters as possible, with essentially only the implementation specific parameters varying. The filters were also configured to use the correct scenario parameter values wherever possible. These parameters include p_D, and p_{FA} as defined above as well as probability of target survival ($p_S = 0.95$) and clutter intensity ($\kappa(z) = V \cdot \lambda_c = 10$). Each filter utilized a birth RFS approximation based on five Gaussian distributions means located at $(0,0), (-500,-500), (-500,500), (500,-500), (500,500)$ in the scenario space (each with $(0,0)$ as the mean velocity components). Each Gaussian component had a diagonal covariance with $\sigma_x^2 = 200^2\text{m}^2$ and $\sigma_{\dot{x}}^2 = 60^2\text{m}^2/\text{s}^2$ as the position and velocity covariance components, respectively. The CBMeMBer birth intensity totaled $r_{BIRTH} = 0.08$, which was equally distributed amongst five birth BRFS, each constructed using one of the five Gaussian birth components described above. As well, each MM-CBMeMBer filter shared common maintenance and reporting parameters such as a BRFS deletion threshold of $N_{DEL} = 0.1$ and maximum number of BRFS in the MBRFS estimate of $N_{BRFS_{MAX}} = 100$.

For the individual filter parameters, the particle based CBMeMBer filters had the specific parameters of number of particle per BRFS estimate set to $N_{PPT} = 100$ along with a minimum of 30 particles for each BRFS estimate. The resampling threshold for each BRFS was set to $N_{RESAMPLE} = 0.75$. For the GM based CBMeMBer filters, the GM deletion threshold parameter was set to $N_{GM\_DELETE} = 10^{-5}$ and the GM merging threshold was set to $N_{GM\_MERGE} = 4$. The maximum number of GM per BRFS was also set to $N_{GMPT} = 100$.

For the MM specific parameters, a three model configuration using two specific types of linear static motion models were used. A single nearly constant velocity (NCV) motion model defined in (3.63) with $q_{NCV} = 0.016\text{m}^2/\text{s}$ as well as two constant, fixed turn-rate (CT) model with values $\theta = \pm\pi/10,\ q_{CT} = 0.0016\text{m}^2/\text{s}$ described in

(3.64).

$$F_{NCV} = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3.62}$$

$$Q_{NCV} = q_{NCV} \begin{bmatrix} \frac{T^3}{3} & \frac{T^2}{2} & 0 & 0 \\ \frac{T^2}{2} & T & 0 & 0 \\ 0 & 0 & \frac{T^3}{3} & \frac{T^2}{2} \\ 0 & 0 & \frac{T^2}{2} & T \end{bmatrix} \tag{3.63}$$

$$F_{CT} = \begin{bmatrix} 1 & \frac{\sin\theta T}{\theta} & 0 & -\frac{1-\cos\theta T}{\theta} \\ 0 & \cos\theta T & 0 & -\sin\theta T \\ 0 & \frac{1-\cos\theta T}{\theta} & 1 & \frac{\sin\theta T}{\theta} \\ 0 & \sin\theta T & 0 & \cos\theta T \end{bmatrix}, \tag{3.64}$$

$$Q_{CT} = q_{CT} \begin{bmatrix} \frac{T^4}{4} & \frac{T^3}{2} & 0 & 0 \\ \frac{T^3}{2} & T^2 & 0 & 0 \\ 0 & 0 & \frac{T^4}{4} & \frac{T^3}{2} \\ 0 & 0 & \frac{T^3}{2} & T^2 \end{bmatrix} \tag{3.65}$$

Note that the CT model given in (3.64) is linear when the turn rate variable θ is a fixed value. These models were given initial probabilities [0.5 0.25 0.25] and model

switching was done under transition matrix given in (3.66).

$$\Pi_k = \begin{bmatrix} 0.2 & 0.2 & 0.2 \\ 0.4 & 0.8 & 0 \\ 0.4 & 0 & 0.8 \end{bmatrix} \tag{3.66}$$

3.3.3 Metrics

The performance of each MM-CBMeMBer filter implementation (i.e. SMC and GM) was measured against their respective single motion model based implementation, using the single NCV motion model as described in (3.63). As well, the two MM implementations were compared against each other in order to determine what conditions are optimal for various MM-CBMeMBer tracking metrics. The filters were measured in several different multi-target tracking aspects including: target detection (cardinality accuracy), target localization (estimate location accuracy) and filter computational complexity. As well, each of the MM implementations was measured for its effectiveness in model (likelihood) detection.

The cardinality estimates reported by each filter were based on the most likely number of targets reported by each CBMeMBer filter at each scan. To measure target localization accuracy, the Optimal Subpattern Assignment (OSPA) metric was used (Schuhmacher *et al.*, 2008). The OSPA is a metric for measuring overall tracking performance of a multitarget tracking algorithm. It considers both the cardinality of the multi-object estimate versus the true number of objects as well as the accuracy of those estimates. The OSPA is described in detail in (Ristic *et al.*, 2010b), however the basic equations are defined in Chapter 1 equation (2.32)

Finally, the model detection accuracy of each MM-CBMeMBer filter was also measured. The likelihood of an MM-CBMeMBer estimate being in each model was determined by computing the normalized weighted sum of the BRFS components (either GMs or particles) for each of the regimes as described in (3.67). At each time step, individual track estimates were associated to truth objects using multi-dimensional assignment. Once the object is assigned to a given target, the likelihood of targets motion models can be measured using the following equation:

$$Pr[\pi = m] = \sum_{i \in \Pi_m} \omega_k^{(i)} \tag{3.67}$$

$$\text{where } \Pi_m = \left\{ i | \pi_k^{(i)} = m \right\} \tag{3.68}$$

3.3.4 Results

The described scenario was processed using each of the MM-CBMeMBer implementations, as well as their basic single motion model equivalents, for 100 Monte Carlo runs. The mean cardinality results over all Monte Carlo runs of each filter are plotted against the actual scenario cardinality in Fig. 3.2. The plot shows that all implementations of the CBMeMBer filter are effective at determining the actual number of targets, with the GM implementations slightly improved compared to the SMC based implementations. Also, each MM-CBMeMBer implementation gives a slightly more accurate cardinality estimate than its single model counterpart.

The overall filter accuracy and performance metric, the OSPA, was also computed for each filter over all Monte Carlo runs. The mean OSPA values are plotted over time in Fig. 3.3. The key observation is that for the majority of the simulation

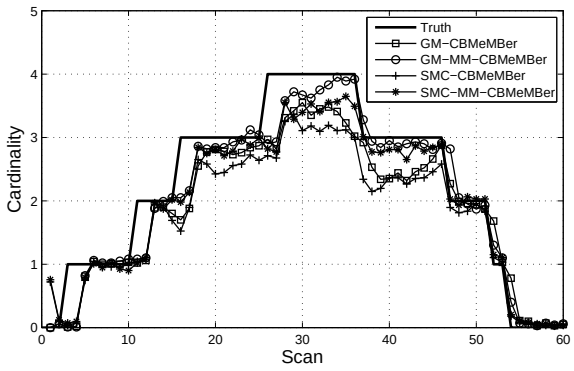

Figure 3.2: Mean filter cardinality estimates

the MM version of both the SMC and GM implementations perform at the same level or better than their single model counterpart. It is interesting to observe that at certain portions the single model implementations slightly outperform the MM implementations. This can typically occur during instances where the majority of targets are undergoing NCV motion. Here the single target filters correctly match the model while false alarms can cause the MM filters to incorrectly interpret a manoeuvre which is a common issue with many MM filters.

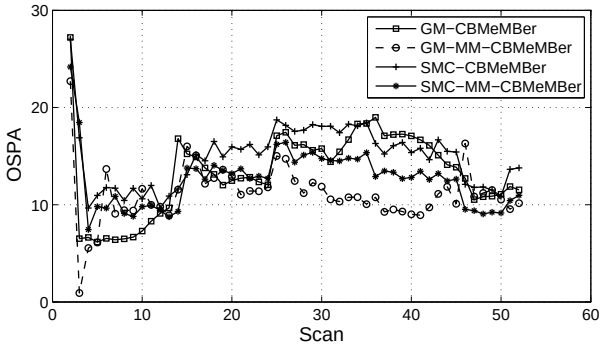

Figure 3.3: Mean OSPA values

The filter complexity was measured by recording the computational time of each time step of the filtering algorithms. The mean and standard deviation of the run time for each filter iteration (a single prediction and update step) in seconds is reported in Table 3.1. First, note that the SMC implementations take significantly more time to compute than the GM implementations, which is expected. It should also be noted that, as expected, the MM-CBMeMBer implementations require more computational time than their respective single model counterparts. However, the increase in computational time is minimal for each implementation. This is somewhat counterintuitive to the theoretical reasoning. The number of the Gaussian components in the GM-BRFS generated after each iteration of the GM-MM-CBMeMBer filter increases proportional to the total number of motion models. Also, the number of particles needed to effectively estimate the state of each SMC-BRFS increases proportional to the number of models (due to the curse of dimensionality (Daum and Huang, 2003)). The fact that this increase in computational time is only marginal can mainly be attributed to the efficient use of pruning and trimming operations in the CBMeMBer filter methods. At each iteration, the number of elements (particles or GM components) are effectively managed by resampling and pruning/merging operations respectively. By using the same filter maintenance parameters as their single model counterparts, the MM filters performs in nearly the same time at each iteration, save some extra work required for the additional maintenance effort.

Finally, the model detection accuracy of the MM-CBMeMBer implementations was measured as described above. The results are plotted for two of the four targets in Fig. 3.4 and Fig. 3.5 below. Target 1 is the one traveling in the S-shaped pattern

Table 3.1: Filter mean computational times in seconds

Filter	Mean (sec)	Std. Dev. (sec)
SMC-CBMeMBer	5.1162	0.1657
SMC-MM-CBMeMBer	5.2342	0.1547
GM-CBMeMBer	0.3191	0.0278
GM-MM-CBMeMBer	0.4101	0.0395

starting from the lower left quadrant and traveling to the upper left. Target 3 is the one that completes the α-shaped manoeuvre in the right quadrant.

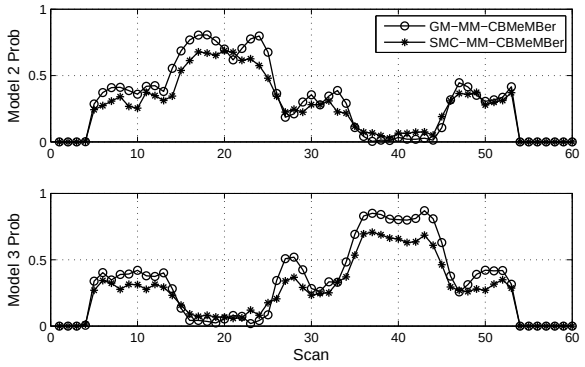

Figure 3.4: Target 1 model probabilities

The results in Fig. 3.4 and Fig. 3.5 show that the MM-CBMeMBer implementations are successfully able to detect when the targets switch motion patterns and begin to manoeuvre. The most evident case of this is shown for target 1 in Figure 3.4. The likelihood of model 2 (counter-clockwise turn) increases near scan 15 when the target is making its counter-clockwise manoeuvre. Similarly, when the likelihood of model 3 (clockwise turn) increases near scan 40 when target 1 is making a clockwise manoeuvre. As well, in Figure 3.5, the clockwise turn model of target 3 becomes much more likely during scans 25–40 when the target is performing its manoeurve.

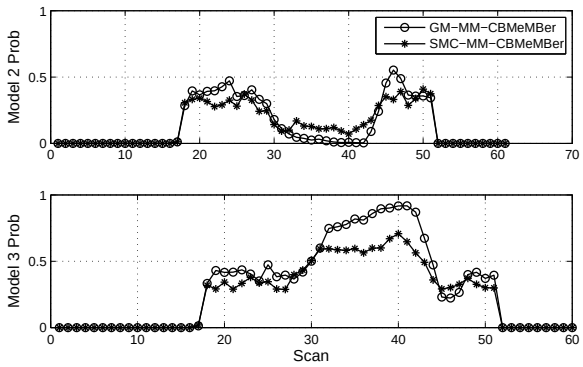

Figure 3.5: Target 3 model probabilities

Chapter 4

Gaussian Mixture Intensity Filter

This chapter will derive and demonstrate a Gaussian Mixture based implementation of the Intensity filter. This is accomplished by approximating the target intensity function using a weighted GM. The clutter intensity space is modeled by a GM that nearly approximates a uniform background, along with a clutter intensity value. This GM contains equally weighted Gaussian components that cover the scenario space evenly and are unaltered throughout the scenario. The basic prediction and update operations of the iFilter can then be determined for both the target and clutter functions. This is done similar to the SMC based implementation of the iFilter.

A simulated multi-target scenario is used to demonstrate the iFilter's tracking performance. Several metrics are collected and compared to quantify the performance of the new filter versus that of the standard GM-PHD filter. The results show that the GM-iFilter provides a more accurate estimate of cardinality and better overall tracking performance, with no additional cost to computational complexity.

The outline of this chapter is as follows. First, the new GM based implementation of the iFilter is introduced in Section 4.1. The new GM-iFilter is evaluated versus a standard GM-PHD filter in Section 4.2.

4.1 GM-iFilter

In order to implement a Gaussian Mixture based version of the iFilter, first the target intensity is approximated by a weighted Gaussian Mixture.

$$f_k(x) \sim \left\{ \left(m_k^{(i)}, P_k^{(i)}, \omega_k^{(i)} \right) \right\}_{i=1}^{N_k^x} \tag{4.1}$$

$$f_k(x) = \sum_{i=1}^{N_k^x} \omega_k^{(i)} \mathcal{N} \left(x; m_k^{(i)}, P_k^{(i)} \right) \tag{4.2}$$

In the SMC implementation of the iFilter, the clutter space is simply modeled as clutter intensity level (Schikora *et al.*, 2011). The actual clutter distribution is assumed as uniform over the scenario space at each iteration of the filter. A similar model can be achieved for GM-iFilter. First, the clutter intensity is modeled by single value, f_k^{ϕ}, representing the number of clutter targets at time k. The clutter space is set to a static Gaussian Mixture that approximates a uniform distribution. This can be done by arranging a set of equally weighted Gaussian components in a grid that evenly covers the scenario space. As with the homogeneous clutter space in the SMC-iFilter implementation, this distribution does not change at any iteration of the filter. In other words, the Gaussian component values of mean, covariance and weight remain the same and only the overall clutter intensity changes. The GM-iFilter can be predicted and updated through the core iFilter operations, as desceribed in Section 1.3.8, by following some basic GM based techniques.

4.1.1 Predict Targets

First, the predicted target set at the next time step is formed from the union of two sets, the set of previous targets transitioned forward and the set of new targets. That is,

$$f_{k|k-1}(x) \sim \left\{ \left(m_{k|k-1}^{(i)}, P_{k|k-1}^{(i)}, \omega_{k|k-1}^{(i)} \right) \right\}_{i=1}^{N_{k-1}^{x}} \cup \left\{ \left(m_{\mathcal{B}_k}^{(i)}, P_{\mathcal{B}_k}^{(i)}, \omega_{\mathcal{B}_k}^{(i)} \right) \right\}_{i=1}^{N^{\mathcal{B}_k}} \tag{4.3}$$

The set $\left\{ \left(m_{k|k-1}^{(i)}, P_{k|k-1}^{(i)}, \omega_{k|k-1}^{(i)} \right) \right\}_{i=1}^{N_{k-1}^{x}}$ represents the predicted targets that remain in the target space. It is computed as follows:

$$\omega_{k|k-1}^{(i)} = \Psi_k^{X|X} \omega_{k|k-1}^{(i)} \tag{4.4}$$

$$m_{k|k-1}^{(i)} = F_{k-1} m_{k-1}^{(i)} \tag{4.5}$$

$$P_{k|k-1}^{(i)} = F_{k-1} P_{k-1}^{(i)} {F_{k-1}}^{T} + Q_{k-1} \tag{4.6}$$

$$\text{for } i = 1, \ldots, N_{k-1}^{x}$$

where F_{k-1} is the linear state transition matrix and Q_{k-1} is the process noise matrix. Also, $\Psi_k^{X|X}$ is the probability of target survival and its value is constant accross the target space. This is similar to other Gaussian based implementations of filters, such as the GM-PHD (Vo and Ma, 2006) and the GM-MeMBer (Vo *et al.*, 2009).

The set of new targets is the set of clutter targets that have transitioned to the target space. The Gaussian components that represent this set are those in the static Gaussian Mixture representing the clutter space. Only the component target weights are adjusted, so the sum of components represents the expected number of new targets

from clutter. Then,

$$\omega_{\mathcal{B}_k}^{(i)} = \Psi_k^{X|\phi} \cdot f_{k-1}^{\phi} / N^{\mathcal{B}_k} \tag{4.7}$$

where $\Psi_k^{X|\phi}$ is the probability of a target transitioning from clutter into target space (target birth) and is again constant.

4.1.2 Predict Clutter

To predict the clutter intensity forward to time k, the intensity of newly absent targets, as well as current clutter objects, are predicted forward. Then,

$$f_{k|k-1}^{\phi} = b_k(\phi) + d_k(\phi) \tag{4.8}$$

$$\text{where } b_k(\phi) = \Psi_k^{\phi|\phi} \cdot f_{k-1}^{\phi} \tag{4.9}$$

$$\text{and } d_k(\phi) = \Psi_k^{\phi|X} \cdot \sum_{i=1}^{N_{k-1}^x} \omega_k^{(i)} \tag{4.10}$$

where $\Psi_k^{\phi|\phi}$ is the probability of no target presence (target remains in clutter) and $\Psi_k^{\phi|X}$ is the probability of a target transitioning to clutter space (target death). Both these values are constant throughout the scenario space.

4.1.3 Predict Measurement Intensity

The predicted measurement intensity, $\lambda_k(z_k)$, can be determined for each measurement $z_k \in Z_k$ as follows:

$$\lambda_k(z_k) = \nu_k(z_k) + \kappa_k(z_k) \tag{4.11}$$

$$\text{where } \nu_k(z_k) = p_k^D(x) \cdot \sum_{i=1}^{N^x_{k|k-1}} \omega^{(i)}_{k|k-1} \tag{4.12}$$

$$\kappa_k(z_k) = p_k^D(\phi)\, f^{\phi}_{k|k-1} \sum_{i=1}^{N^x_{k|k-1}} \mathcal{N}\left(z_k; H_k m^{(i)}_{k|k-1}, S^{(i)}_k\right) \tag{4.13}$$

$$\text{and } S^{(i)}_k = H_k P^{(i)}_{k|k-1} {H_k}^T + R_k \tag{4.14}$$

In the above, H_k is the measurement transformation matrix and R_k is the measurement covariance matrix. Also, $p_k^D(x)$ and $p_k^D(\phi)$ are the probability of detection of a target and of detection of clutter, respectively. Both values are constant across the scenario space.

4.1.4 Update Target Intensity

The GM-iFilter target intensity can be updated using a set of measurements Z_k. The updated set is represented by the following Gaussian Mixture:

$$\left\{\left(m_k^{(i,j)}, P_k^{(i,j)}, \omega_k^{(i,j)}\right) | i = 1, \ldots, N^x_{k|k-1} \text{ and } \; j = 0, 1, \ldots, |Z_k|\right\} \tag{4.15}$$

where the values of the Gaussian mixture components are computed as follows:

$$\omega_k^{(i,j)} = \frac{\mathcal{N}\left(z_k^{(j)}; H_k m_{k|k-1}^{(i)}, m_{k|k-1}^{(i)}\right)}{\lambda_k\left(z_k^j\right)} \omega_{k|k-1}^{(i)} \tag{4.16}$$

$$m_k^{(i,j)} = m_{k|k-1}^{(i)} + K_k^{(i,j)}\left(z_k^{(j)} - H_k m_{k|k-1}^{(i)}\right) \tag{4.17}$$

$$P_k^{(i,j)} = \left[I - K_k^{(i,j)} H_k\right] P_{k|k-1}^{(i,j)} \tag{4.18}$$

$$K_k^{(i,j)} = P_{k|k-1}^{(i)} {H_k}^T \left[S_k^{(i,j)}\right]^{-1} \tag{4.19}$$

$$S_k^{(i,j)} = H_k P_{k|k-1}^{(i)} {H_k}^T + R_k \tag{4.20}$$

$$\text{for } i = 1, \ldots, N_{k|k-1}^x \text{ and } \; j = 1, \ldots, |Z_k|$$

Also, when $j = 0$ we update each Gaussian component with a missed detection as follows:

$$\omega_k^{(i,0)} = \left(1 - p_k^D(x)\right) \omega_{k|k-1}^{(i)} \tag{4.21}$$

$$m_k^{(i,0)} = m_{k|k-1}^{(i)} \tag{4.22}$$

$$P_k^{(i,0)} = P_{k|k-1}^{(i)} \tag{4.23}$$

$$\text{for } i = 1, \ldots, N_{k|k-1}^x$$

where $p_k^D(x)$ is again the constant probability of target detection.

4.1.5 Update Hypothesis Intensity

The clutter intensity value can be updated using the following equation:

$$f_k^\phi = \left[\left(1 - p_k^D(\phi)\right) + p_k^D(\phi) \sum_{z_k \in Z_k} \frac{p_k(z_k|\phi)}{\lambda_k(z_k)}\right] f_{k|k-1}^\phi \tag{4.24}$$

where $p_k^D(z_k|\phi)$ is the probability of the measurement z_k given it is a clutter detection.

4.1.6 Gaussian Mixture Maintenance

Based on the equations given throughout Sections 4.1.1–4.1.5, the number of Gaussian components in the GM approximation of the target space intensity increases at each iteration of the GM-iFilter. This becomes computationally expensive and eventually intractable over time. Therefore, the number of components must be reduced. The Gaussian Mixture is pruned using minimum weight thresholds, component merge operations as well as a maximum number of components threshold, similar to those used in the GM-PHD and GM-MeMBer filters (Vo and Ma, 2006; Vo *et al.*, 2009). First, the set of Gaussian components whose weights fall below a deletion threshold, T_{del}, are removed from the Gaussian Mixture. Next, pairs of Gaussian components are compared using their statistical distance and similar components, whose distance is below the merge threshold, T_{merge}, are combined. Finally, only the $N_{GM_{MAX}}$ highest weighted Gaussian components are kept as part of the Gaussian Mixture estimate.

4.1.7 State Extraction

At each step of the GM-iFilter a multitarget state estimate can be formed by choosing a random realization of the PPP represented by the GM intensity function of the state space $f_k(x)$. This is accomplished by first estimating the number of targets N_k using (1.82) and setting $g(s) = f_k(x)$. This reduces the estimate of number of targets to the sum of the components weights, just as in (1.89). Next, each of the N_k individual states are estimated by choosing points from the pdf described in (1.83) with, again, $g(s) = f_k(x)$. This can be done by choosing the mean and covariance of the N_k

highest weighted Gaussian components to form the individual target estimates.

4.1.8 GM-iFilter Extensions

Along with a standard Gaussian Mixture implementation given above, several other extension of the iFilter can also be derived using the same framework. This can be accomplished by replacing the Kalman filter prediction equations in (4.4)–(4.6) and update equations in (4.16)–(4.18) with those from either the Extended Kalman Filter (EKF) (Anderson and Moore, 2005)(Jazwinski, 1970) or the Unscented Kalman Filter (UKF) (Julier and Uhlmann, 2004)(Julier and Uhlmann, 1997). This accomplished in a similar manner to the EKF and UKF extension proposed for the GM-PHD (Vo and Ma, 2006) and GM-MeMBer filters (Vo *et al.*, 2009). These new versions are denoted the EKF-iFilter and UKF-iFilter respectively. For brevity, the details of these extensions are not given in this chapter.

4.2 Simulations

4.2.1 Scenario

In order to test the performance of the GM-iFilter, it is tested against a standard GM-PHD filter using a simulated scenario. Consider a two dimensional scenario space of size 100m in each dimension. The scenario contains two crossing targets moving under a near constant velocity motion pattern that each appear and disappear during the simulation. Linear measurements are generated at a time interval of $t = 1$s for 100s. Each target generates a measurement with uniform, state independent probability of $p_D = 0.95$ and each target measurement contains additive Gaussian

noise with standard deviation $\sigma_x = 1$m in each dimension independently. The sensor also generates false measurements with a uniform spatial distribution and density $p_{FA} = 9 \times 10^{-4}$, which translates into an average of $\lambda = 9$ false alarms per scan. Both the target trajectories and the measurements from a single, sample run of the scenario described are shown in Figure 4.1.

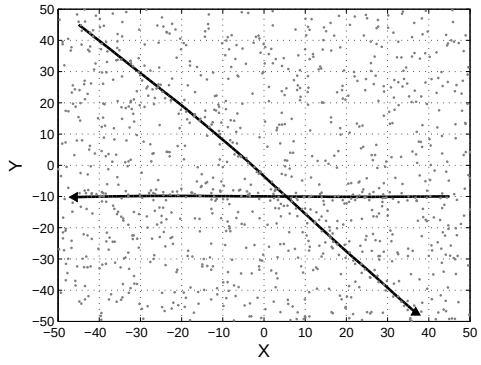

Figure 4.1: Crossing target scenario

4.2.2 Filter Configuration

The scenario was used to run two filters, a GM-iFilter as described in this chapter as well as a GM-PHD filter as described in (Vo and Ma, 2006). Both filters were configured using the same set of parameters, most of which were set to match the scenario parameters wherever possible. Aside from the scenario based parameters, the GM maintenance parameters were set as $T_{del} = 10^{-3}$ $T_{merge} = 2^2$ and $N_{GM_{MAX}} = 1000$ for both filters. The specific parameters of the GM-iFilter set are described in Table 4.1. The unknown clutter space in the GM-iFilter was modeled using a grid that spanned the scenario space. The grid was formed as a Gaussian mixture with means

Table 4.1: GM-iFilter Parameter Values

Parameter	Value
$\Psi_k^{X\|X}$	0.99
$\Psi_k^{\phi\|X}$	0.02
$\Psi_k^{X\|\phi}$	0.02
$\Psi_k^{\phi\|\phi}$	0.99
$p_k^D(x)$	0.95
$p_k^D(\phi)$	0.5

set in a grid along with equal weights and covariances as follows:

$$f(\phi) \sim \left\{ \left(\omega_\phi, [10i, 0, 10j, 0]^T, P_\phi \right) | i, j = -5, -4, \ldots, 4, 5 \right\} \tag{4.25}$$

$$\text{where } \omega_\phi = \frac{1}{11^2} \tag{4.26}$$

$$\text{and } P_\phi = \begin{bmatrix} 10^2 & 0 & 0 & 0 \\ 0 & 1.5^2 & 0 & 0 \\ 0 & 0 & 10^2 & 0 \\ 0 & 0 & 0 & 1.5^2 \end{bmatrix} \tag{4.27}$$

The same Gaussian mixture was also used as the birth intensity for the GM-PHD filter, but with a total birth weight of 0.02, (i.e. $\omega_{\mathcal{B}} = 0.02/11^2$). Both filters were run over 100 Monte Carlo runs of the scenario.

4.2.3 Metrics

In order to analyze the performance of the GM-iFilter versus the standard GM-PHD filter, a variety of metrics are used. The first metric is the cardinality estimate as reported by each of the filters. The second metric used is the Optimal Subpattern Assignment (OSPA) metric (Schuhmacher *et al.*, 2008). The OSPA metric is used

for measuring the overall multitarget estimation accuracy. It encompasses both the cardinality as well as the individual estimate accuracy of the estimate target set versus the true target set. Further details of the OSPA metric can be in found in (Ristic *et al.*, 2010b), however the basic equations are defined as follows:

$$\bar{d}_p^{(c)}(X,Y) = \left(\frac{1}{n} \left(\min_{\pi \in \Pi_k} \sum_{i=1}^{m} d^{(c)}\left(x_i, y_{\pi(i)}\right)^p + c^p \left(n-m\right) \right) \right)^{\frac{1}{p}} \tag{4.28}$$

where $d^{(c)}(x,y) = \min\left\{c, \|x-y\|\right\}$ and Π_k is the set of permutations of the set $\{1,\ldots,k\}$. Also note that in general $p \geq 1$, $c > 0$, $|X| = m$ and $|Y| = n$. In this simulation, the OSPA configuration values of $c = 20, p = 1$ were used. Finally, the computational complexity of the filter are measured by computing the total run time of each of the filters.

4.2.4 Results

The scenario described above was processed using each of the filters for 100 Monte Carlo runs. The metric results are described in this section. First, the mean cardinality results of each of the filters are plotted against the true target cardinality in Figure 4.2. As seen in the Figure, the GM-PHD filter initially provides a more accurate estimate of cardinality. However, the GM-iFilter eventaully provides a more accurate estimate of the scenario cardinality than the GM-PHD filter for the remainder of the scenario. The mean GM-iFilter cardinality value is an accurate estimate of the true cardinality, while the mean GM-PHD filter cardinality estimate is slightly higher than the true value. This indicates that the GM-PHD filter provides additional or spurious track estimates more often than the GM-iFilter.

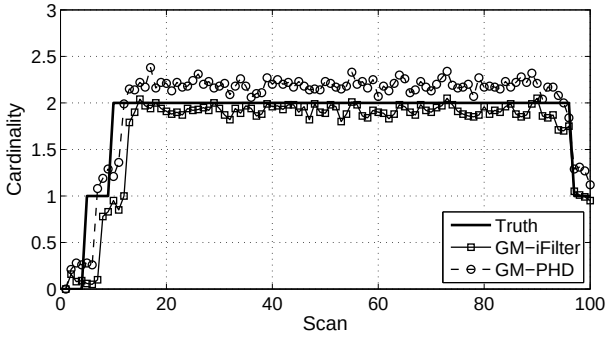

Figure 4.2: Mean filter cardinality estimates

The mean OSPA results for each filter over all Monte Carlo runs of the above simulation are shown in Figure 4.3. As seen in the Figure, the GM-iFilter estimates outperforms the GM-PHD filter throughout the scenario. This is consistent with the cardinality results given in Figure 4.2.

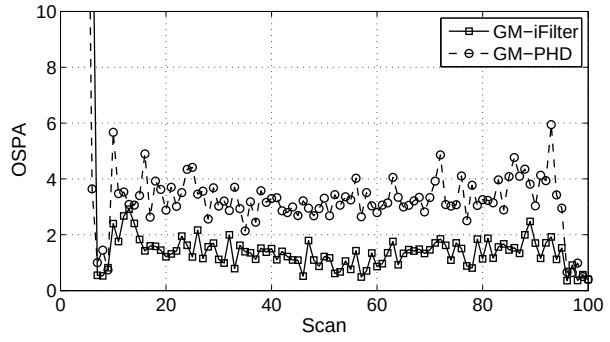

Figure 4.3: OSPA Results

The complexity of each of the filters was measured using the total run time of each of the filters. The GM-iFilter had a mean run time of 60.9s with a standard deviation of 2.62s while the GM-PHD had a mean run time of 63.2s with a standard deviation of 3.01s. These results show that there GM-iFilter has essentially the same

computational requirements as the GM-PHD filter. Thus, the GM-iFilter can provide improved tracking results without any increase in computational load.

Chapter 5

Conclusions

This thesis contributed three important improvements to three different Random Finite Set based multitarget tracking algorithms.

The WPPHD filter was shown to be a viable method for tracking targets continually over time using the PHD framework. Partitioning a PHD surface based on both spatial characteristics and expected target weight is both a natural and straight forward way of identifying individual targets within the multitarget set estimate. The resulting estimates were ensured to represent a single target in both its total weight as well as its spatial location. The labeling of these partitions allowed for the continual monitoring of these objects over time. Two distinct implementations, CLEAN-SMC-WPPHD and LM-GM-WPPHD, of the WPPHD filter were shown to be functional multitarget tracking algorithms. The CLEAN based SMC implementation was shown to have some improved track continuity when compared to previously known basic track continuity techniques. It also had increased accuracy for nearby track initialization. However, these tracking improvements came at the expense of slightly higher

computational time. The LM-GM implementation demonstrated an increase in tracking accuracy throughout the scenario. The track continuity was very much improved and the track completeness ratio was slightly improved. These improvements came with only a slight increase in computational time.

The MM-CBMeMBer filter was given as an extension to the Cardinality Balanced Multitarget Multi-Bernoulli filter for tracking manoeuvring targets. Extensions for both SMC and GM implementations were derived. Simulations showed that each implementation was an improved option for tracking multiple manoeuvring targets against its basic counterparts. The MM-CBMeMBer implementations showed improvement in both cardinality estimation, as well as, overall localization. Both MM-CBMeMBer implementations showed to be capable methods for estimating the motion model that a manoeuvring target may be traveling under. These increased tracking benefits were accomplished with only a minor increase in the computational requirements of the tracking algorithms.

The new Gaussian Mixture based implementation of the intensity filter for scenarios containing certain linear properties was derived. Other possible EKF and UKF based extensions for non-linear motion models and measurement transformations were also suggested. Simulations show that the GM-iFilter provides improved estimates of both cardinality and overall tracking metrics without any incurred cost of computation.

5.1 Future Work

There remains ample opportunity for future work in each of the methods described in this thesis.

The idea of using weight based partitioning and monitoring of individual elements in PHD filters is still a new concept. The algorithms and ideas described in this thesis represent only some of the possible implementations of the WPPHD concept. The partition and label propagation schemes described in this paper can be substituted, extended and enhanced to provide new, possibly better, WPPHD based filtering algorithms. Furthermore, other WPPHD implementations including a Gaussian Particle (GP-PHD) based implementation of the WPPHD as well the partitioning of the Cardinalized PHD filter are likely possible.

The techniques outlined for the MM-CBMeMBer filter were only the basic implementations of multiple model tracking in the CBMeMBer filter. There is no doubt room for improving the implementations and configuration of the techniques described in this work in order to produce better overal tracking results. As well, the MM concept may also be applied to other implementations of the CBMeMBer filter, such as the Gaussian Particle implementation. The MeMBer filter is one of the less developed and implemented filters in the FISST family. Its native design of target separation gives it a distinct advantage over other more popular RFS based filter techniques, such as PHD filters. With further research, the CBMeMBer filter has the possibility of being an equally or more powerful multitarget tracking technique. Extending the CBMeMBer filter is just one of several key advancements that can be applied to the

CBMeMBer filter in order to make it a more viable multitarget tracking candidate. Other MeMBer filter extensions, such as estimate smoothing, as well as analyzing the viability of using CBMeMBer filtering in a variety of tracking environments such as dim targets, extended targets and track before detect (TBD) are also key future research areas for the MeMBer filter.

The iFilter is still a relative new and unexplored multitarget tracking algorithm that is only in its infancy. Beyond the 2 basic SMC and GM based implementations, only a small amount of research has been completed. Some of extensions that have been explored within the other filters referenced in this thesis could also be explored within the iFilter context. In the current implementations, the iFilter acts only as an estimator of the multitarget state and could be extended to allow for monitoring individual estimates continually, over multiple scans, using mechanisms similar to those used with the PHD filter. As well, other important tracking improvements and scenarios could be examined such as manoeuvring targets, track before detect (TBD) and smoothing the iFilter estimate.

Appendix A

WPPHD Derivations

The WPPHD decomposes the PHD surface into the summation of smaller distinct PHD surfaces that represent individual targets within the surveillance area. In the WPPHD the entire PHD surafce is predicted and updated and the summation of the partitions are intact afterwards. This is due to the near linear-like properties of the PHD prediction and update operations as described in (1.24) and (1.26). This appendix demonstrates the closure and the sometimes, linearity of applying the PHD predicition and update operation to the WPPHD summation.

A.1 WPPHD Prediction

The prediction operation on a WPPHD summation breaks down as follows:

$$
\begin{aligned}
&D_{k|k-1}\left(x|Z^{(k)}\right) = b_{k|k-1}(x) + \dots \\
&\quad \int F_{k|k-1}(x|w) \cdot D_{k-1|k-1}\left(w|Z^{(k-1)}\right) dw && \text{(A.1)} \\
&= b_{k|k-1}(x) + \dots \\
&\quad \int F_{k|k-1}(x|w) \cdot \sum_{j=1}^{T_k} D^{(j)}_{k-1|k-1}\left(w\middle|Z^{(k-1)}\right) dw && \text{(A.2)} \\
&= b_{k|k-1}(x) + \dots \\
&\quad \int \sum_{j=1}^{T_k} F_{k|k-1}(x|w) \cdot D^{(j)}_{k-1|k-1}\left(w\middle|Z^{(k-1)}\right) dw && \text{(A.3)} \\
&= b_{k|k-1}(x) + \dots \\
&\quad \sum_{j=1}^{T_k} \int F_{k|k-1}(x|w) \cdot D^{(j)}_{k-1|k-1}\left(w\middle|Z^{(k-1)}\right) dw && \text{(A.4)}
\end{aligned}
$$

Here the PHD prediction operation is not quite linear as the addition of only a single birth partition is included. However, the operation is closed, as the birth intensity is itself a PHD intensity independent of any previous partitions and thus the mapping is a summation of PHD surfaces.

A.2 WPPHD Update

$$D_{k|k}\left(x|Z^{(k)}\right) = \Bigg[1 - p_D(x) + \cdots \sum_{z\in Z_k} \frac{p_D(x)\, f_{k+1}(z|x)}{\lambda c(z) + D_{k|k-1}\left[p_D L_z\right]}\Bigg] \cdot D_{k|k-1}\big(x|Z^{(k)}\big) \tag{A.5}$$

$$D_{k|k}\left(x|Z^{(k)}\right) = \Bigg[1 - p_D(x) + \cdots \sum_{z\in Z_k} \frac{p_D(x)\, f_{k+1}(z|x)}{\lambda c(z) + D_{k|k-1}\left[p_D L_z\right]}\Bigg] \cdot \sum_{j=1}^{T_k} D^{(j)}_{k|k-1}\big(x\big|Z^{(k)}\big) \tag{A.6}$$

$$D_{k|k}\left(x|Z^{(k)}\right) = \sum_{j=1}^{T_k} \Bigg(\Bigg[1 - p_D(x) + \cdots \sum_{z\in sZ_k} \frac{p_D(x)\, f_{k+1}(z|x)}{\lambda c(z) + D_{k|k-1}\left[p_D L_z\right]}\Bigg] \cdot D^{(j)}_{k|k-1}\big(x\big|Z^{(k)}\big)\Bigg) \tag{A.7}$$

Here it can be seen that the PHD update operation described in (1.26) is a completely linear operation when performed on the WPPHD summation. Thus performing the update operation on a summation of independent partitions can also be completed by performing a single update step on the entire PHD surface.

Appendix B

MM-MeMBer Derivations

This appendix will give some of the detailed derivations used in the construction of the SMC and GM MM-MeMBer filters.

B.1 SMC-MM-MeMBer

The equations for the prediction and update of the SMC-MM-MeMBer are derived by substituting the SMC approximation into the original MM-MeMBer prediction and update equations given in 3.2. The derivation of the key elements of those equations are shown.

B.1.1 SMC-MM-MeMBer Prediction

First, the derivation of the total survival weight over all models $\left\langle p_{k-1}^{(j)}, p_S \right\rangle$ is as follows:

$$\left\langle p_{k-1}^{(j)}, p_S \right\rangle \tag{B.8}$$

$$= \sum_{\pi_{k-1} \in \Pi_{k-1}} \int_{X_{k-1}} p_{k-1}^{(j)}(x_{k-1}, \pi_{k-1})\, p_S(x_{k-1}, \pi_{k-1})\, dx_{k-1} \tag{B.9}$$

$$= \sum_{\pi_{k-1} \in \Pi_{k-1}} \int_{X_{k-1}} p_S(x_{k-1}, \pi_{k-1}) \sum_{i=1}^{L_{k-1}^{(j)}} \omega_{k-1}^{(i,j)} \delta_{x_{k-1}^{(i,j)}, \pi_{k-1}^{(i,j)}}(x_{k-1}, \pi_{k-1})\, dx_{k-1} \tag{B.10}$$

$$= \sum_{i=1}^{L_{k-1}^{(j)}} \sum_{\pi_{k-1} \in \Pi_{k-1}} \int_{X_{k-1}} p_S(x_{k-1}, \pi_{k-1})\, \omega_{k-1}^{(i,j)} \delta_{x_{k-1}^{(i,j)}, \pi_{k-1}^{(i,j)}}(x_{k-1}, \pi_{k-1})\, dx_{k-1} \tag{B.11}$$

$$= \sum_{i=1}^{L_{k-1}^{(j)}} p_S\left(x_{k-1}^{(i,j)}, \pi_{k-1}^{(i,j)}\right) \omega_{k-1}^{(i,j)} \tag{B.12}$$

Next, the derivation of the multiple model target prediction integration

$\left\langle f_{k|k-1}(x,\pi|\cdot)\,, p_{k-1}^{(j)} p_S \right\rangle$ is as follows:

$$\left\langle f_{k|k-1}(x,\pi|\cdot)\,, p_{k-1}^{(j)} p_S \right\rangle \tag{B.13}$$

$$= \sum_{\pi_{k-1}\in\Pi} \int_{X_{k-1}} p\Big(\pi\Big|\pi_{k-1}\Big)\, f_{k|k-1}(x|\pi, x_{k-1}, \pi_{k-1})\, p_{k-1}^{(j)}(x_{k-1}, \pi_{k-1})\, p_S(x_{k-1}, \pi_{k-1})\, dx_{k-1} \tag{B.14}$$

$$= \sum_{\pi_{k-1}\in\Pi} \int_{X_{k-1}} p\Big(\pi\Big|\pi_{k-1}\Big)\, f_{k|k-1}(x|\pi, x_{k-1}, \pi_{k-1})\, p_S(x_{k-1}, \pi_{k-1}) \cdots$$

$$\cdots \sum_{i=1}^{L_{k-1}^{(j)}} \omega_{k-1}^{(i,j)} \delta_{x_{k-1}^{(i,j)}\pi_{k-1}^{(i,j)}}(x,\pi)\, dx_{k-1} \tag{B.15}$$

$$= \sum_{i=1}^{L_{k-1}^{(j)}} \sum_{\pi_{k-1}\in\Pi} \int_{X_{k-1}} p\Big(\pi\Big|\pi_{k-1}\Big)\, f_{k|k-1}(x|\pi, x_{k-1}, \pi_{k-1})\, p_S(x_{k-1}, \pi_{k-1}) \cdots$$

$$\cdots \omega_{k-1}^{(i,j)} \delta_{x_{k-1}^{(i,j)}\pi_{k-1}^{(i,j)}}(x,\pi)\, dx_{k-1} \tag{B.16}$$

$$= \sum_{i=1}^{L_{k-1}^{(j)}} p\Big(\pi\Big|\pi_{k-1}^{(i,j)}\Big)\, f_{k|k-1}\Big(x|\pi, x_{k-1}^{(i,j)}\pi_{k-1}^{(i,j)}\Big)\, p_S\Big(x_{k-1}^{(i,j)}\pi_{k-1}^{(i,j)}\Big)\, \omega_{k-1}^{(i,j)} \tag{B.17}$$

$$\text{where } \pi_{k|k-1}^{(i,j)} \sim \begin{cases} q_\pi\Big(\cdot|\pi_{k-1}^{(i)}\Big) & i = 1,\ldots,L_{k-1} \\ \beta_\pi(\cdot) & i = L_{k-1}+1,\ldots,L_{k-1}+J_k \end{cases} \tag{B.18}$$

$$\text{for } j = 1,\ldots,M_{k-1}$$

B.1.2 SMC-MM-MeMBer Update

The update equation of the MM-MeMBer filter contains the multiple model total measurement intensity value $\left\langle p_{k|k-1}^{(j)}, \Psi_{k,z} \right\rangle$. The SMC implementation of this integration

is derived as follows:

$$\left\langle p^{(j)}_{k|k-1}, \Psi_{k,z} \right\rangle \tag{B.19}$$

$$= \sum_{\pi_{k|k-1} \in \Pi_{k|k-1}} \int_{X_{k|k-1}} p^{(j)}_{k|k-1}\left(x_{k|k-1}, \pi_{k|k-1}\right), p_D\left(x_{k|k-1}, \pi_{k|k-1}\right) dx_{k|k-1} \tag{B.20}$$

$$= \sum_{\pi_{k|k-1} \in \Pi_{k|k-1}} \int_{X_{k|k-1}} p_D\left(x_{k|k-1}, \pi_{k|k-1}\right) \sum_{i=1}^{L^{(j)}_{k|k-1}} \omega^{(i,j)}_{k|k-1} \delta_{x^{(i,j)}_{k|k-1}, \pi_{k|k-1}}\left(x, \pi_{k|k-1}\right) dx_{k|k-1} \tag{B.21}$$

$$= \sum_{\pi_{k|k-1} \in \Pi_{k|k-1}} \sum_{i=1}^{L^{(j)}_{k|k-1}} \int_{X_{k|k-1}} p_D\left(x_{k|k-1}, \pi_{k|k-1}\right) \omega^{(i,j)}_{k|k-1} \delta_{x^{(i,j)}_{k|k-1}, \pi^{(i,j)}_{k|k-1}}\left(x, \pi_{k|k-1}\right) dx_{k|k-1} \tag{B.22}$$

$$= \sum_{\pi_{k|k-1} \in \Pi_{k|k-1}} \sum_{i=1}^{L^{(j)}_{k|k-1}} p_D\left(x^{(i,j)}_{k|k-1}, \pi^{(i,j)}_{k|k-1}\right) \omega^{(i,j)}_{k|k-1} \tag{B.23}$$

B.2 GM-MM-MeMBer

The equations for the prediction and update of the GM-MM-MeMBer are derived similar to those of the SMC implementations. A GM approximation is substituted into the original MM-MeMBer prediction and update equations given in 3.2. The key derivations are given here.

B.2.1 GM-MM-MeMBer Predict

The following multiple model GM derivations are used to derive the prediction equations in (3.47)−(3.51). First, the total target survival intensity $\left\langle p^{(j)}_{k-1}, p_S \right\rangle$ is computed

as follows:

$$\left\langle p_{k-1}^{(j)}, p_S \right\rangle \tag{B.24}$$

$$= \sum_{\pi_{k-1} \in \Pi} \int_{X_{k-1}} p_{k-1}^{(j)}(x_{k-1}, \pi_{k-1})\, p_S(x_{k-1}, \pi_{k-1})\, dx_{k-1} \tag{B.25}$$

$$= \sum_{\pi_{k-1} \in \Pi} \int_{X_{k-1}} p_S(\pi_{k-1}) \sum_{i=1}^{M_{k-1}^{(j)}} \omega_{k-1}^{(i,j)}(\pi_{k-1}) \mathcal{N}\left(x_{k-1}; m_{k-1}^{(i,j)}(\pi_{k-1}), P_{k-1}^{(i,j)}(\pi_{k-1})\right) dx_{k-1} \tag{B.26}$$

$$= \sum_{\pi_{k-1} \in \Pi} \sum_{i=1}^{M_{k-1}^{(j)}} p_S(\pi_{k-1})\, \omega_{k-1}^{(i,j)}(\pi_{k-1}) \int_{X_{k-1}} \mathcal{N}\left(x_{k-1}; m_{k-1}^{(i,j)}(\pi_{k-1}), P_{k-1}^{(i,j)}(\pi_{k-1})\right) dx_{k-1} \tag{B.27}$$

$$= \sum_{\pi_{k-1} \in \Pi} \sum_{i=1}^{M_{k-1}^{(j)}} p_S(\pi_{k-1})\, \omega_{k-1}^{(i,j)}(\pi_{k-1}) \tag{B.28}$$

Next, the mutliple model target prediction integral is computed as follows:

$$\left\langle f_{k|k-1}(x_k,\pi_k|\cdot)\,,p_{k-1}^{(j)}p_S\right\rangle \tag{B.29}$$

$$= \sum_{\pi_{k-1}\in\Pi}\int_{X_{k-1}} p\Big(\pi_k\Big|\pi_{k-1}\Big)\, f_{k|k-1}(x_k|\pi_k,x_{k-1},\pi_{k-1})\, p_{k-1}^{(j)}(x_{k-1},\pi_{k-1})\, p_S(\pi_{k-1})\, dx_{k-1} \tag{B.30}$$

$$= \sum_{\pi_{k-1}\in\Pi}\int_{X_{k-1}} p\Big(\pi_k\Big|\pi_{k-1}\Big)\, p_S(\pi_{k-1})\,\mathcal{N}\left(x_k;F_{k-1}(\pi_k)x_{k-1},Q_{k-1}(\pi_k)\right)\dots$$

$$\dots\sum_{i=1}^{M_k^{(j)}}\omega_{k-1}^{(i,j)}(\pi_{k-1})\mathcal{N}\left(x_{k-1};m_{k-1}^{(i,j)}(\pi_{k-1}),P_{k-1}^{(i,j)}(\pi_{k-1})\right)dx_{k-1} \tag{B.31}$$

$$= \sum_{\pi_{k-1}\in\Pi}\sum_{i=1}^{M_k^{(j)}} p\Big(\pi_k\Big|\pi_{k-1}\Big)\, p_S(\pi_{k-1})\,\omega_{k-1}^{(i,j)}(\pi_{k-1})\dots$$

$$\dots\int_{X_{k-1}}\mathcal{N}\left(x_k;F_{k-1}(\pi_k)x_{k-1},Q_{k-1}(\pi_k)\right)\mathcal{N}\left(x_{k-1};m_{k-1}^{(i,j)}(\pi_{k-1}),P_{k-1}^{(i,j)}(\pi_{k-1})\right)dx_{k-1} \tag{B.32}$$

$$= \sum_{\pi_{k-1}\in\Pi}\sum_{i=1}^{M_k^{(j)}} p\Big(\pi_k\Big|\pi_{k-1}\Big)\, p_S(\pi_{k-1})\,\omega_{k-1}^{(i,j)}(\pi_{k-1})\dots$$

$$\mathcal{N}\left(x_k;F_{k-1}(\pi_k)m_{k-1}^{(i,j)},Q_{k-1}(\pi_k)+F_{k-1}(\pi_k)P_{k-1}^{(i,j)}F_{k-1}(\pi_k)^T\right)\cdots$$

$$\cdots\mathcal{N}\left(x_{k-1};m_{k-1}^{(i,j)}(\pi_{k-1}),P_{k-1}^{(i,j)}(\pi_{k-1})\right) \tag{B.33}$$

B.2.2 GM-MM-MeMBer Update

The GM-MM-MeMBer update steps are also derived by substituting the GM-MM approximation into the original MM-MeMBer update equations. The key derivations are given here. The multiple model total measurement intensity $\left\langle p_{k|k-1}^{(j)},\Psi_{k,z}\right\rangle$ is computed as follows:

$$\left\langle p_{k|k-1}^{(j)}, \Psi_{k,z} \right\rangle \tag{B.34}$$

$$= \sum_{\pi_{k|k-1} \in \Pi_{k|k-1}} \int_{X_{k|k-1}} p_{k|k-1}^{(j)}\big(x_{k|k-1}, \pi_{k|k-1}\big)\, p_D\big(x_{k|k-1}, \pi_{k|k-1}\big)\, g_k\big(z|x_{k|k-1}, \pi_{k|k-1}\big)\, dx_{k|k-1}$$

(B.35)

$$= \sum_{\pi_{k|k-1} \in \Pi_{k|k-1}} p_D\big(\pi_{k|k-1}\big) \sum_{i=1}^{M_k^{(j)}} \omega_{k|k-1}^{(i,j)}(\pi_{k|k-1}) \ldots$$

$$\ldots \int_{X_{k|k-1}} \mathcal{N}\left(z; H_k(\pi_{k|k-1})x_{k-1}, R_k(\pi_{k|k-1})\right) \ldots$$

$$\ldots \mathcal{N}\left(x_{k|k-1}; m_{k|k-1}^{(i,j)}(\pi_{k|k-1}), P_{k|k-1}^{(i,j)}(\pi_{k|k-1})\right) dx_{k|k-1} \tag{B.36}$$

$$= \sum_{\pi_{k|k-1} \in \Pi_{k|k-1}} p_D\big(\pi_{k|k-1}\big) \sum_{i=1}^{M_k^{(j)}} \omega_{k|k-1}^{(i,j)}(\pi_{k|k-1}) \int_{X_{k|k-1}} q_k(z)\, \mathcal{N}\left(x_{k|k-1}; m_k^{(i,j)}, P_k^{(i,j)}\right) dx_{k|k-1}$$

(B.37)

$$= \sum_{\pi_{k|k-1} \in \Pi_{k|k-1}} p_D\big(\pi_{k|k-1}\big) \sum_{i=1}^{M_k^{(j)}} q_k(z)\, \omega_{k|k-1}^{(i,j)}(\pi_{k|k-1}) \tag{B.38}$$

$$\text{where } q_k(z) = \mathcal{N}\left(z; H_k(\pi_{k|k-1}), H_k(\pi_{k|k-1}) P_k^{(i,j)} H_k(\pi_{k|k-1})^T + R_k(\pi_{k|k-1})\right)$$

Bibliography

Anderson, B. and Moore, J. (2005). *Optimal Filtering*. Dover Publications.

Bar-Shalom, Y. and Fortmann, T. (1988). *Tracking and data association*. Mathematics in science and engineering. Academic Press.

Bar-Shalom, Y. and Li, X. (1995). *Multitarget Multisensor Tracking: Principles and Techniques*. YBS Publishing.

Bar-Shalom, Y., Kirubarajan, T., and Li, X.-R. (2002). *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, Inc., New York, NY, USA.

Blackman, S. (1986). *Multiple-target tracking with radar applications*. Artech House, Inc.

Blackman, S. and Popoli, R. (1999). *Design and Analysis of Modern Tracking Systems*. Artech House Publishers.

Blom, H. (1984). An efficient filter for abruptly changing systems. In *Decision and Control, 1984. The 23rd IEEE Conference on*, volume 23, pages 656 –658.

Bose, R., Freedman, A., and Steinberg, B. (2002). Sequence CLEAN: a modified

deconvolution technique for microwave images of contiguous targets. *IEEE Transactions on Aerospace and Electronic Systems*, **38**(1), 89–97.

Clark, D. and Bell, J. (2005). Data association for the PHD filter. In *Proceedings of the 2005 International Conference on Intelligent Sensors, Sensor Networks and Information Processing Conference 2005*, pages 217–222.

Clark, D. and Bell, J. (2006). Convergence results for the particle PHD filter. *IEEE Transactions on Signal Processing*, **54**(7), 2652–2661.

Clark, D. and Bell, J. (2007). Multi-target state estimation and track continuity for the particle PHD filter. *IEEE Transactions on Aerospace and Electronic Systems*, **43**(4), 1441–1453.

Coraluppi, S., Grimmett, D., and de Theije, P. (2006). Benchmark evaluation of multistatic trackers. In *9th International Conference on Information Fusion 2006*, pages 1–7.

Daum, F. and Huang, J. (2003). Curse of dimensionality and particle filters. In *Proceedings 2003 IEEE Aerospace Conference*, volume 4, pages 1979–1993.

Dunne, D. and Kirubarajan, T. (2011). Weight partitioned probability hypothesis density filters. In *Proceedings of the 14th International Conference on Information Fusion (FUSION) 2011*, pages 1–8.

Dunne, D. and Kirubarajan, T. (2012a). MeMBer filter for manoeuvring targets. In *Proceedings of SPIE Vol. 8392: Signal Processing, Sensor Fusion, and Target Recognition XXI*.

Dunne, D. and Kirubarajan, T. (2012b). Multiple model multi-bernoulli filters for manoeuvering targets. Accepted in *IEEE Transactions on Aerospace and Electronic Systems*.

Dunne, D. and Kirubarajan, T. (2013). Gaussian mixture intensity filter. To be submitted to *IEEE Transactions on Aerospace and Electronic Systems*.

Dunne, D., Ratnasingham, T., Lang, T., and Kirubarajan, T. (2009). SMC-PHD-based multi-target tracking with reduced peak extraction. In *Proc. SPIE 7445-Signal Processing, Sensor Fusion, and Target Recognition XX*.

Dunne, D., Chen, X., and Kirubarajan, T. (2012). Weight partitioned probability hypothesis density filters for multitarget tracking. Under 2nd Review with *Signal Processing*.

El-Fallah, A., Ravichandran, R., Mehra, R., Hoffman, J., Zajic, T., Stelzig, C., Mahler, R., and Alford, M. (2001). Scientific performance evaluation for distributed sensor management and adaptive data fusion. In *SPIE Proc. 4380-Signal Processing, Sensor Fusion, and Target Recognition*, pages 328–338.

Erdinc, O., Willett, P., and Bar-Shalom, Y. (2005). Probability hypothesis density filter for multitarget multisensor tracking. In *8th International Conference on Information Fusion 2005*, volume 1.

Erdinc, O., Willett, P., and Bar-Shalom, Y. (2006). Physical-space approach for the probability hypothesis density and cardinalized probability hypothesis density filters. In *Proc. SPIE 6236, 623619-Signal and Data Procesing of Small Targets 2006*.

Erdinc, O., Willett, P., and Bar-Shalom, Y. (2009). The bin-occupancy filter and its connection to the phd filters. *IEEE Transactions on Signal Processing*, **57**(11), 4232–4246.

Fortmann, T., Bar-Shalom, Y., and Scheffe, M. (1980). Multi-target tracking using joint probabilistic data association. In *Decision and Control including the Symposium on Adaptive Processes, 1980 19th IEEE Conference on*, volume 19, pages 807 –812.

Gadsden, S., Dunne, D., Habibi, S., and Kirubarajan, T. (2009). Comparison of extended and unscented Kalman, particle, and smooth variable structure filters on a bearing-only target tracking problem. In *Proceedings of SPIE Vol. 7445: Signal and Data Processing of Small Targets*.

Gadsden, S., Habibi, S., and Kirubarajan, T. (2010). A novel interacting multiple model method for nonlinear target tracking. In *13th Conference on Information Fusion (FUSION) 2011*, pages 1–8.

Gadsden, S., Dunne, D., Tharmarasa, R., Habibi, S., and Kirubarajan, T. (2011a). Application of the smooth variable structure filter to a multi-target tracking problem. In *Proceedings of SPIE Vol. 8050: Signal Processing, Sensor Fusion, and Target Recognition XX*.

Gadsden, S., Dunne, D., Habibi, S., and Kirubarajan, T. (2011b). Combined particle and smooth variable structure filtering for nonlinear estimation problems. In *Information Fusion (FUSION), 2011 Proceedings of the 14th International Conference on*, pages 1 –8.

Gadsden, S., Habibi, S., Dunne, D., and Kirubarajan, T. (2012). Nonlinear estimation techniques applied on target tracking problems. *Journal of Dynamic Systems, Measurement, and Control*, **134**(5), 054501.

Goodman, I., Mahler, R., and Nguyen, H. (1997). *Mathematics of Data Fusion*. Theory and decision library: Series B : Mathematical and Statistical Methods. Springer.

Gordon, N., Salmond, D., and Smith, A. (1993). Novel approach to nonlinear/non-gaussian bayesian state estimation. *Radar and Signal Processing, IEE Proceedings F*, **140**(2), 107 –113.

Hoffman, J. and Mahler, R. (2004). Multitarget miss distance via optimal assignment. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, **34**(3), 327–336.

Högbom, J. A. (1974). Aperture synthesis with a non-regular distribution of interferometer baselines. *Astronomy and Astrophysics Supplement*, **15**, 417.

Jazwinski, A. (1970). *Stochastic Processes and Filtering Theory*. Elsevier Science.

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

Julier, S. and Uhlmann, J. (2004). Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, **92**(3), 401 – 422.

Kingman, J. (1993). *Poisson Processes*. Oxford University Press, USA.

Li, X. and Bar-Shalom, Y. (1993). Design of an interacting multiple model algorithm for air traffic control tracking. *Control Systems Technology, IEEE Transactions on*, **1**(3), 186 –194.

Li, X.-R. and Jilkov, V. (2003). Survey of maneuvering target tracking. Part I. Dynamic models. *IEEE Transactions on Aerospace and Electronic Systems*, **39**(4), 1333 – 1364.

Li, X.-R. and Jilkov, V. (2005). Survey of maneuvering target tracking. Part V. Multiple-model methods. *IEEE Transactions on Aerospace and Electronic Systems*, **41**(4), 1255–1321.

Li, X. R., Zwi, X., and Zwang, Y. (1999). Multiple-model estimation with variable structure. III. Model-group switching algorithm. *Aerospace and Electronic Systems, IEEE Transactions on*, **35**(1), 225 –241.

Lin, L., Bar-Shalom, Y., and Kirubarajan, T. (2004). Data association combined with the probability hypothesis density filter for multitarget tracking. In *Proc. SPIE 5428*.

Lin, L., Bar-Shalom, Y., and Kirubarajan, T. (2006). Track labeling and PHD filter for multitarget tracking. *IEEE Transactions on Aerospace and Electronic Systems*, **42**(3), 778–795.

Mahler, R. (2001). A survey of PHD filter and CPHD filter implementations. In *SPIE Proc. Signal Processing, Sensor Fusion, and Target Recognition*.

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

Mahler, R. (2007a). PHD filters of higher order in target number. *IEEE Transactions on Aerospace and Electronic Systems*, **43**(4), 1523 –1543.

Mahler, R. (2007b). *Statistical Multisource-Multitarget Information Fusion*. Artech House, Inc., Norwood, MA, USA.

Mahler, R., Vo, B.-T., and Vo, B.-N. (2012). Forward-backward probability hypothesis density smoothing. *Aerospace and Electronic Systems, IEEE Transactions on*, **48**(1), 707–728.

McDonald, M., Dunne, D., Damini, A., and Kirubarajan, T. (2009). Event-based characterization and simulation of sea clutter. In *Proceedings of SPIE 7445: Signal and Data Processing of Small Targets*.

Musicki, D. and La Scala, B. (2008). Multi-target tracking in clutter without measurement assignment. *IEEE Transactions on Aerospace and Electronic Systems*, **44**(3), 877–896.

Nadarajah, N., Kirubarajan, T., Lang, T., Mcdonald, M., and Punithakumar, K. (2011). Multitarget tracking using probability hypothesis density smoothing. *Aerospace and Electronic Systems, IEEE Transactions on*, **47**(4), 2344–2360.

Panta, K., Ba-Ngu-Vo, and Clark, D. (2006). An efficient track management scheme for the Gaussian-mixture probability hypothesis density tracker. In *Fourth International Conference on Intelligent Sensing and Information Processing, 2006. ICISIP 2006*, pages 230–235.

Panta, K., Ba-Ngu, V., and Singh, S. (2007). Novel data association schemes for

the probability hypothesis density filter. *IEEE Transactions on Aerospace and Electronic Systems*, **43**(2), 556–570.

Pasha, A., Vo, B., Tuan, H., and Ma, W.-K. (2006). Closed form PHD filtering for linear jump Markov models. In *9th International Conference on Information Fusion 2006*, pages 1 –8.

Pasha, A., Vo, B.-N., Tuan, H., and Ma, W.-K. (2009). A Gaussian mixture PHD filter for jump Markov system models. *IEEE Transactions on Aerospace and Electronic Systems*, **45**(3), 919 –936.

Punithakumar, K., Kirubarajan, T., and Sinha, A. (2005). A sequential Monte Carlo probability hypothesis density algorithm for multitarget track-before-detect. In *Proceedings of SPIE 5913: Signal and Data Processing of Small Targets*.

Punithakumar, K., Kirubarajan, T., and Sinha, A. (2008). Multiple-model probability hypothesis density filter for tracking maneuvering targets. *IEEE Transactions on Aerospace and Electronic Systems*, **44**(1), 87–98.

Reid, D. (1979). An algorithm for tracking multiple targets. *Automatic Control, IEEE Transactions on*, **24**(6), 843 – 854.

Ristic, B., Arulampalm, S., and Gordon, N. (2004). *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House, Boston, Ma.

Ristic, B., Clark, D., and Vo, B.-N. (2010a). Improved SMC implementation of the PHD filter. In *13th Conference on Information Fusion (FUSION) 2010*, pages 1 –8.

Ristic, B., Vo, B.-N., and Clark, D. (2010b). Performance evaluation of multi-target tracking using the OSPA metric. In *13th Conference on Information Fusion (FUSION) 2010*, pages 1–7.

Schikora, M., Koch, W., Streit, R., and Cremers, D. (2011). Sequential Monte Carlo method for the iFilter. In *Information Fusion (FUSION), 2011 Proceedings of the 14th International Conference on*, pages 1–8.

Schuhmacher, D., Vo, B.-T., and Vo, B.-N. (2008). A consistent metric for performance evaluation of multi-object filters. *IEEE Transactions on Signal Processing*, **56**(8), 3447–3457.

Streit, R. (2008). Multisensor multitarget intensity filter. In *11th International Conference on Information Fusion 2008*, pages 1–8.

Streit, R. (2009). PHD intensity filtering is one step of a MAP estimation algorithm for positron emission tomography. In *Information Fusion, 2009. FUSION '09. 12th International Conference on*, pages 308 –315.

Streit, R. (2010). *Poisson Point Processes: Imaging, Tracking, and Sensing*. Springer US.

Streit, R. and Stone, L. (2008). Bayes derivation of multitarget intensity filters. In *11th International Conference on Information Fusion 2008*, pages 1–8.

Tang, X. and Wei, P. (2010). Improved peak extraction algorithm in SMC implementation of PHD filter. In *2010 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*, pages 1–4.

Tobias, M. (2006). *Probability Hypothesis Densities for Multitarget, Multisensor Tracking with Application to Passive Radar*. Ph.D. thesis, School of Electrical and Computer Engineering, Georgia Institute of Technology.

Tobias, M. and Lanterman, A. (2004). A probability hypothesis density-based multitarget tracker using multiple bistatic range and velocity measurements. In *Proceedings of the Thirty-Sixth Southeastern Symposium on System Theory 2004*, pages 205–209.

Tobias, M. and Lanterman, A. (2008). Techniques for birth-particle placement in the probability hypothesis density particle filter applied to passive radar. *Radar, Sonar Navigation, IET*, **2**(5), 351–365.

Vo, B.-N. and Ma, W.-K. (2006). The Gaussian mixture probability hypothesis density filter. *IEEE Transactions on Signal Processing*, **54**(11), 4091–4104.

Vo, B.-N., Singh, S., and Doucet, A. (2005). Sequential Monte Carlo methods for multitarget filtering with random finite sets. *IEEE Transactions on Aerospace and Electronic Systems*, **41**(4), 1224–1245.

Vo, B.-N., Vo, B.-T., Pham, N.-T., and Suter, D. (2010). Joint detection and estimation of multiple objects from image observations. *IEEE Transactions on Signal Processing*, **58**(10), 5129 –5141.

Vo, B.-T., Vo, B.-N., and Cantoni, A. (2009). The cardinality balanced multi-target multi-Bernoulli filter and its implementations. *IEEE Transactions on Signal Processing*, **57**(2), 409–423.

Vo, B.-T., Clark, D., Vo, B.-N., and Ristic, B. (2011). Bernoulli forward-backward smoothing for joint target detection and tracking. *Signal Processing, IEEE Transactions on*, **59**(9), 4473 –4477.

Wang, Y., Jing, Z., and Hu, S. (2008). Data association for PHD filter based on MHT. In *11th International Conference on Information Fusion 2008*, pages 1–8.

Wood, T., Clark, D., and Ristic, B. (2010). Efficient resampling and basic track continuity for the SMC-PHD filter. In *Proc. Cognitive Systems with Interactive Sensors 2010*.

Yin, J. and Zhang, J. (2010). The nonlinear multi-target multi-Bernoulli filter using polynomial interpolation. In *IEEE 10th International Conference on Signal Processing (ICSP), 2010*, pages 2551–2554.

Yin, J., Zhang, J., and Zhao, J. (2010a). The Gaussian particle multi-target multi-Bernoulli filter. In *2nd International Conference on Advanced Computer Control (ICACC) 2010*, volume 4, pages 556–560.

Yin, J., Zhang, J., Hu, B., and Lu, Q. (2010b). The polynomial predictive Gaussian mixture MeMBer filter. In *Sensor Array and Multichannel Signal Processing Workshop (SAM), 2010 IEEE*, pages 233–236.

Yin, J., Zhang, J., and Ni, L. (2010c). The polynomial predictive particle MeMBer filter. In *2nd International Conference on Mechanical and Electronics Engineering (ICMEE) 2010*, volume 1, pages V1–18–V1–22.

Zhu, H., Han, C., and Lin, Y. (2011). Particle labeling PHD filter for multi-target

track-valued estimates. In *14th International Conference on Information Fusion (FUSION) 2011*, pages 1–8.