Retrodiction for Multitarget Tracking

RETRODICTION FOR MULTITARGET TRACKING

By

N. NADARAJAH, B.Sc.Eng., M.A.Sc.

A Thesis

Submitted to the School of Graduate Studies

in Partial Fulfilment of the Requirements

for the Degree of

Doctor of Philosophy

McMaster University

DOCTOR OF PHILOSOPHY (2009) MCMASTER UNIVERSITY

(Electrical and Computer Engineering) Hamilton, Ontario, Canada

TITLE: **Retrodiction for Multitarget Tracking**

AUTHOR: N. Nadarajah

B.Sc.Eng.

University of Peradeniya

Sri Lanka, 2001

M.A.Sc.

McMaster University

Canada, 2005

SUPERVISOR: Dr. T. Kirubarajan

NUMBER OF PAGES: xvii, 144

Abstract

Multi-Target Tracking (MTT), where the number of targets as well as their states are time-varying, concerns with the estimation of both the number of targets and the individual states from noisy sensor measurements, whose origins are unknown. Filtering typically produces the best estimates of the target state based on all measurements up to current estimation time. Smoothing or retrodiction, which uses measurements beyond the current estimation time, provides better estimates of target states. This thesis proposes smoothing methods for various estimation methods that produce delayed, but better, estimates of the target states.

First, we propose a novel smoothing method for the Probability Hypothesis Density (PHD) estimator. The PHD filter, which propagates the first order statistical moment of the multitarget state density, is a computationally efficient MTT algorithm. By evaluating the PHD, the number of targets as well as their individual states can be extracted. Recent Sequential Monte Carlo (SMC) implementations of the PHD filter have paved the way to its application to realistic nonlinear non-Gaussian problems. The proposed PHD smoothing method involves forward multitarget filtering using the standard PHD filter recursion followed by backward smoothing recursion using a novel recursive formula.

Second, we propose a Multiple Model PHD (MMPHD) smoothing method for

tracking of maneuvering targets. Multiple model approaches have been shown to be effective for tracking maneuvering targets. MMPHD filter propagates mode-conditioned PHD recursively. The proposed backward MMPHD smoothing algorithm involves the estimation of a continuous state for target dynamic as well as a discrete state vector for the mode of target dynamics.

Third, we present a smoothing method for the Gaussian Mixture PHD (GMPHD) state estimator using multiple sensors. Under linear Gaussian assumptions, the PHD filter can be implemented using a closed-form recursion, where the PHD is represented by a mixture of Gaussian functions. This can be extended to nonlinear systems by using the Extended Kalman Filter (EKF) or the Unscented Kalman Filter (UKF). In the case of multisensor systems, a sequential update of the PHD has been suggested in the literature. However, this sequential update is susceptible to the imperfections in the last sensor. In this thesis, a parallel update for GMPHD filter is proposed. The resulting filter outputs are further improved using a novel closed-form backward smoothing recursion.

Finally, we propose a novel smoothing method for Kalman based Interacting Multiple Model (IMM) estimator for tracking agile targets. The new method involves forward filtering followed by backward smoothing while maintaining the fundamental spirit of the IMM. The forward filtering is performed using the standard IMM recursion, while the backward smoothing is performed using a novel interacting smoothing recursion. This backward recursion mimics the IMM estimator in the backward direction, where each mode conditioned smoother uses standard Kalman smoothing recursion.

To Appa and Amma

Acknowledgements

I would like to express my gratitude to all those who helped me complete this thesis. First, I wish to express my deepest gratitude to my supervisor Dr. T. Kirubarajan for offering me an opportunity to be a graduate student, and for his guidance and inspiration throughout the entire research for this thesis. His patience with students and willingness to always provide extra help are greatly appreciated. I would also like to thank my supervisory committee members, Dr. S. Haykin and Dr. T. Field, for their feedback on the research in this thesis.

My appreciation also goes to all the instructors in the ECE Department who taught me various courses. What they have taught me in the class has greatly helped me understand the technical papers. Department staff members, especially Cheryl Gies and Helen Jachna, are also gratefully acknowledged for their constant administrative help and support. I thank Terry Greenlay and Cosmin Coriou assisting in resolving computer related issues.

I am thankful to my fellow students in the ETF Laboratory for many thought-provoking and stimulating discussions. All above, I would like to acknowledge the immeasurable contribution of my parents in teaching me the value of education. I am deeply grateful to them, for their tremendous sacrifice a lot to bring my siblings and me to where we are now. Finally, I wish to thank my roommates and friends. Their continuous encouragement has been extraordinary.

Contents

List of Tables

List of Figures

Nomenclature

Acronyms

EAP	Expected *a posteriori*
EKF	Extended Kalman Filter
EM	Expectation Maximization
FISST	Finite Set Statistics
FOV	Field of View
GMPHD	Gaussian Mixture PHD
GMF	Gaussian Mixture Filter
GNN	Global Nearest Neighbor
I.I.D	Independent and Identically Distributed
IMM	Interacting Multiple Model
JPDA	Joint PDA
MAP	Maximum *a posterior*
MHT	Multiple Hypotheses Tracking
MMPHD	Multiple Model PHD
MMSE	Minimum MSE
MSE	Mean Square Error

MTT	Multi-Target Tracking
NEES	Normalized Estimation Error Squared
NN	Nearest Neighbor
PDA	Probabilistic Data Association
PHD	Probability Hypothesis Density
PMHT	Probabilistic MHT
RFS	Random Finite Sets
RTS	Rauch-Tung-Striebel
SMC	Sequential Monte Carlo
SN	Strongest Neighbor
UKF	Unscented Kalman Filter

Mathematical notations

$b_{k\|k-1}$	intensity function of the spawning target
b_i	ith bin
$\|b_i\|$	hyper-volume of the ith bin
$c_k(\cdot)$	spacial density of clutter
$\mathbb{E}[\cdot]$	expectation operator
$f_{k\|k-1}$	nonlinear state transition model
F_k	state transition matrix
h_k	nonlinear measurement model
H_k	measurement matrix
K_k	Kalman gain
$\mathbf{m}_{t\|k}$	mean of Gaussian density at time t given measurement up to time k
n_x	dimension of the state vector $\mathbf{x}_k$
n_z	dimension of the measurement vector $\mathbf{z}_k$
N_k	number of targets at time k
N_p	number of particles
$\hat{N}_{t\|k}$	expected number of targets at time t given measurements up to time k
$\mathcal{N}(\mathbf{x}; \mathbf{m}, P)$	Gaussian distribution with mean $\mathbf{m}$ and covariance P
p	probability density function
p_Ξ	multitarget density function
P_d	probability of detection
P_s	probability of target survival
$P_{t\|k}$	covariance of Gaussian density at time t given measurement up to time k
S_k	innovation covariance
$u(\cdot)$	indicator function

$\mathbf{x}_k$	state vector at time k
$\mathbf{x}_k^{(j)}$	state of the jth particle
$\hat{\mathbf{x}}_{t\|k}$	estimate of target state at time t given measurements up to time k
X	set of target states
$\mathbf{z}_k$	measurement vector at time k
$\mathbf{z}_{1:k}$	measurement sequence from time 1 to k
Z_k	measurement set at time k
$Z_{1:k}$	sequence of measurement sets from time 1 to k
$\gamma_k(\cdot)$	intensity of spontaneous target birth
$\delta(\cdot)$	Dirac delta function
λ_k	average number of clutter return
Ω	sample space
$\Phi_{k\|k-1}$	PHD prediction operator
Ψ_k	PHD update operator
Ξ	random finite set

Chapter 1

Introduction

Multi-Target Tracking (MTT) problem arises in various applications such as surveillance, navigation, control, failure detection, signal processing, medicine, and economics [5][7][8][15][32][92]. In the MTT problem, the objective is to estimate states (e.g., positions and velocities) of targets of interest using noisy observations from one or more sensors. In general, the number of targets in the region of interest may vary over time due to their spontaneous appearance and disappearance. Sensors such as radar and sonar not only report target originated measurements with a probability of detection less than one, but also generate spurious measurements from noise or unwanted sources. Typically, sensors return measurements, whose origins are unknown, in regular intervals. Most tracking algorithms estimate target states recursively using Bayesian principle with Markov assumption on target state evaluation.

1.1 Single Target State Estimation

Consider a clean environment, where a single target evolves and sensor provides only target originated measurement with a probability of detection one. In this simple case

(Figure 1.1), target tracking can be cast as state estimation of a dynamic system. In this formulation, target state is assumed to evolve according to some dynamic model and follow an unobserved Markov process, whereby the target state at a given time is conditionally independent of all the earlier states given the state at the last observation time. It is also assumed that the measurement at a given time is related to the target state at that time by a measurement model and is conditionally independent of all the previous target states given the state at the time of measurement. With these assumptions, the partially observed target states can be estimated using a recursive Bayesian filter.

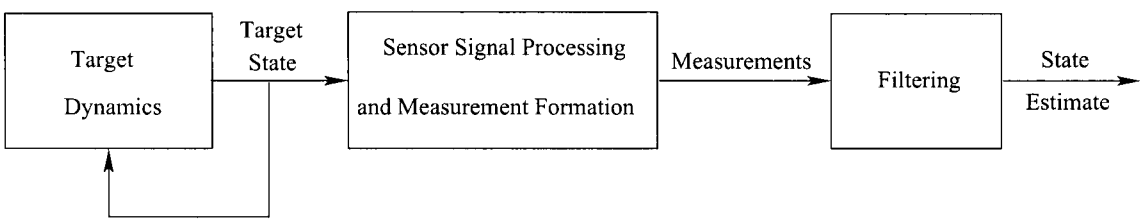


Figure 1.1: Single target tracking

In the case of linear Gaussian problem, where system and measurement models are linear and both process and measurement noises are additive Gaussian, the Kalman filer provides optimal estimate [7]. This Bayesian filter propagates the mean and the covariance of the target state that are sufficient statistics of the posterior for the linear Gaussian problem. Alpha-Beta filter, which is the constant gain Kalman filter, propagates only the mean of the posterior.

For a problem with nonlinear system and/or measurement models, modified versions of the Kalman filter can be used. The Extended Kalman Filter (EKF) performs the Kalman filter recursion with locally linearized models. The Unscented Kalman Filter (UKF) [47], which is a derivative free method, uses unscented transformation to perform Kalman-like recursion. For nonlinear non-Gaussian problems, Sequential

Monte Carlo (SMC) methods (also known as the particle filter) can be used [4][22]. In this method, the target posterior density is represented by a set of random samples.

In maneuvering target tracking problems, where target may change its motion dynamic, the detection and tracking of motion model changes are crucial to get accurate state estimates. For such problems, Multiple Model (MM) approaches have been shown to be highly effective. In these approaches, the target is assumed to evolve according to one of many models from a known set of models throughout the entire process. The optimal MM estimator, which keeps track of exhaustive model histories, has exponentially increasing computational complexity over the time. All MM approaches used in practice are approximate versions of the optimal one. The Interacting Multiple Model (IMM) method cleverly keeps track of model paths at any time by making a soft decision and by maintaining interaction between them [7][11].

1.2 MTT Algorithms

In contrast, multitarget tracking (Figure 1.2) is more challenging due to the time-varying number of targets and measurement origin uncertainties [6][8]. The number of targets varies over time as targets in the area of interest continually appear and disappear. Sensors report both target-originated measurements with a probability of detection less than one and a set of spurious measurements not originating from any of the targets of interest. In such a situation, an MTT algorithm must jointly estimate the number of targets and their states using the collection of measurements reported by the sensor at each time step. One way of solving this problem is to find explicit associations between measurements and targets, and then to filter associated measurements for individual target states [6][8][56]. The Nearest Neighbor (NN) and

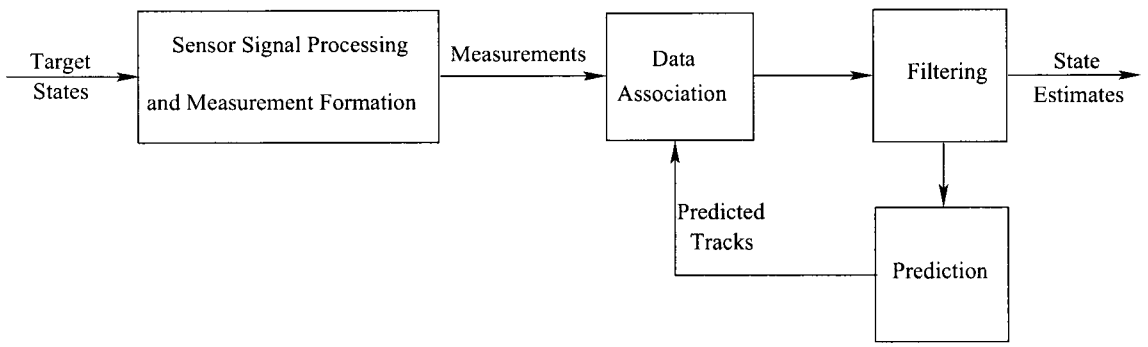


Figure 1.2: Conventional multitarget tracking

Strongest Neighbor (SN) use the nearest and strongest measurement as associated measurement, respectively [5]. The Multiple Hypotheses Tracking (MHT) algorithm builds all possible association hypotheses for the sequence of measurements [8][9][80]. In multiframe assignment, which is a sliding window version of the MHT, the MTT problem is solved using an optimization formulation [67]. The Joint Probabilistic Data Association (JPDA) filter [6], and the Probabilistic MHT (PMHT) algorithm [93] consider soft association by weighting the measurements probabilistically. These explicit association methods require combinatorial enumeration of targets and measurements that results in a huge computational load [63][98].

On the other hand, the multitarget problem can be modelled using Random Finite Sets (RFS) [32][66]. This association-free approach considers the collection of target states as a single meta-target state and the collection of observations as a single meta-observation. Finite Set Statistics (FISST) offers a systematic foundation for multitarget tracking based on the theory of RFS [61][62]. It leads to an elegant multitarget generalization of the single-target Bayes filter [63][64][65][66]. The optimal multitarget Bayes filter based on RFS, which involves set integration, is usually computationally expensive and intractable [96].

1.2.1 Probability Hypothesis Density Filter

The Probability Hypothesis Density (PHD) filter, which is the first-order moment approximation of the optimal multitarget Bayes filter based on RFS, is a computationally tractable alternative. This recursive filter assumes that the predicted multitarget state density is Poisson for which the PHD (the first order statistical moment) completely characterizes the underline dynamic Poisson point process. The PHD is a positive function defined on single target state space. The integral of the PHD in any region is equal to the expected number of targets in that region. By evaluating the PHD recursively, the number of targets as well as their individual states can be extracted [66][63].

Though the PHD recursion consists of equations that are considerably simpler than those of the optimal multitarget Bayes filter, it still requires solving multidimensional integrals that do not have closed-form solutions in general. SMC implementations of the PHD filter, where the PHD is represented by set of random samples particles, have paved the way to its application to realistic nonlinear non-Gaussian problems [88][95][96]. For maneuvering target tracking, Multiple Model PHD (MM-PHD) filter, which propagates mode dependent intensity recursively, was proposed [76][77].

Recently, the Gaussian mixture PHD (GMPHD) filter [96][98], where the PHD surface is represented by a mixture of Gaussian intensity functions, was proposed for the linear-Gaussian MTT problem with a Gaussian mixture model for target birth. Similar to the Gaussian sum filter [1][91], GMPHD filter propagates means, covariances, and weights of the constituent Gaussian components of the posterior intensity surface through two steps: namely, prediction and update. This closed-form recursion can be extended to mildly nonlinear problems using the EKF or the UKF.

The convergence property of the SMC based PHD filter has been established in [46]. The convergence analysis of GMPHD filter is provided in [18]. However, it is indicated in [26] that the PHD filter is dependent on current measurements, especially in the case of low observable target problems (i.e., estimates are sensitive to missed detections and false alarms) that is the motivation behind introducing smoothing method for PHD based estimator in this thesis.

1.3 Smoothing

Using filtering algorithms, one can typically achieve the best estimates of the target states at a given time based on all measurements up to the current estimation time. This best estimate at a given time can be improved significantly by smoothing or retrodiction, which uses more measurements beyond the current estimation time [38]. If a certain time delay can be tolerated, accurate estimates can be obtained using smoothing, which requires some additional computational load to incorporate information to the current state from measurements beyond current estimation time.

The Kalman based smoothing algorithms can be found in [7]. Smoothing has been incorporated into a number of different tracking algorithms. Probabilistic Data Association (PDA) smoothing algorithm is proposed in [59] to improve the tracking performance in a clutter environment. Maneuvering target tracking is improved by IMM smoothing method in [40]. IMM-PDA smoothing is reported in [17] to improve the tracking of agile targets in clutter environment. Improved multitarget tracking is demonstrated using IMM-MHT smoothing in [54].

Similarly, particle based smoothing algorithms are proposed to improve nonlinear non-Gaussian tracking problems. In [51], smoothing is performed by using a two-filter

formula, where the smoothed distribution is obtained by combining the results of forward and backward filtering algorithms. In [28], a block-based particle smoothing method was proposed. In [42][21][31], particle smoothing is performed using forward filtering followed by backward smoothing.

1.4 Contributions

The main focus of this thesis is to develop smoothing method for various state estimation algorithms. All methods, which are Rauch-Tung-Striebel (RTS) type smoothers [30][79], involve forward filtering using existing standard algorithms followed by novel backward smoothing algorithms. These methods provide delayed, but better, estimates for target states that requires additional computational effort. The following sections outline the main contributions of the thesis.

1.4.1 PHD Smoothing

In this work, we propose a PHD smoothing algorithm to improve the capability of PHD based state estimator. It involves a forward multitarget filtering using the standard PHD filter recursion and then a backward smoothing recursion. This backward smoothing recursion is performed with a novel recursive formula, which is derived using the physical-space approach presented in [27]. The resulting backward recursion incorporates intensity for surviving targets as well as disappearing targets. Compared to optimal multitarget Bayesian smoothing, this first order recursion is simple and evolves in single target state space. However, it does not yield any closed-form recursion. To mitigate this problem, we propose an SMC implementation of the smoothing method. We show that this SMC implementation requires much more computational

effort to compute smoothed particle weights. To alleviate this, we introduce a fast implementation [52], which uses the N-body algorithm from [34]. This fast method requires that target transition density be defined in metric space. However, we show that most commonly used transition models can be converted into a function defined in a metric space using simple transformations.

1.4.2 Multiple Model PHD Smoothing

MMPHD filter is a natural extension of the PHD filter for maneuvering target tracking. In this work, we propose a smoothing algorithm for MMPHD based state estimator. Here, the mode dependent PHD is propagated recursively. SMC implementation of the backward recursion involves continuous density as well as probability mass function due to discrete variable for mode. This discrete density does not permit the use of the fast method discussed in the previous section. However, we demonstrate that the fast method can still be used for a special case where targets switch between two dynamic models and have symmetric mode transition matrix.

1.4.3 Gaussian Mixture PHD Smoothing

Next, we propose a smoothing method for the GMPHD state estimator using multiple sensors. Under linear Gaussian assumption, the PHD filter can be implemented using closed-form recursion, where the PHD is represented by a mixture of Gaussian functions. This can be extended to nonlinear systems by using the EKF or the UKF. In the case of multisensor systems, a sequential update of the PHD has been suggested in the literature. However, this sequential update is susceptible to the imperfections in the last sensor. In this thesis, a parallel update for GMPHD filter is proposed. The resulting filter outputs are further improved using a novel closed-form backward

smoothing recursion.

1.4.4 IMM Smoothing

Finally, we propose a novel smoothing method for Kalman filter based IMM estimators for tracking agile targets. The new method involves forward filtering followed by backward smoothing while maintaining the fundamental spirit of the IMM estimator. The forward filtering is performed using the standard IMM recursion, while the backward smoothing is performed using a novel interacting smoothing recursion. This backward recursion mimics the IMM estimator in the backward direction, where each mode conditioned smoother uses the standard Kalman smoothing recursion.

1.5 Organization of the Thesis

This thesis is structured as follows: Chapter 2 focuses on a brief review of filtering algorithms including single target filtering and multitarget filtering. Chapter 3 outlines the PHD filter and the derivation of its recursion using physical space approach. Chapter 4 provides an overview of smoothing algorithms. In Chapter 5, the backward PHD smoothing algorithm is derived using the physical-space approach and implemented using SMC method. Chapter 6 provides MMPHD smoothing algorithm. It also briefly discusses an efficient implementation of this backward smoothing. In Chapter 7, GMPHD smoothing is developed. This closed-form recursion is applied to multisensor systems. Chapter 8 proposes a new smoothing algorithm for Kalman filter IMM estimator. Finally, Chapter 9 concludes the findings.

Chapter 2

Filtering

In target tracking, filtering plays crucial role whereby we estimate target state from noisy observations. Target state, which includes position and velocity of the target, is partially observed by one or more sensors. The objective of filtering algorithms is to produce optimal estimate of the target state based on some criteria such as Minimum Mean Square Error (MMSE). This chapter reviews filtering methods, which are part of smoothing methods (i.e., forward filtering and backward smoothing) developed in this thesis. Section 2.1 reviews single target filtering methods, while multitarget filtering methods are outlined in Section 2.2.

2.1 Single Target Filtering

In a clean environment, where a single target evolves and sensor provides only target originated measurement with the probability of detection one, the state estimation problem can be defined as follows: state sequence $\mathbf{x}_k \in \mathbb{R}^{n_x}$ at time steps $k \in \mathbb{N}$ evolve according to some state transition model, where n_x is the dimension of target

state. That is,

$$\mathbf{x}_k = f_{k|k-1}\left(\mathbf{x}_{k-1}, \mathbf{v}_{k-1}\right) \tag{2.1}$$

where $f_{k|k-1} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_v} \rightarrow \mathbb{R}^{n_x}$ is a possibly nonlinear function of the previous state $\mathbf{x}_{k-1}$ and an I.I.D process noise sequence $\mathbf{v}_{k-1} \in \mathbb{R}^{n_v}$ with dimension n_v. Sensor observation process is modeled as follows:

$$\mathbf{z}_k = h_k\left(\mathbf{x}_k, \mathbf{n}_k\right) \tag{2.2}$$

where $h_k : \mathbb{R}^{n_x} \times \mathbb{R}^{n_n} \rightarrow \mathbb{R}^{n_z}$ is a possibly nonlinear function of the current state $\mathbf{x}_k$ and an I.I.D measurement noise sequence $\mathbf{n}_k \in \mathbb{R}^{n_n}$.

The objective of the filtering algorithms is to find estimate $\widehat{\mathbf{x}}_k$ for state $\mathbf{x}_k$ using measurement sequence $\mathbf{z}_{1:k} = \{\mathbf{z}_j, j = 1, \ldots, k\}$ up to time k and given the above models. For this dynamic estimation problem, the recursive Bayesian approach provides an optimal solution [81]. In the Bayesian filter, posterior Probability Density Function (PDF) $p_{k|k}\left(\mathbf{x}_k|\mathbf{z}_{1:k}\right)$ of target state $\mathbf{x}_k$ is recursively constructed starting from initial PDF $p_0(\mathbf{x}_0)$.

2.1.1 The Bayesian Filter

In Bayes filter, the recursive estimation of the state is performed with following assumptions:

- Target state is conditionally independent of all earlier states given the immediate previous state. That is,

$$p\left(\mathbf{x}_k|\mathbf{x}_{k-1}, \ldots, \mathbf{x}_0\right) = p\left(\mathbf{x}_k|\mathbf{x}_{k-1}\right) \tag{2.3}$$

- The measurement at time k is dependent only on the state $\mathbf{x}_k$ and conditionally independent of all other states. That is,

$$p\left(\mathbf{z}_k|\mathbf{x}_k,\ldots,\mathbf{x}_0\right)=p\left(\mathbf{z}_k|\mathbf{x}_k\right) \tag{2.4}$$

With these assumptions, the posterior PDF $p_{k|k}\left(\mathbf{x}_k|\mathbf{z}_{1:k}\right)$ at time t can be found by following two-step recursion.

- Prediction – Given by Chapman-Kolmogorov equation as follows:

$$p_{k|k-1}\left(\mathbf{x}_k|\mathbf{z}_{1:k-1}\right)=\int p_{k|k-1}\left(\mathbf{x}_k|\mathbf{x}_{k-1}\right)p_{k-1|k-1}\left(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1}\right)d\mathbf{x}_{k-1} \tag{2.5}$$

- Update – Given by Bayes rule as follows:

$$p_{k|k}\left(\mathbf{x}_k|\mathbf{z}_{1:k}\right)=\frac{p_k\left(\mathbf{z}_k|\mathbf{x}_k\right)p_{k|k-1}\left(\mathbf{x}_k|\mathbf{z}_{1:k-1}\right)}{p_k\left(\mathbf{z}_k|\mathbf{z}_{1:k-1}\right)} \tag{2.6}$$

where the normalizing constant

$$p_k\left(\mathbf{z}_k|\mathbf{z}_{1:k-1}\right)=\int p_k\left(\mathbf{z}_k|\mathbf{x}_k\right)p_{k|k-1}\left(\mathbf{x}_k|\mathbf{z}_{1:k-1}\right)d\mathbf{x}_k \tag{2.7}$$

According to Bayesian inference, the posterior PDF $p_{k|k}\left(\mathbf{x}_k|\mathbf{z}_{1:k}\right)$ contains all information of the target state and completely characterizes all statistical uncertainty in the target state. Hence, given the posterior density, the estimate of the state can be obtained under certain criterion such that it minimizes an objective function called the Bayes risk [82]. The most common estimators are the Maximum *a posterior* (MAP)

and the Expected *a posteriori* (EAP). They are defined as

$$\widehat{\mathbf{x}}_k^{MAP} = \arg\sup_{\mathbf{x}_k} p_{k|k}\left(\mathbf{x}_k|\mathbf{z}_{1:k}\right) \tag{2.8}$$

$$\widehat{\mathbf{x}}_k^{EAP} = \int \mathbf{x}_k p_{k|k}\left(\mathbf{x}_k|\mathbf{z}_{1:k}\right) d\mathbf{x}_k \tag{2.9}$$

The EAP estimate minimizes the Mean Square Error (MSE) of the estimate, while the MAP corresponds to maxima of posterior PDF. Due to the multi-dimensional integration in (2.5) and (2.6), in general, the Bayes filter cannot be realized for most practical problems. There are several approaches that approximate the Bayes recursion. In the following sections, an outline of the closed-form recursion for a special case and most common approximate approaches are provided.

2.1.2 Kalman Filter

The Kalman filter [48] provides a closed-form recursion for a special case, where system and measurement models, (2.1) and (2.2), are linear Gaussian as follows:

$$\mathbf{x}_k = F_k\mathbf{x}_{k-1} + \mathbf{v}_{k-1} \tag{2.10}$$

$$\mathbf{z}_k = H_k\mathbf{x}_k + \mathbf{n}_k \tag{2.11}$$

where F_k is the state transition matrix, H_k is the measurement matrix, and $\mathbf{v}_{k-1}$ and $\mathbf{n}_k$ are Gaussian noises with covariance matrices Q_{k-1} and R_k. It is also assumed that the initial density is Gaussian. That is,

$$p_0(\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_0; \mathbf{m}_0, P_0) \tag{2.12}$$

where $\mathcal{N}(\mathbf{x}; \mathbf{m}, P)$ is a Gaussian distribution characterized by its mean, $\mathbf{m}$, and covariance, P. The Kalman filter works because, if $p_{k-1|k-1}(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})$ and $p_{k|k-1}(\mathbf{x}_k|\mathbf{x}_{k-1})$ and $p_{k|k}(\mathbf{z}_k|\mathbf{x}_k)$ are all Gaussian distributions, then so is $p_{k|k}(\mathbf{x}_k|\mathbf{z}_{1:k})$ [35][78]. That is, if

$$p_{k-1|k-1}(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1}) = \mathcal{N}(\mathbf{x}_{k-1}; \mathbf{m}_{k-1|k-1}, P_{k-1|k-1}) \tag{2.13}$$

$$p_{k|k-1}(\mathbf{x}_k|\mathbf{x}_{k-1}) = \mathcal{N}(\mathbf{x}_k; F_k\mathbf{x}_{k-1}, Q_{k-1}) \tag{2.14}$$

$$p_{k|k}(\mathbf{z}_k|\mathbf{x}_k) = \mathcal{N}(\mathbf{z}_k; H_k\mathbf{x}_k, R_k) \tag{2.15}$$

then the Kalman filter recursion is given as follows:

- Prediction

$$p_{k|k-1}(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = \mathcal{N}(\mathbf{x}_k; \mathbf{m}_{k|k-1}, P_{k|k-1}) \tag{2.16}$$

- Update

$$p_{k|k}(\mathbf{x}_k|\mathbf{z}_{1:k}) = \mathcal{N}(\mathbf{x}_k; \mathbf{m}_{k|k}, P_{k|k}) \tag{2.17}$$

with the parameters given by

$$\mathbf{m}_{k|k-1} = F_k\mathbf{m}_{k-1|k-1} \tag{2.18}$$

$$P_{k|k-1} = F_kP_{k-1|k-1}F_k^T + Q_{k-1} \tag{2.19}$$

$$\begin{aligned}
\mathbf{m}_{k|k} &= \mathbf{m}_{k|k-1} + K_k(\mathbf{z}_k - H_k\mathbf{m}_{k|k-1}) \\
P_{k|k} &= (I - K_k H_k)P_{k|k-1} \\
K_k &= P_{k|k-1}H_k^T S_k^{-1} \\
S_k &= H_k P_{k|k-1} H_k^T
\end{aligned}$$

The matrix K_k is referred to as the Kalman gain, the residual $\mathbf{z}_k - H_k\mathbf{m}_{k|k-1}$ is referred to as the innovation and the matrix S_k is the innovation covariance. An interesting point to note about the recursion formed by (2.16) and (2.17) is that there is no effect of the measurement on the covariances of the normal distributions. The covariance is defined entirely by the matrices that comprise the system, F_k, H_k, Q_{k-1}, R_k, P_0. If the system matrices are known and constant over time, then the covariances can be precalculated. It is also worth noting that in such cases, the covariance will tend to a value. This means that, after some initial period, the uncertainty is constant. This steady state is the result of a balance between the increase in uncertainty as a result of the prediction step and the reduction in uncertainty due to the update step. This observation leads to the $\alpha - \beta$ filter, which can be viewed as a special case of the Kalman filter.

2.1.3 Interacting Multiple Model Estimator

Since the Kalman filter assumes a fixed model for the state evolution, its performance is not satisfactory in terms of the estimation error when it is used to estimate the state of systems whose model vary with time [49]. In such a scenario, Multiple Model (MM) estimators perform better than a Kalman filter. In the MM approach, it is assumed that the system obeys one of a finite number of models and a Bayesian framework is used: starting with prior probabilities of each model being correct, (i.e.,

system is in particular mode), the corresponding posterior probabilities are obtained.

A system that can undergo model switching in time can be described by the following equations:

$$\mathbf{x}_k = F_k(r_k)\mathbf{x}_{k-1} + \mathbf{v}_{k-1}(r_k) \tag{2.20}$$

$$\mathbf{z}_k = H_k(r_k)\mathbf{x}_k + \mathbf{n}_k(r_k) \tag{2.21}$$

where r_k denotes the mode or model at time k. The mode at time k is assumed to be among the possible N_r modes

$$r_k \in \{r^j\}_{j=1}^{N_r} \tag{2.22}$$

The event that model j is in effect at time k is denoted as

$$r_k^j = \{r_k = r^j\} \tag{2.23}$$

It will be assumed that the mode switching is a Markov process (Markov chain) with known mode transition probabilities

$$p_{ij} = P\{r_k^j | r_{k-1}^i\} \tag{2.24}$$

These mode transition probabilities will be assumed time-invariant and independent of the base state. The lth mode history – or sequence of models – through time k is denoted as

$$r_{k,l} = \{r_1^{i_{1,l}}, \ldots, r_k^{i_{k,l}}\} \tag{2.25}$$

where $i_{\kappa,l}$ is the model index at time κ from history l and

$$1 \leq i_{\kappa,l} \leq N_r \qquad \kappa = 1, \ldots, k \tag{2.26}$$

The conditional PDF of the state at time k is obtained using the total probability theorem with respect to the mutually exclusive and exhaustive set of events (2.25), as a Gaussian mixture with an exponentially increasing number of terms

$$p_{k|k}(\mathbf{x}_k|\mathbf{z}_{1:k}) = \sum_{l=1}^{N_r^k} p_{k|k}(\mathbf{x}_k|r_{k,l}, \mathbf{z}_{1:k}) P\{r_{k,l}|\mathbf{z}_{1:k}\} \tag{2.27}$$

Since to each mode sequence one has to match a filter, it can be seen that an exponentially increasing number of filters are needed to estimate the (base) state, which makes the optimal approach impractical. All MM approaches used in practice are approximate versions of the optimal one. The Interacting Multiple Model (IMM) method cleverly keeps track of model paths at any time by making soft decision and having interaction between them [10][11]. The IMM recursion at time k starts with the Gaussian mixture, $\mathcal{M}_{k-1|k-1} = \{\mathcal{M}^i_{k-1|k-1}\}_{i=1}^{N_r}$, which represents the posterior density at time $k-1$. Here, $\mathcal{M}^i_{k-1|k-1}$ denotes the ith mode density, which is a Gaussian component with mode probability $\mu^i_{k-1|k-1}$, mean $\mathbf{m}^i_{k-1|k-1}$, and covariance matrix $P^i_{k-1|k-1}$. The IMM recursion can be summarized as follows [7]:

- Mixing probability calculation: The mixing probability is given by

$$\mu^{i|j}_{k-1|k-1} = \frac{1}{\bar{c}_j} p_{ij} \mu^i_{k-1|k-1} \quad i, j = 1, \ldots, N_r \tag{2.28}$$

where

$$\bar{c}_j = \sum_{i=1}^{N_r} p_{ij}\mu^i_{k-1|k-1} \quad j = 1, \ldots, N_r. \tag{2.29}$$

- Mixing: The mean and the covariance matrix for the jth mode-matched filter are given by

$$\mathbf{m}^{0j}_{k-1|k-1} = \sum_{i=1}^{N_r} \mu^{i|j}_{k-1|k-1}\mathbf{m}^i_{k-1|k-1} \tag{2.30}$$

$$\begin{aligned} P^{0j}_{k-1|k-1} = & \sum_{i=1}^{N_r} \mu^{i|j}_{k-1|k-1}\left\{P^i_{k-1|k-1} + \left[\mathbf{m}^i_{k-1|k-1} - \mathbf{m}^{0j}_{k-1|k-1}\right]\right. \\ & \left. \times \left[\mathbf{m}^i_{k-1|k-1} - \mathbf{m}^{0j}_{k-1|k-1}\right]^T\right\} \end{aligned} \tag{2.31}$$

- Mode-matched filtering: The mean and the covariance in (2.30) and (2.31) are used as input to the mode-matched filter with model r^j_k using measurement $\mathbf{z}_k$. The mode likelihood is given by

$$\Lambda^j_k = p(\mathbf{z}_k|r^j_k, \mathbf{Z}^k_1) = \mathcal{N}\left(\mathbf{z}_k; \mathbf{z}^j_{k|k-1}(\mathbf{m}^{0j}_{k-1|k-1}), S^j_k(P^{0j}_{k-1|k-1})\right) \tag{2.32}$$

where $\mathbf{z}^j_{k|k-1}(\mathbf{m}^{0j}_{k-1|k-1})$ and $S^j_k(P^{0j}_{k-1|k-1})$ are the predicted measurement and the innovation covariance for mode filter r^j_k, respectively.

- Mode probability update: The updated mode probability is given by

$$\mu^j_{k|k} = \frac{1}{c}\Lambda^j_k\bar{c}_j \tag{2.33}$$

where the normalizing constant $c = \sum_{j=1}^{N_r}\Lambda^j_k\bar{c}_j$.

- Estimate: Finally the estimate and corresponding covariance matrix are found using moment matching. That is,

$$\hat{\mathbf{x}}_{k|k} = \sum_{i=1}^{N_r} \mu_{k|k}^{j} \mathbf{m}_{k|k}^{j} \tag{2.34}$$

$$P_{k|k} = \sum_{i=1}^{N_r} \mu_{k|k}^{j} \left\{ P_{k|k}^{j} + \left[\mathbf{m}_{k|k}^{j} - \hat{\mathbf{x}}_{k|k} \right] \left[\mathbf{m}_{k|k}^{j} - \hat{\mathbf{x}}_{k|k} \right]^{T} \right\} \tag{2.35}$$

The algorithm is also summarized in Figure 2.1.

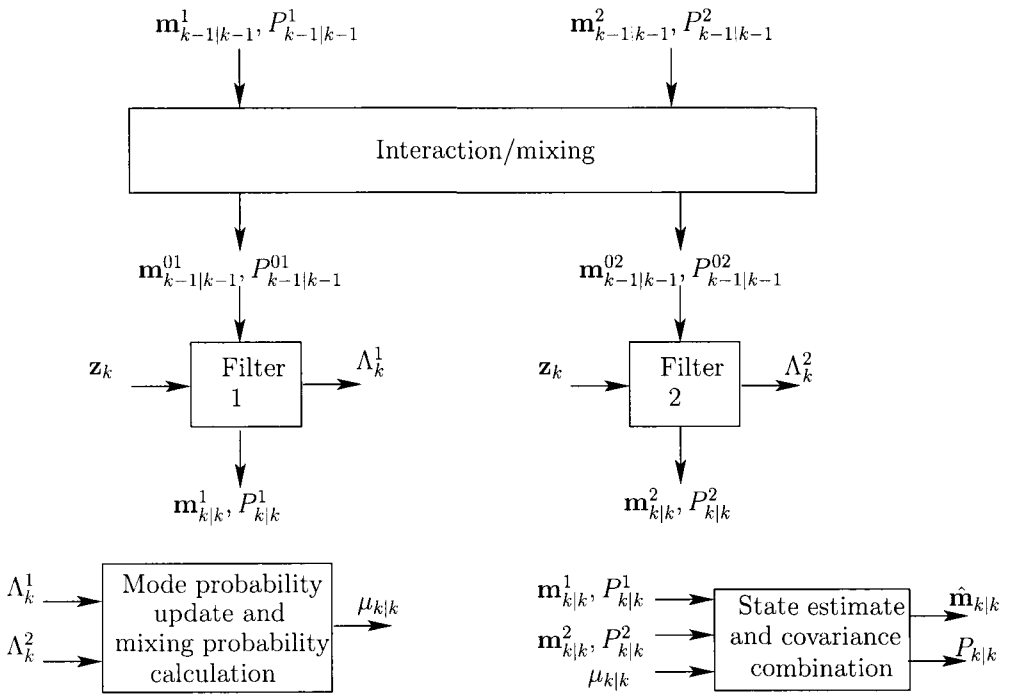


Figure 2.1: IMM filter with two models

2.1.4 Extended Kalman Filter

In many situations of interest, linear Gaussian assumption of the Kalman filter does not hold. It is then necessary to make approximation. The Extended Kalman Filter

(EKF) relaxes linear assumption such that new models are given by

$$\mathbf{x}_k = f_{k|k-1}(\mathbf{x}_{k-1}) + \mathbf{v}_{k-1} \tag{2.36}$$

$$\mathbf{z}_k = h_k(\mathbf{x}_k) + \mathbf{n}_k \tag{2.37}$$

where functions $f_{k|k-1}(\cdot)$ and $h_k(\cdot)$ are nonlinear, $\mathbf{v}_{k-1}$ and $\mathbf{n}_k$ are still zero mean Gaussian and statically independent. The EKF linearizes the models about the current mean and the covariance using Taylor series approximation and then apply the standard Kalman filter equations [2][7]. The local linearization is as follows:

$$\tilde{F}_k = \left.\frac{df_{k|k-1}(\mathbf{x}_{k-1})}{d\mathbf{x}_{k-1}}\right|_{\mathbf{x}_{k-1}=\mathbf{m}_{k-1|k-1}} \tag{2.38}$$

$$\tilde{H}_k = \left.\frac{dh_k(\mathbf{x}_k)}{d\mathbf{x}_k}\right|_{\mathbf{x}_k=\mathbf{m}_{k|k-1}} \tag{2.39}$$

Here, the system and the measurement matrices F_k and H_k in the Kalman recursion (2.16) and (2.17) are replaced with above Jacobian matrices $\tilde{F}_k$ and $\tilde{H}_k$, respectively. Such a local approximation of the models may be a sufficient description of the nonlinearity. However, it may fail to describe highly nonlinear system and measurement models. A higher order EKF that retains further terms in the Taylor expansions exists and results in a closer approximation to the true posterior. The additional complexity has prohibited its widespread use.

2.1.5 Unscented Kalman Filter

The Unscented Kalman Filter (UKF) [47][100] is another approximate approach to the Kalman filter based on the unscented transform. This approach considers a

set of points that are deterministically sampled from Gaussian approximations for $p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})$. These points are all propagated through the true nonlinearity and the parameters of the Gaussian approximation are re-estimated as follows. If the posterior density at time $k-1$ is Gaussian

$$p_{k-1|k-1}(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1}) = \mathcal{N}(\mathbf{x}_{k-1}; \mathbf{m}_{k-1|k-1}, P_{k-1|k-1}) \tag{2.40}$$

start with the augment mean and covariance

$$\begin{aligned} \bar{\mathbf{y}}_k &= [\mathbf{m}_{k-1|k-1}\ \mathbf{0}^T\ \mathbf{0}^T]^T \\ C_k &= \text{diag}(P_{k-1|k-1}, Q_{k-1}, R_k) \end{aligned} \tag{2.41}$$

and generate a set of $L_y = 2n_y + 1$ sigma points $\{\mathbf{y}_k^{(l)}\}_{l=0}^{L_y}$ and weights $\{w_k^{(l)}\}_{l=0}^{L_y}$ according to

$$\begin{aligned} \mathbf{y}_k^{(l)} &= \bar{\mathbf{y}}_k & w_k^{(l)} &= \kappa/(n_y+\kappa) & l &= 0 \\ \mathbf{y}_k^{(l)} &= \bar{\mathbf{y}}_k + \left(\sqrt{(n_y+\kappa)C_k}\right)_l & w_k^{(l)} &= 1/2(n_y+\kappa) = & l &= 1,\ldots,n_y \\ \mathbf{y}_k^{(n_y+l)} &= \bar{\mathbf{y}}_k - \left(\sqrt{(n_y+\kappa)C_k}\right)_l & w_k^{(n_y+l)} &= 1/2(n_y+\kappa) & l &= 1,\ldots,n_y \end{aligned}$$

where $\kappa \in \mathbb{R}$ and $\left(\sqrt{(n_y+\kappa)C_k}\right)_l$ denotes the lth row of the square root of matrix $(n_y+\kappa)C_k$. Each sigma point consists of following partitions

$$\mathbf{y}_k^{(l)} = \left[\left(\mathbf{x}_{k-1}^{(l)}\right)^T \left(\mathbf{v}_{k-1}^{(l)}\right)^T \left(\mathbf{n}_k^{(l)}\right)^T\right]^T \tag{2.42}$$

For the prediction, the sigma points are propagated through the transition function according to $\mathbf{x}_{k|k-1}^{(l)} = f_{k|k-1}\left(\mathbf{x}_{k-1}^{(l)}, \mathbf{v}_{k-1}^{(l)}\right)$ for $l = 0,\ldots,L_y$, and the predicted density

at time k is approximated as the Gaussian

$$p_{k|k-1}(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = \mathcal{N}(\mathbf{x}_k; \mathbf{m}_{k|k-1}, P_{k|k-1}) \tag{2.43}$$

where

$$\begin{aligned}
\mathbf{m}_{k|k-1} &= \sum_{l=0}^{L_y} w_k^{(l)} \mathbf{x}_{k|k-1}^{(l)} \\
P_{k|k-1} &= \sum_{l=0}^{L_y} w_k^{(l)} \left(\mathbf{x}_{k|k-1}^{(l)} - \mathbf{m}_{k|k-1}\right) \left(\mathbf{x}_{k|k-1}^{(l)} - \mathbf{m}_{k|k-1}\right)^T
\end{aligned}$$

For the update, the sigma points are propagated through the measurement model according to $\mathbf{z}_{k|k-1}^{(l)} = h_k\left(\mathbf{x}_{k|k-1}^{(l)}, \mathbf{n}_k^{(l)}\right)$ for $l = 0, \ldots, L_y$, and the updated density at time k is approximated as Gaussian

$$p_{k|k}(\mathbf{x}_k|\mathbf{z}_{1:k}) = \mathcal{N}(\mathbf{x}_k; \mathbf{m}_{k|k}, P_{k|k}) \tag{2.44}$$

where

$$\begin{aligned}
\mathbf{m}_{k|k} &= \mathbf{m}_{k|k-1} + K_k(\mathbf{z}_k - \mathbf{z}_{k|k-1}) \\
P_{k|k} &= P_{k|k-1} - P_{xz} S_k^{-1} P_{xz}^T \\
\mathbf{z}_{k|k-1} &= \sum_{l=0}^{L_y} w_k^{(l)} \mathbf{z}_{k|k-1}^{(l)} \\
K_k &= P_{xz} S_k^{-1} \\
S_k &= \sum_{l=0}^{L_y} w_k^{(l)} \left(\mathbf{z}_{k|k-1}^{(l)} - \mathbf{z}_{k|k-1}\right) \left(\mathbf{z}_{k|k-1}^{(l)} - \mathbf{z}_{k|k-1}\right)^T \\
P_{xz} &= \sum_{l=0}^{L_y} w_k^{(l)} \left(\mathbf{x}_{k|k-1}^{(l)} - \mathbf{m}_{k|k-1}\right) \left(\mathbf{z}_{k|k-1}^{(l)} - \mathbf{z}_{k|k-1}\right)^T
\end{aligned}$$

UKF is equivalent to a higher order EKF and its algorithmic simplicity makes it more attractive than the higher order EKF. If the true distribution of non-Gaussian then a Gaussian, however good approximation, can never describe the distribution well. It is possible to approximate the state space as consisting of a grid of points and then use an approximate grid based approach. This is frequently the approach taken by the speech processing research community. In such cases, approximate grid based filters yield an improvement in performance in comparison to that of the EKF [3]. The Gaussian Mixture Filter (GMF) has been proposed in [1] to handle non-Gaussian distribution. GMF works by approximating the non-Gaussian distribution with a mixture of Gaussian terms. However it requires linear approximations to the system and measurement models. The GMF results in the exponential growth in the number of mixture components.

2.1.6 Particle Filter

Recently, a large number of filters based on Sequential Monte Carlo (SMC) approximations to the Bayes filter, called particle filters, were proposed [4][22][81], which require no linear or Gaussian assumptions on the dynamical models. Rather than approximating the models in order to be able to fit a distribution of a given type to the posterior, a particle filter explicitly approximates the distribution so that it can handle highly nonlinear non-Gaussian models. The approach has also been known as bootstrap filter [33], condensation algorithm [58] and SMC filtering [22].

In particle filtering, the required posterior density function is represented by a set of random samples ('particles') with associated weights [4][33]. Let $\{\mathbf{x}_{k-1}^{(j)}\}_{j=1}^{N_p}$, with associated weights $\{w_{k-1|k-1}^{(j)}\}_{j=1}^{N_p}$ be the random samples representing the posterior density $p_{k-1|k-1}(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})$. Here, N_p is the number of particles. The weights are

normalized such that $\sum_{j=1}^{N_p} w_{k-1|k-1}^{(j)} = 1$. There are many variants in particle filter. The simplest and widely implemented variant is Sampling Importance Resampling (SIR) particle filter.

2.1.6.1 SIR Particle Filter

Sampling Importance Resampling (SIR)[83][90] propagates and updates the particles in which the $p_{k-1|k-1}(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})$ is represented by equally weighted particles. Then

$$p_{k-1|k-1}(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1}) \approx \frac{1}{N_p} \sum_{j=1}^{N_p} \delta(\mathbf{x}_{k-1} - \mathbf{x}_{k-1}^{(j)}) \tag{2.45}$$

where $\delta(\cdot)$ is the Dirac Delta function. The prediction and update of the particles in SIR method are given as follows:

Prediction: Take each existing particle, $\mathbf{x}_{k-1}^{(j)}$ and sample from Importance density ($\mathbf{x}_k^{(*j)} \sim f_{k|k-1}\left(\mathbf{x}_{k-1}^{(j)}, \mathbf{v}_{k-1}^{(j)}\right)$), using the system model, where $\mathbf{v}_{k-1}^{(j)}$ is a random sample from the distribution of the process noise $\mathbf{v}_{k-1}$. The set $\{\mathbf{x}_k^{(*j)}\}_{j=1}^{N_p}$ gives an approximation of the prior, $p_{k|k-1}(\mathbf{x}_k|\mathbf{z}_{1:k-1})$, at time k, i.e.,

$$p_{k|k-1}(\mathbf{x}_k|\mathbf{z}_{1:k-1}) \approx \frac{1}{N_P} \sum_{j=1}^{N_P} \delta(\mathbf{x}_k - \mathbf{x}_k^{(*j)}) \tag{2.46}$$

Update: At each measurement epoch, to account for the fact that the samples, $\mathbf{x}_k^{(*j)}$ are not drawn from $p_{k|k}(\mathbf{x}_k|\mathbf{z}_{1:k})$, the weights are modified using the principle of Importance sampling. When using the prior as the Importance density, it can be shown that the weights are given by

$$w_{k|k}^{(*j)} \propto p_k(\mathbf{z}_k|\mathbf{x}_k = \mathbf{x}_k^{(*j)}, \mathbf{z}_{1:k-1}) \tag{2.47}$$

Reselection: Resample (with replacement) from $\{\mathbf{x}_k^{(*j)}\}_{j=1}^{N_p}$, using the weights, $\{w_{k|k}^{(*j)}\}_{j=1}^{N_p}$, to generate a new sample, $\{\mathbf{x}_k^{(j)}\}_{j=1}^{N_p}$, then set $w_{k|k}^{(j)} = 1/N_p$ for $j = 1, \ldots, N_p$. We then have:

$$p_{k|k}(\mathbf{x}_k|\mathbf{z}_{1:k}) \approx \frac{1}{N_p} \sum_{j=1}^{N_p} \delta(\mathbf{x}_k - \mathbf{x}_k^{(j)}) \tag{2.48}$$

At each stage the mean of the posterior distribution is used to estimate, $\hat{\mathbf{x}}_{k|k}$ of the target state, $\mathbf{x}_k$, i.e.,

$$\begin{aligned} \hat{\mathbf{x}}_{k|k} &= \mathbb{E}[\mathbf{x}_k|\mathbf{z}_{1:k}] && (2.49) \\ &= \int_{\mathbf{x}_k} \mathbf{x}_k \, p(\mathbf{x}_k|\mathbf{z}_{1:k}) d\mathbf{x}_k && (2.50) \\ &\approx \frac{1}{N_p} \sum_{j=1}^{N_p} \mathbf{x}_k^{(j)} && (2.51) \end{aligned}$$

2.2 Multitarget Filtering

In a multitarget tracking problem, the number of targets changes over time as new targets may appear in the region of interest due to spontaneous target birth or target spawning. Moreover, existing targets may not survive to the next time step and disappear from the scene. The sensor does not generate measurements corresponding to all targets present in its field of view. Furthermore, the collections of measurements available at each time step often include spurious measurements not generated by the targets, known as clutter. Finally, measurements are indistinguishable from each other; hence there is no way of knowing which measurement is generated by a target or is clutter. As a result, multi-target tracking involves jointly estimating the number as well as the states of a finite but time varying number of targets from a given set

of finite and time varying number of measurements of uncertain origins.

To summarize, let X_k denote the collection of target states in the region of interest at time k, i.e., $X_k = \{\mathbf{x}_k^1, \ldots, \mathbf{x}_k^{N_k}\} \subseteq \mathbb{R}^{n_x}$. Similarly, Z_k denotes the collection of measurements at time k, i.e., $Z_k = \{\mathbf{z}_{k,1}, \ldots, \mathbf{z}_{k,M_k}\} \subseteq \mathbb{R}^{n_z}$. A measurement $\mathbf{z}_{k,i}$ is either generated by one of N_k targets or is clutter. M_k and N_k represent the number of measurements and targets respectively at time k, and are both time dependent. Let $Z_{1:k} = \{Z_1, \ldots, Z_k\}$ denote the sequence of measurement sets received from the sensor up to time k. Given $Z_{1:k}$, the objective is to find the estimate of the target number, $\hat{N}_{k|k}$, and the estimates of individual states.

2.2.1 Conventional Multitarget Filtering

In conventional methods, multitarget tracking is treated as two separate functions [6][8][56]; association of correct measurement to existing tracks and estimation of target states based on these associations using single target state estimation method discussed in Section 2.1. A simple approach to perform the association is the Nearest Neighbor (NN) method, i.e., associate a measurement that is closest in some statistical sense to a track. Only the measurements that fall within the gate are considered. However, when one or more measurements fall within gates of more than one track, likely associations amongst measurements to tracks depend on which track is considered first. The Global Nearest Neighbor (GNN) method considers all possible ways of associating measurements to tracks and chooses the one that minimizes the sum of the statistical distances between the tracks and measurements.

Multiple Hypothesis Tracking (MHT) is a widely used technique that considers all possible associations amongst measurements and tracks at each time step and allows measurement that will arrive in subsequent time steps to resolve the uncertainty in

associations at present by maintaining and updating their probabilities. However, the complexity and inherent computational costs of such exhaustive data association are considerable. In practice, MHT uses various ad hoc techniques to manage the number of hypotheses from growing exponentially with time. In multidimensional assignment (also known as s-D assignment) algorithm , which a sliding window version of the MHT, the MTT problem is solved using optimization formulation [67]. The 2-D assignment algorithm is equivalent to GNN. The Joint Probabilistic Data Association (JPDA) filter [6][29], and the Probabilistic MHT (PMHT) algorithm [93] consider soft association by weighting the measurements probabilistically. Instead of allowing all feasible associations to propagate ahead in time, JPDA considers associations that survive gating and combines these associations in proportion to their likelihoods. PMHT forms synthetic measurement for each track from gated measurements using Expectation Maximization (EM) algorithm. These explicit association methods require combinatorial enumeration of targets and measurements that result in a huge computational load [63][98].

2.2.2 Unified Multitarget Filtering

On the other hand, the multitarget problem can be modeled using RFS [32][66]. This association-free approach considers the collection of target states as a single meta-target state and the collection of observations as a single meta-observation. It leads to an elegant multitarget generalization of the single-target Bayes filter [63][64][65][66]. By generalizing the single target recursive Bayes filter, the multitarget prediction and

update can be written respectively as follows [66][63]:

$$p_{\Xi,k|k-1}(X_k|Z_{1:k-1}) = \int p_{\Xi,k|k-1}(X_k|X_{k-1})p_{\Xi,k-1|k-1}(X_{k-1}|Z_{1:k-1})\delta X_{k-1} \tag{2.52}$$

$$p_{\Xi,k|k}(X_k|Z_{1:k}) = \frac{p_{\Xi,k}(Z_k|X_k)p_{\Xi,k|k-1}(X_k|Z_{1:k-1})}{p_{\Xi,k}(Z_k|Z_{1:k-1})} \tag{2.53}$$

where $p_{\Xi,k|k}(X_k|Z_{1:k})$ is the multitarget posterior density at time k, $p_{\Xi,k}(Z_k|X_k)$ is the multitarget likelihood function, $p_{\Xi,k|k-1}(X_k|X_{k-1})$ is the multitarget Markov transition density, $p_{\Xi,k|k-1}(X_k|Z_{1:k-1})$ is the prediction of the multitarget posterior $p_{\Xi,k-1|k-1}(X_{k-1}|Z_{1:k-1})$, and $p_{\Xi,k}(Z_k|Z_{1:k-1})$ is the Bayes normalizing factor. It is not possible to compare the states of different dimensions using ordinary Bayesian statistics of fixed dimensional spaces. However, FISST [32][61] paves the way to address this problem by constructing the multitarget densities from multitarget transition functions using the computation of set derivatives of belief-mass functions [61], which makes it possible to combine states of different dimensions. This approach is limited by the dimension of the full state space, which increases with number of targets, hence the computational load for propagating the full posterior increases exponentially with the number of targets. To avoid this problem, the first-order moment, also known as the PHD [61], of multitarget state density can be propagated. Since the PHD is defined over the single target state space in contrast to the full posterior distribution, which is defined over the state space of all the targets, the computational cost of propagating the PHD over time is much lower than that of propagating the full posterior density. A comparison of multitarget filtering using the complete FISST particle filter and the PHD particle filter in terms of computation and estimation accuracy is given in [88].

Chapter 3

PHD Filter

The PHD filter is a suboptimal but computationally tractable alternative to the multitarget Bayes filter described in Section 2.2.2. As shown below, it is a recursion that only propagates the first order moments of the RFS of the targets, which is known as the PHD function or the intensity function. Rest of this chapter introduces the intensity function and its recursion via the PHD filter. It also outlines the derivation of PHD filter recursion using physical space approach. Finally, it provides an overview of implementations of the PHD filter.

$$\begin{array}{ccccccccc} \cdots & \rightarrow & p_{\Xi,k-1|k-1}(X|Z_{1:k-1}) & \overset{\text{time prediction}}{\rightarrow} & p_{\Xi,k|k-1}(X|Z_{1:k-1}) & \overset{\text{Bayes' rule}}{\rightarrow} & p_{\Xi,k|k}(X|Z_{1:k}) & \rightarrow & \cdots \\ & & \downarrow & & \downarrow & & \downarrow & & \\ \cdots & \rightarrow & D_{k-1|k-1} & \overset{\text{predictor}}{\rightarrow} & D_{k|k-1} & \overset{\text{corrector}}{\rightarrow} & D_{k|k} & \rightarrow & \cdots \end{array}$$

3.1 The PHD

The first order statistical moment (the PHD) $D(\mathbf{x})$ of the multitarget state density $p_{\Xi}(X)$ is defined as [63] [66]

$$\begin{aligned} D(\mathbf{x}) &\triangleq \mathbb{E}[\delta_{\Xi}(\mathbf{x})] = \int \delta_X(\mathbf{x}) p_{\Xi}(X) \delta X \\ &= \int p_{\Xi}\Big(\{\mathbf{x}\} \cup X\Big) \delta X \end{aligned} \tag{3.1}$$

where $\mathbf{x}$ is the target state vector, X is the multitarget state set, $\delta_{\Xi}(\mathbf{x})$ is the density on RFS Ξ and equals the summation of Dirac delta functions centered at $\mathbf{w}$ for each $\mathbf{w} \in \Xi$. That is,

$$\delta_{\Xi}(\mathbf{x}) = \sum_{\mathbf{w} \in \Xi} \delta_{\mathbf{w}}(\mathbf{x}) \tag{3.2}$$

The PHD is a unique function defined on single target state space E and its integral over a measurable subset $S \subseteq E$ (e.g., the region of interest) yields the expected number of targets in S. That is,

$$\hat{N}_S = \int_S D(\mathbf{x}) d\mathbf{x} \tag{3.3}$$

The suboptimal filter, which propagates the PHD, provides estimates for the number of targets as well as individual target state.

3.2 PHD Filter Recursion

The recursive propagation involves two steps: the prediction (Section 3.2.1) and the update (Section 3.2.2) with the following assumptions [63]:

- Targets evolve independent of each other
- Target originated measurements are independent of each other
- Clutter is Poisson and independent of target originated measurements
- The predicted multitarget RFS follows a Poisson process

Here, the reader is referred to [63] for further mathematical details of the PHD filter. The local maxima of the PHD correspond to approximate expected states of the targets. One cycle of the recursive propagation of the PHD can be summarized as follows.

3.2.1 Prediction of PHD

In the prediction step, the predicted PHD, $D_{k|k-1}(\mathbf{x}_k|Z_{1:k-1})$, at time k given all measurements up to time $k-1$ is given by

$$D_{k|k-1}(\mathbf{x}_k|Z_{1:k-1}) = \Phi_{k|k-1}D_{k-1|k-1}(\mathbf{x}_{k-1}|Z_{1:k-1}) \tag{3.4}$$

where $D_{k-1|k-1}(\mathbf{x}_{k-1}|Z_{1:k-1})$ is the PHD at time $k-1$ and the PHD prediction operator $\Phi_{k|k-1}$ is defined by

$$(\Phi_{k|k-1}\alpha)(\mathbf{x}) = \gamma_k(\mathbf{x}) + \int \phi_{k|k-1}(\mathbf{x},\xi)\alpha(\xi)d\xi \tag{3.5}$$

for any integrable function α on E with $\gamma_k(\cdot)$ denoting the intensity function of newly appearing targets at time k,

$$\phi_{k|k-1}(\mathbf{x},\xi) = b_{k|k-1}(\mathbf{x}|\xi) + P_s(\mathbf{x})f_{k|k-1}(\mathbf{x}|\xi) \tag{3.6}$$

Here, $P_s(\mathbf{x})$ is the probability of target survival, $f_{k|k-1}(\mathbf{x}|\xi)$ is the single-target Markov transition density, and $b_{k|k-1}(\mathbf{x}|\xi)$ is the intensity function of the spawning targets.

3.2.2 Single-Sensor Update of PHD

For single sensor tracking, the updated PHD $D_{k|k}(\mathbf{x}_k|Z_{1:k})$ at time k is given by

$$D_{k|k}(\mathbf{x}_k|Z_{1:k}) \cong \Psi_k D_{k|k-1}(\mathbf{x}_k|Z_{1:k-1}) \tag{3.7}$$

where it is assumed that the multitarget posterior is approximately Poisson and the update operator Ψ_k is defined by

$$(\Psi_k \alpha)(\mathbf{x}) = \left[\sum_{\mathbf{z}_k \in Z_k} \frac{P_d(\mathbf{x}) p_k(\mathbf{z}_k|\mathbf{x})}{\lambda_k c_k(\mathbf{z}_k) + \psi(\mathbf{z}_k, \alpha)} + 1 - P_d(\mathbf{x})\right] \alpha(\mathbf{x}) \tag{3.8}$$

with

$$\psi(\mathbf{z}_k, \alpha) = \int P_d(\mathbf{x}) h_k(\mathbf{z}_k|\mathbf{x}) \alpha(\mathbf{x}) d\mathbf{x} \tag{3.9}$$

Here, $P_d(\mathbf{x})$ is the probability of detection, $p_k(\mathbf{z}_k|\mathbf{x})$ is the single target likelihood function, λ_k is the false alarm intensity, and $c_k(\mathbf{z}_k)$ is the false alarm spacial density. Since the domain of the intensity function is the same as the state space of an individual target, its propagation requires much less computational power than that of the multitarget posterior. However, the above recursion still involves multiple integrals that have no closed-form expressions in general. Therefore, approximate methods can be sought: namely the SMC method [88][95][96] and the Gaussian mixture method [98].

3.3 Derivation of the PHD Filter Recursion

In this section, the derivation of the PHD filter recursion using the physical-space approach [27], also used in derivation of the novel backward PHD smoothing formula in Section 5.1, is reviewed (more details can be found in [66, pp. 599–609]). According to the physical-space approach, PHD is the target density corresponding to the probability of having a target in an elementary (hyper)volume in the state space. It is assumed that the surveillance region is divided into infinitesimal bins, $(b_i,\ i = 0, \ldots, \infty)$, and each bin has at most one target. Let $|b_i|$ denote the bin-(hyper)volume and $u(\cdot)$ denote the indicator function, which is defined as

$$u(b_i) = \begin{cases} 1 & \text{Bin } b_i \text{ having a target} \\ 0 & \text{Otherwise} \end{cases}$$

Proposition 3.3.1 *The bin probability, $P\{u(b_i) = 1\}$, of the* ith *bin approaches the PHD, $D(\mathbf{x})$, when bin (hyper)volume approaches* 0. *That is,*

$$\lim_{|b_i| \to 0} \frac{P\{u(b_i) = 1\}}{|b_i|} = D(\mathbf{x}) \tag{3.10}$$

Proof Consider a locally compact Hausdorff separable space E (e.g., $\mathbb{R}^{n_x}$). An RFS Ξ on E is defined as a measurable mapping [32]

$$\Xi : \Omega \to \mathcal{F}(E)$$

where Ω is a sample space with a probability measure P defined on $\sigma(\Omega)$ and $\mathcal{F}(E)$ denotes the collection of finite subsets of E. The bin probability in (3.10) can be

written as

$$\begin{aligned} P\{u(b_i)=1\} &= P\{\omega : \Xi(\omega) \in S_{b_i}\} \\ &= \int_{S_{b_i}} \left(\sum_{\mathbf{x}\in X} \mathbf{1}_{S_{b_i}}(\mathbf{x}) \right) \cdot f(X)\delta X \end{aligned} \tag{3.11}$$

where S_{b_i} is the set of states in the bin b_i, $\mathbf{1}_{S_{b_i}}(\mathbf{x})$ is the indicator function of set S_{b_i}, and $f(X)$ is the multitarget state density. Set integration is defined as follows [66]:

$$\int g(X)\delta X = \sum_{n=0}^{\infty} \frac{1}{n!} \int g\left(\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}\right) d\mathbf{x}_1, \ldots, d\mathbf{x}_n \tag{3.12}$$

where $g(X)$ is any integrable function of set X. Using (3.12), (3.11) can be simplified as follows:

$$\begin{aligned} P\{u(b)=1\} &= \sum_{n=0}^{\infty} \frac{1}{n!} \int_{S_{b_i}} \left(\mathbf{1}_{S_{b_i}}(\mathbf{x}_1) + \ldots + \mathbf{1}_{S_{b_i}}(\mathbf{x}_n) \right) \cdot f\left(\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}\right) d\mathbf{x}_1, \ldots, d\mathbf{x}_n \\ &= \sum_{n=0}^{\infty} \frac{n}{n!} \int_{S_{b_i}} \mathbf{1}_{S_{b_i}}(\mathbf{x}_n) \cdot f\left(\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}\right) d\mathbf{x}_1, \ldots, d\mathbf{x}_n \\ &= \sum_{n=0}^{\infty} \frac{1}{(n-1)!} \int_{S_{b_i}} f\left(\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}\right) d\mathbf{x}_1, \ldots, d\mathbf{x}_n \\ &= \int_{b_i} \left(\sum_{j=0}^{\infty} \frac{1}{j!} \int f\left(\{\mathbf{x}_1, \ldots, \mathbf{x}_j, \mathbf{x}\}\right) d\mathbf{x}_1, \ldots, d\mathbf{x}_j \right) d\mathbf{x} \\ &= \int_{b_i} \left(\int f\left(Y \cup \{\mathbf{x}\}\right) \delta Y \right) d\mathbf{x} \end{aligned} \tag{3.13}$$

In the above, the second line is due to the assumption that states are independent

and identical. Consider the following limit

$$\begin{aligned}\lim_{|b_i|\to 0}\frac{P\{u(b_i)=1\}}{|b_i|} &= \lim_{|b_i|\to 0}\frac{1}{|b_i|}\int_{b_i}\left(\int f\left(Y\cup\{\mathbf{x}\}\right)\delta Y\right)d\mathbf{x} \\ &= \lim_{|b_i|\to 0}\frac{1}{|b_i|}\int_{b_i}d\mathbf{x}\int f\left(Y\cup\{\mathbf{x}\}\right)\delta Y \qquad (3.14)\end{aligned}$$

In the above, the second line is due to the infinitesimal size of the bin, which includes state $\mathbf{x}$. The first integration in (3.14) is equal to the (hyper)volume $|b_i|$. From (3.1), the second integration is equal to the PHD that leads to the limiting augment of the physical space approach in (3.10). ∎

Using this interpretation, the PHD filtering steps that are given by (3.5) and (3.7) can be obtained by considering the limiting values of respective bin probabilities as follows.

3.3.1 Prediction

Consider the possible events for the ith bin having target at time k

- Spontaneous appearance of a target in bin i, with probability $P_b(u_k(b_i)=1)$
- The target in the jth bin at time $k-1$ moved to the ith bin at time k. Its probability is given by

$$P\Big\{u_{k-1}(b_j)=1|Z_{1:k-1}\Big\}\times P_s\Big\{u_{k-1}(b_j)=1\Big\}\times P\Big\{u_k(b_i)=1|u_{k-1}(b_j)=1\Big\}$$

  The three terms in the above expression are the probability for the jth bin having a target at time $k-1$, its survival probability and the probability for that target moving into the ith bin at time k, respectively.

- The target in the jth bin at time $k-1$ spawned a target in the ith bin at time k. Its probability is given by

$$P\Big\{u_{k-1}(b_j)=1|Z_{1:k-1}\Big\}\times P_g\Big\{u_k(b_i)=1|u_{k-1}(b_j)=1\Big\}$$

  Here, the last term is the probability for the target in the jth bin spawning a target in the ith bin at time k.

Since the assumption of infinitesimal bins makes the above events to be mutually exclusive, the bin probabilities at time k given the measurement set $Z_{1:k-1}$ can be written as

$$\begin{aligned}P\Big\{u_k(b_i)=1|Z_{1:k-1}\Big\} &= P_b\Big\{u_k(b_i)=1\Big\}\\ &\quad+\sum_j\Big[P_s\Big\{u_{k-1}(b_j)=1\Big\}P\Big\{u_k(b_i)=1|u_{k-1}(b_j)=1\Big\}\\ &\quad+P_g\Big\{u_k(b_i)=1|u_{k-1}(b_j)=1\Big\}\Big]\\ &\quad\times P\Big\{u_{k-1}(b_j)=1|Z_{1:k-1}\Big\}\end{aligned}\tag{3.15}$$

Dividing both sides by (hyper)volume $|b_i|$ and rearranging result in

$$\begin{aligned}\frac{P\Big\{u_k(b_i)=1|Z_{1:k-1}\Big\}}{|b_i|} &= \frac{P_b\Big\{u_k(b_i)=1\Big\}}{|b_i|}\\ &\quad+\sum_j\Bigg[P_s\Big\{u_{k-1}(b_j)=1\Big\}\cdot\frac{P\Big\{u_k(b_i)=1|u_{k-1}(b_j)=1\Big\}}{|b_i|}\\ &\quad+\frac{P_g\Big\{u_k(b_i)=1|u_{k-1}(b_j)=1\Big\}}{|b_i|}\Bigg]\\ &\quad\times\frac{P\Big\{(u_{k-1}(b_j)=1|Z_{1:k-1}\Big\}}{|b_j|}\cdot|b_j|\end{aligned}\tag{3.16}$$

Note that the (hyper)volume $|b_j|$ is inserted in two places such that they cancel each other. In the limiting case, bin probabilities approach to respective PHDs as follows:

$$\begin{aligned}
\frac{P\left\{u_k(b_i)=1|Z_{1:k-1}\right\}}{|b_i|} &\rightarrow D_{k|k-1}(\mathbf{x}_k|Z_{1:k-1}) \\
\frac{P_b\left\{u_k(b_i)=1\right\}}{|b_i|} &\rightarrow \gamma_k(\mathbf{x}_k) \\
\frac{P\left\{u_{k-1}(b_j)=1|Z_{1:k-1}\right\}}{|b_j|} &\rightarrow D_{k-1|k-1}(\mathbf{x}_{k-1}|Z_{1:k-1}) \\
\frac{P_g\left\{u_k(b_i)=1|u_{k-1}(b_j)=1\right\}}{|b_i|} &\rightarrow b_{k|k-1}(\mathbf{x}_k|\mathbf{x}_{k-1})
\end{aligned}$$

In addition, the state transition probability $\frac{P\{u_k(b_i)=1|u_{k-1}(b_j)=1\}}{|b_i|}$ approaches to state transition density $f_{k|k-1}(\mathbf{x}_k|\mathbf{x}_{k-1})$, the survival probability $P_s\{u_{k-1}(b_j)=1\} \rightarrow P_s(\mathbf{x}_{k-1})$, and the bin (hyper)volume $|b_j| \rightarrow d\mathbf{x}_{k-1}$. The substitution of these limiting values into (3.16) leads to the continuous PHD prediction equation of (3.5).

3.3.2 Update

Using Bayes' rule, the posterior bin probability can be written as

$$P\left\{u_k(b_i)=1|Z_{1:k}\right\} = \frac{P\left\{Z_k|u_k(b_i)=1,Z_{1:k-1}\right\}}{P\{Z_k|Z_{1:k-1}\}} P\left\{u_k(b_i)=1|Z_{1:k-1}\right\} \tag{3.17}$$

where the normalizing constant $P\{Z_k|Z_{1:k-1}\}$ is given by

$$\begin{aligned}
P\{Z_k|Z_{1:k-1}\} &= P\left\{Z_k|u_k(b_i)=1,Z_{1:k-1}\right\}P\left\{u_k(b_i)=1|Z_{1:k-1}\right\} \\
&\quad + P\left\{Z_k|u_k(b_i)=0,Z_{1:k-1}\right\} \\
&\quad \times \left[1-P\left\{u_k(b_i)=1|Z_{1:k-1}\right\}\right]
\end{aligned} \tag{3.18}$$

Since the infinitesimal bin size forces the bin probability, $P\Big\{u_k(b_i) = 1|Z_{1:k-1}\Big\}$, to an arbitrarily small value, the first-order Taylor approximation to the right hand side of (3.17) results in [66]

$$P\Big\{u_k(b_i) = 1|Z_{1:k}\Big\} \cong \frac{P\Big\{Z_k|u_k(b_i) = 1, Z_{1:k-1}\Big\}}{P\Big\{Z_k|u_k(b_i) = 0, Z_{1:k-1}\Big\}} P\Big\{u_k(b_i) = 1|Z_{1:k-1}\Big\} \tag{3.19}$$

The above expression can be simplified under the assumption that the number of targets in the surveillance region follows Poisson distribution as follows [66][27]:

$$\begin{aligned} &P\Big\{u_k(b_i) = 1|Z_{1:k}\Big\} \cong \\ &\left[\sum_{\mathbf{z}_k \in Z_k} \frac{P_d\{b_i\} P\Big\{\mathbf{z}_k|u_k(b_i) = 1\Big\}}{\lambda_k c(\mathbf{z}_k) + \sum_j P_d\{b_j\} P\Big\{\mathbf{z}_k|u_k(b_j) = 1\Big\} P\Big\{u_k(b_j) = 1|Z_{1:k-1}\Big\}} + \Big(1 - P_d\{b_i\}\Big)\right] \\ &\times P\Big\{u_k(b_i) = 1|Z_{1:k-1}\Big\} \end{aligned} \tag{3.20}$$

After dividing both sides by (hyper)volume $|b_i|$, for the limiting case, (3.20) becomes the continuous PHD corrector equation of (3.7).

3.4 Sequential Monte Carlo Implementation of PHD

In the SMC implementation of the PHD filter [88][95][96], the PHD surface is represented by a set of particles. These particles are propagated according to the PHD recursion given by (3.5) and (3.7). Each recursion involves particle prediction, weight update and resampling. The kth recursion starts with the particle set $\Big\{\mathbf{x}_{k-1}^{(j)}, w_{k-1|k-1}^{(j)}\Big\}_{j=1}^{L_{k-1}}$ that represents the PHD at time $k-1$. Here, L_{k-1} is the number of particles at time $k-1$. In the prediction step, samples for existing targets are

drawn as follows:

$$\mathbf{x}_{k|k-1}^{(j)} \sim q_k(\cdot|\mathbf{x}_{k-1}^{(j)}, Z_k) \quad j = 1, \ldots, L_{k-1}$$

where $q_k(\cdot)$ is the proposal density function. The associated weights are given by

$$w_{k|k-1}^{(j)} = \frac{P_s(\mathbf{x}_{k-1}^{(j)}) f_{k|k-1}(\mathbf{x}_{k|k-1}^{(j)}|\mathbf{x}_{k-1}^{(j)}) + b_{k|k-1}(\mathbf{x}_{k|k-1}^{(j)}|\mathbf{x}_{k-1}^{(j)})}{q_k(\mathbf{x}_{k|k-1}^{(j)}|\mathbf{x}_{k-1}^{(j)}, Z_k)} w_{k-1|k-1}^{(j)}$$

Also, samples for new-born targets are drawn from birth proposal intensity function $\rho_k(\cdot)$. Then

$$\mathbf{x}_{k|k-1}^{(j)} \sim \rho_k(\cdot|Z_k) \quad j = L_{k-1} + 1, \ldots, L_{k-1} + J_k$$

where J_k is the number of particles per new-born target. The associated weights are given by

$$w_{k|k-1}^{(j)} = \frac{\gamma_k(\mathbf{x}_{k|k-1}^{(j)})}{\rho_k(\mathbf{x}_{k|k-1}^{(j)}|Z_k)}$$

In the update step, the particle weights are updated as follows:

$$w_{k|k}^{(*j)} = \left[\left(1 - P_d(\mathbf{x}_{k|k-1}^{(j)})\right) + \sum_{i=1}^{N_k^Z} \frac{P_d(\mathbf{x}_{k|k-1}^{(j)}) f_{k|k-1}(\mathbf{z}_k^i|\mathbf{x}_{k|k-1}^{(j)})}{\lambda_k c_k(\mathbf{z}_k^i) + \Psi_k(\mathbf{z}_k^i)} \right] w_{k|k-1}^{(j)}$$

where

$$\Psi_k(\mathbf{z}_k^i) = \sum_{j=1}^{L_{k-1}+J_k} P_d(\mathbf{x}_{k|k-1}^{(j)}) f(\mathbf{z}_k^i|\mathbf{x}_{k|k-1}^{(j)}) w_{k|k-1}^{(j)}$$

The expected number of targets at time k is given by

$$\hat{N}_{k|k} = \sum_{j=1}^{L_{k-1}+J_k} w_{k|k}^{(*j)}$$

Finally, the particle set $\left\{w_{k|k}^{(*j)}/\hat{N}_{k|k}, \mathbf{x}_{k|k-1}^{(j)}\right\}_{j=1}^{L_{k-1}+J_k}$ is resampled to get a new set of particles $\left\{w_{k|k}^{(j)}/\hat{N}_{k|k}, \mathbf{x}_k^{(j)}\right\}_{j=1}^{L_k}$, where $L_k = \hat{N}_{k|k}N_p$ and N_P is the number of particles per target. For output purposes, the estimated number of targets is given by $\hat{N}_k = \text{round}(\hat{N}_{k|k})$, where round$(\cdot)$ refers to the nearest integer. The state estimates for $\hat{N}_k$ targets can be extracted from the particles approximating the posterior PHD function using clustering techniques [96] or Expectation-Maximization (EM) [94].

3.5 Gaussian Mixture Implementation of PHD

In Gaussian mixture implementation of the PHD filter, a closed-form recursion is performed with the following assumptions [97] [98]:

- Targets evolve according to a linear Gaussian dynamic model, i.e.,

$$f_{k|k-1}(\mathbf{x}|\zeta) = \mathcal{N}(\mathbf{x}; F_{k-1}\zeta, Q_{k-1}) \tag{3.21}$$

  where $\mathcal{N}(\cdot; \mathbf{m}, P)$ denotes a Gaussian density with mean $\mathbf{m}$ and covariance P.

- Measurement model is linear Gaussian, i.e.,

$$h_k(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}; H_k\mathbf{x}, R_k) \tag{3.22}$$

- The target survival and detection probabilities are state independent, i.e.,

$$P_s(\mathbf{x}) = P_s \tag{3.23}$$

$$P_d(\mathbf{x}) = P_d \tag{3.24}$$

- The intensities of the birth and spawn RFSs are Gaussian mixtures of the form

$$\gamma_k(\mathbf{x}) = \sum_{i=1}^{J_{\gamma,k}} w_{\gamma,k}^i \mathcal{N}(\mathbf{x}; \mathbf{m}_{\gamma,k}^i, P_{\gamma,k}^i) \tag{3.25}$$

$$b_{k|k-1}(\mathbf{x}|\zeta) = \sum_{i=1}^{J_{b,k}} w_{b,k}^i \mathcal{N}(\mathbf{x}; F_{b,k-1}^i \zeta + \mathbf{d}_{b,k-1}^i, Q_{b,k-1}^i) \tag{3.26}$$

where $J_{\gamma,k}$ is the number of Gaussian components of the birth intensity surface, and $w_{\gamma,k}^i$, $\mathbf{m}_{\gamma,k}^i$, and $P_{\gamma,k}^i$ are the weight, mean and covariance of the ith component respectively. Similarly, $J_{b,k}$, $w_{b,k}^i$, $F_{b,k-1}^i$, and $Q_{b,k-1}^i$, $i = 1, \ldots, J_{b,k}$ are parameters of the spawning intensity of a target with previous state ζ.

The GMPHD filter recursion at time k starts with the PHD at time $k-1$, which is represented by a Gaussian mixture, i.e.,

$$D_{k-1|k-1}(\mathbf{x}_{k-1}|Z_{1:k-1}) = \sum_{i=1}^{J_{k-1}} w_{k-1|k-1}^i \mathcal{N}(\mathbf{x}; \mathbf{m}_{k-1|k-1}^i, P_{k-1|k-1}^i) \tag{3.27}$$

The predicted PHD at time k is given by

$$D_{k|k-1}(\mathbf{x}_k|Z_{1:k-1}) = D_{s,k|k-1}(\mathbf{x}_k|Z_{1:k-1}) + D_{b,k|k-1}(\mathbf{x}_k|Z_{1:k-1}) + \gamma_k(\mathbf{x}_k) \tag{3.28}$$

The term $D_{S,k|k-1}(\mathbf{x}_k|Z_{1:k-1})$ corresponds to surviving targets and is a Gaussian mixture, whose components are computed using the standard Kalman prediction equations as follows:

$$D_{s,k|k-1}(\mathbf{x}_k|Z_{1:k-1}) = P_s \sum_{i=1}^{J_{k-1}} w_{k-1|k-1}^i \mathcal{N}(\mathbf{x}; \mathbf{m}_{s,k|k-1}^i, P_{s,k|k-1}^i) \quad (3.29)$$

$$\mathbf{m}_{s,k|k-1}^i = F_{k-1}\mathbf{m}_{k-1|k-1}^i \quad (3.30)$$

$$P_{s,k|k-1}^i = Q_{k-1} + F_{k-1}P_{k-1|k-1}^i F_{k-1}^T \quad (3.31)$$

The second term in (3.28) corresponds to spawning targets and is given by

$$D_{b,k|k-1}(\mathbf{x}_k|Z_{1:k-1}) = \sum_{i=1}^{J_{k-1}} \sum_{j=1}^{J_{b,k}} w_{k-1|k-1}^i w_{b,k}^j \mathcal{N}(\mathbf{x}; \mathbf{m}_{b,k|k-1}^{i,j}, P_{b,k|k-1}^{i,j}) \quad (3.32)$$

$$\mathbf{m}_{b,k|k-1}^{i,j} = F_{b,k-1}^j \mathbf{m}_{k-1|k-1}^i + \mathbf{d}_{b,k-1}^j \quad (3.33)$$

$$P_{b,k|k-1}^{i,j} = Q_{b,k-1}^j + F_{b,k-1}^j P_{k-1|k-1}^i (F_{b,k-1}^j)^T \quad (3.34)$$

Here, A^T denotes the transpose of matrix A. The last term in (3.28) corresponds to newly born targets and is given by (3.25). The update PHD at time k is given by

$$D_{k|k}(\mathbf{x}_k|Z_{1:k}) = (1 - P_d)D_{k|k-1}(\mathbf{x}_k|Z_{1:k-1}) + \sum_{\mathbf{z}\in Z_k} D_{z,k}(\mathbf{x}_k; \mathbf{z}) \quad (3.35)$$

where the first term corresponds to missed-detection and the second term corresponds to measurements. The parameters of the each component are computed using the

standard Kalman update equations as follows:

$$D_{z,k}(\mathbf{x}_k;\mathbf{z}) = \sum_{i=1}^{J_{k|k-1}} w^i_{k|k}(\mathbf{z})\mathcal{N}(\mathbf{x};\mathbf{m}^i_{k|k}(\mathbf{z}),P^i_{k|k}) \tag{3.36}$$

$$w^i_k = \frac{P_d w^i_{k|k-1} q^i_k(\mathbf{z})}{\lambda_k c_k(\mathbf{z}_k) + P_d \sum_{j=1}^{J_{k|k-1}} w^j_{k|k-1} q^j_k(\mathbf{z})} \tag{3.37}$$

$$q^j_k(\mathbf{z}) = \mathcal{N}(\mathbf{z};H_k\mathbf{m}^j_{k|k-1},R_k + H_k P^j_{k|k-1}H^T_k) \tag{3.38}$$

$$\mathbf{m}^i_{k|k}(\mathbf{z}) = \mathbf{m}^i_{k|k-1} + K^i_k(\mathbf{z} - H_k\mathbf{m}^i_{k|k-1}) \tag{3.39}$$

$$P^i_{k|k} = [I - K^i_k H_K]P^i_{k|k-1} \tag{3.40}$$

$$K^i_k = P^i_{k|k-1}H^T_k\left(H_k P^i_{k|k-1}H^T_k + R_k\right)^{-1} \tag{3.41}$$

For nonlinear systems, the EKF [7] or the UKF [47] can be used. In this work, the UKF is used to handle the nonlinearity of the system. This Gaussian mixture PHD filter suffers from computation problems associated with the increasing number of Gaussian components as time progresses. Gaussian mixture reduction methods [84][86][87] can be used to mitigate this problem. A simple heuristic pruning method is given in [98].

Chapter 4

Smoothing

The state estimation problem can be cast as the determination of the conditional density $p_{t|k}(\mathbf{x}_t|\mathbf{z}_{1:k})$, where $\mathbf{x}_t$ is the target state at time t and $\mathbf{z}_{1:k}$ is the collection of measurements up to time k. If $t > k$ then it is called prediction; if $t = k$ as in the case of Chapter 2, then it is known as filtering; and if $t < k$ then it is called smoothing or retrodiction [23][24][25]. Since smoothing uses more measurements beyond the current estimation time, it yields improved estimates than that of filtering or prediction. If a certain time delay can be tolerated, accurate estimates can be obtained using smoothing, which requires some additional computational load to incorporate to the information current state from the measurements at time beyond current estimation time.

Smoothing problems can be categorized into three classes according to the point at which the estimate is interested. Fixed-interval smoothing is concerned with finding the density $p_{t|k}(\mathbf{x}_t|\mathbf{z}_{1:k})$ for all time indices $t = 1, \ldots, k$. It is most commonly used in off-line applications. Fixed-point smoothing is always concerned with the density at a fixed time t while k varies. Fixed-lag smoothing is concerned with the density

at time $t = k - L$, where L is the fixed time lag. It is very suitable for online tracking problems that can tolerate delayed estimates. All three problems can be solved by employing a single smoothing scheme based on fixed-interval smoothing [14][89]. The most common schemes used to perform the fixed-interval smoothing are two-filter method [13][50][51] and forward-backward smoothing [2][89]. The two-filter method involves fusion of estimates from forward and back filters. The backward filter requires invertible state transition model. On the other hand forward-backward is easy to implement and does not require invertible state transition model [89].

This thesis is concerned with improving the estimates for multitarget problem using fixed-lag smoothing, which is performed using forward filtering followed by backward smoothing. This RTS type smoothing is incorporated into a number of existing tracking algorithms. In section 4.1, single target smoothing methods are reviewed. In Section 4.2, multitarget smoothing methods are reviewed.

4.1 Single Target Smoothing

For single target, the marginal smoothed posterior distribution $p_{t|k}(\mathbf{x}_t|\mathbf{z}_{1:k})$, where $t < k$, can be obtained using following forward-backward recursive expression.

$$\begin{aligned} p_{t|k}(\mathbf{x}_t|\mathbf{z}_{1:k}) &= \int p_{t|k}(\mathbf{x}_t, \mathbf{x}_{t+1}|\mathbf{z}_{1:k})d\mathbf{x}_{t+1} \\ &= \int p_{t+1|k}(\mathbf{x}_{t+1}|\mathbf{z}_{1:k})p_{t|k}(\mathbf{x}_t|\mathbf{x}_{t+1}, \mathbf{z}_{1:k})d\mathbf{x}_{t+1} \\ &= \int p_{t+1|k}(\mathbf{x}_{t+1}|\mathbf{z}_{1:k})p_{t|k}(\mathbf{x}_t|\mathbf{x}_{t+1}, \mathbf{z}_{1:t})d\mathbf{x}_{t+1} \\ &= p_{t|t}(\mathbf{x}_t|\mathbf{z}_{1:t}) \int \frac{p_{t+1|k}(\mathbf{x}_{t+1}|\mathbf{z}_{1:k})p_{t+1|t}(\mathbf{x}_{t+1}|\mathbf{x}_t)}{p_{t+1|t}(\mathbf{x}_{t+1}|\mathbf{z}_{1:t})}d\mathbf{x}_{t+1} \end{aligned} \tag{4.1}$$

In the above, the first two lines are due to the total probability theorem and the chain rule, respectively. The third line is due to Markov assumption. Finally, the application of the Bayes' rule leads to the required backward smoothing recursion. It is thus possible to compute the filtered and predicted distributions in a forward (filtering) recursion of the algorithm (by calculating $p_{t|t}(\mathbf{x}_t|\mathbf{z}_{1:t})$), and then execute a backward recursion with each smoothed distribution ($p_{t|k}(\mathbf{x}_t|\mathbf{z}_{1:k})$) relying upon the quantities calculated in the previous (in reverse time) smoothed distribution ($p_{t+1|k}(\mathbf{x}_{t+1}|\mathbf{z}_{1:k})$).

4.1.1 Kalman Smoothing

For linear Gaussian problem defined in (2.10) and (2.11), the Bayesian backward recursion in (4.1) become a closed-form recursion, which propagates the mean and the covariance matrix of the smoothed posterior distribution. The backward smoothing recursion at t starts with the filtered density at t ($p_{t|t}(\mathbf{x}_t|\mathbf{z}_{1:t})$) and smoothed density at $t+1$ ($p_{t+1|k}(\mathbf{x}_{t+1}|\mathbf{z}_{1:k})$) that are Gaussian as follows:

$$p_{t|t}(\mathbf{x}_t|\mathbf{z}_{1:t}) = \mathcal{N}(\mathbf{x}_t; \mathbf{m}_{t|t}, P_{t|t}) \tag{4.2}$$

$$p_{t+1|k}(\mathbf{x}_{t+1}|\mathbf{z}_{1:k}) = \mathcal{N}(\mathbf{x}_{t+1}; \mathbf{m}_{t+1|k}, P_{t+1|k}) \tag{4.3}$$

Given the above densities, the smoothed density at t ($p_{t|k}(\mathbf{x}_t|\mathbf{z}_{1:k})$) is given by [20][55]

$$p_{t|k}(\mathbf{x}_t|\mathbf{z}_{1:k}) = \mathcal{N}(\mathbf{x}_t; \mathbf{m}_{t|k}, P_{t|k}) \tag{4.4}$$

where

$$\mathbf{m}_{t|k} = \mathbf{m}_{t|t} + A_t\left(\mathbf{m}_{t+1|k} - \mathbf{m}_{t+1|t}\right) \tag{4.5}$$

$$P_{t|k} = P_{t|t} + A_t\left(P_{t+1|k} - P_{t+1|t}\right)A_t^T \tag{4.6}$$

$$A_t = P_{t|t}F_t^T P_{t+1|t}^{-1} \tag{4.7}$$

$$\mathbf{m}_{t+1|t} = F_t\mathbf{m}_{t|t} \tag{4.8}$$

$$P_{t+1|t} = Q_t + F_t P_{t|t} F_t^T \tag{4.9}$$

This backward recursion is initialized by the filtering result at the present scan k. In the case of nonlinear problems as in (2.36) and (2.37), either extended Kalman smoother [19][85] or unscented Kalman smoother [101] can be used.

4.1.2 Particle Smoothing

Using SMC, the smoothed posterior density in (4.1) can be represented by a set of particles $\{\mathbf{x}_{k-1}^{(j)},\ w_{k-1|k-1}^{(j)}\}_{j=1}^{N_p}$. That is,

$$p_{t|k}(\mathbf{x}_t|\mathbf{z}_{1:k}) \approx \sum_{j=1}^{N_p} w_{t|k}^{(j)}\delta(\mathbf{x}_t - \mathbf{x}_t^{(j)}) \tag{4.10}$$

where the importance weights $w_{t|k}^{(j)}$ are obtained through the following backward recursion:

$$w_{t|k}^{(j)} = w_{t|t}^{(j)}\left[\sum_{i=1}^{N_p} w_{t+1|k}^{(i)}\frac{p_{t+1|t}\left(\mathbf{x}_{t+1}^{(j)}|\mathbf{x}_t^{(i)}\right)}{\sum_{l=1}^{N_p} w_{t|t}^{(l)} p_{t+1|t}\left(\mathbf{x}_{t+1}^{(j)}|\mathbf{x}_t^{(l)}\right)}\right] \tag{4.11}$$

4.2 Multitarget Smoothing

In the case of multitarget tracking problem, smoothing has been incorporated into a number of different conventional tracking algorithms. Probabilistic Data Association (PDA) smoothing algorithm is proposed in [59] to improve the tracking performance in a clutter environment. Maneuvering target tracking is improved by IMM smoothing method in [40]. IMM-PDA smoothing is reported in [17] to improve the tracking of agile targets in clutter environment. In [25], a fixed-lag smoothing is applied to JPDA filtering. Improved multitarget tracking is demonstrated using IMM-MHT smoothing in [53][54].

On the other hand, multitarget smoothing can be unified scheme, where distribution of the multitarget state set are smoothed. Similar to multitarget forward filtering generalization in Section 2.2.2, the backward smoothing can also be generalized with RFS. Given the measurement sets up to time k, the smoothed multitarget state density at time t, $(t < k)$, can be written as

$$p_{\Xi,t|k}(X_t|Z_{1:k}) = p_{\Xi,t|t}(X_t|Z_{1:t}) \int \frac{p_{\Xi,t+1|k}(X_{t+1}|Z_{1:k}) p_{\Xi,t+1|t}(X_{t+1}|X_t)}{p_{\Xi,t+1|t}(X_{t+1}|Z_{1:t})} \delta X_{t+1} \quad (4.12)$$

where $p_{\Xi,t|t}(X_t|Z_{1:t})$ is the filtered multitarget state density at time t, $p_{\Xi,t+1|k}(X_{t+1}|Z_{1:k})$ is the smoothed multitarget state density at time $t+1$, $p_{\Xi,t+1|t}(X_{t+1}|X_t)$ is the multitarget Markov transition density, and $p_{\Xi,t+1|t}(X_{t+1}|Z_{1:t})$ is the Bayes normalizing factor. Since it involves set integration, a computationally tractable first order approximation is useful for practical application that is the prime goal of this thesis.

Chapter 5

PHD Smoothing

In this chapter, a novel smoothing method for PHD based state estimator is developed. The proposed method involves a forward multitarget filtering using the standard PHD filter recursion (Chapter 3) and then a backward smoothing recursion. This backward smoothing recursion is performed with a novel recursive formula, which is derived using the physical-space approach (Section 5.1). The resulting backward recursion incorporates intensity for surviving targets as well as disappearing targets. Compared to optimal multitarget Bayesian smoothing, this first order recursion is simple and evolves in single target state space. However, it does not admit any closed-form recursion. To mitigate this problem, we proposed an SMC implementation of the smoothing method in Section 5.2. We show that this SMC implementation requires much computational effort to compute smoothed particle weights. For this, we introduced a fast implementation in Section 5.2.1. This fast method requires that target transition density should be defined in metric space. However, we have shown that most commonly used transition models can be converted into a function defined in a metric space using simple transformations.

5.1 Derivation of the PHD Smoothing Recursion

In the derivation below, the notations of the physical-space approach in Section 3.3 are extensively used. In order to find the smoothed PHD $D_{t|k}(\mathbf{x}_t|Z_{1:k})$, consider the event where the ith bin has a target at time t, given measurements set $Z_{1:k}$. In the limiting case (i.e., as the bin (hyper)volume approaches zero), its bin probability $P\Big\{u_t(b_i) = 1|Z_{1:k}\Big\}$ will approach the required smoothed PHD. Consider the following possible events for a target in the ith bin at time t:

- It can occupy the jth bin at time $t+1$ with probability

$$P_s\Big\{u_t(b_i) = 1\Big\} \times P\Big\{u_t(b_i) = 1|u_{t+1}(b_j) = 1, Z_{1:k}\Big\} \times P\Big\{u_{t+1}(b_j) = 1|Z_{1:k}\Big\}$$

- It does not survive after time t with probability

$$P\Big\{u_t(b_i) = 1|d_t(b_i), Z_{1:k}\Big\} \times \Big(1 - P_s\Big\{u_t(b_i) = 1\Big\}\Big)$$

  where $d_t(b_i)$ is the event that the target in the ith bin at time t does not survive and $P_s\Big\{u_t(b_i) = 1\Big\}$ is the probability of target survival.

Using the total probability theorem, the above bin probability can be written as

$$\begin{aligned} &P\Big\{u_t(b_i) = 1|Z_{1:k}\Big\} = \\ &\quad \sum_j P_s\Big\{u_t(b_i) = 1\Big\} P\Big\{u_t(b_i) = 1|u_{t+1}(b_j) = 1, Z_{1:k}\Big\} P\Big\{u_{t+1}(b_j) = 1|Z_{1:k}\Big\} \\ &\quad + P\Big\{u_t(b_i) = 1|d_t(b_i), Z_{1:k}\Big\} \Big(1 - P_s\Big\{u_t(b_i) = 1\Big\}\Big) \end{aligned} \tag{5.1}$$

The term $P\Big\{u_t(b_i) = 1|u_{t+1}(b_j) = 1, Z_{1:k}\Big\}$ can be simplified as follows:

$$\begin{aligned} P\Big\{u_t(b_i) = 1|u_{t+1}(b_j) = 1, Z_{1:k}\Big\} &= P\Big\{u_t(b_i) = 1|u_{t+1}(b_j) = 1, Z_{1:t}\Big\} \\ &= \frac{P\Big\{u_{t+1}(b_j) = 1|u_t(b_i) = 1, Z_{1:t}\Big\}}{P\Big\{u_{t+1}(b_j) = 1|Z_{1:t}\Big\}} \\ &\quad \times P\Big\{u_t(b_i) = 1|Z_{1:t}\Big\} \end{aligned} \tag{5.2}$$

In the above, the first line is due to the occupation of a target in bin b_i at time t is independent of measurements $Z_{t+1:k}$ given that it occupies bin b_j at time $t+1$. The second line results from the application of the Bayes' rule. Using Markov assumption, one has $P\Big\{u_t(b_i) = 1|d_t(b_i), Z_{1:k}\Big\} = P\Big\{u_t(b_i) = 1|Z_{1:t}\Big\}$. The substitution of these terms into (5.1) results in following backward smoothing iteration for bin probability:

$$\begin{aligned} &P\Big\{u_t(b_i) = 1|Z_{1:k}\Big\} = \\ &\quad P\Big\{u_t(b_i) = 1|Z_{1:t}\Big\}\Big(1 - P_s\Big\{u_t(b_i) = 1\Big\}\Big) + P\Big\{u_t(b_i) = 1|Z_{1:t}\Big\} \\ &\quad \times \sum_j \frac{P_s\Big\{u_t(b_i) = 1\Big\}P\Big\{u_{t+1}(b_j) = 1|Z_{1:k}\Big\}P\Big\{u_{t+1}(b_j) = 1|u_t(b_i) = 1\Big\}}{P\Big\{u_{t+1}(b_j) = 1|Z_{1:t}\Big\}} \end{aligned} \tag{5.3}$$

Dividing both sides by (hyper)volume $|b_i|$ and rearranging result in

$$\frac{P\Big\{u_t(b_i)=1|Z_{1:k}\Big\}}{|b_i|} = \frac{P\Big\{u_t(b_i)=1|Z_{1:t}\Big\}}{|b_i|} \times \Bigg[\sum_j \frac{P_s\Big\{u_t(b_i)=1\Big\}\frac{P\Big\{u_{t+1}(b_j)=1|Z_{1:k}\Big\}}{|b_j|}\frac{P\Big\{u_{t+1}(b_j)=1|u_t(b_i)=1\Big\}}{|b_j|}}{\frac{P\Big\{u_{t+1}(b_j)=1|Z_{1:t}\Big\}}{|b_j|}} \cdot |b_j| + \Big(1-P_s\Big\{u_t(b_i)=1\Big\}\Big)\Bigg] \tag{5.4}$$

Note that the (hyper)volume $|b_j|$ is inserted in four places such that they cancel each other. In the limiting case (i.e., as bin (hyper)volumes $|b_i|, |b_j| \to 0$), bin probabilities approach the respective PHDs. That is,

$$\begin{aligned}
\frac{P\Big(u_t(b_i)=1|Z_{1:k}\Big)}{|b_i|} &\to D_{t|k}(\mathbf{x}_t|Z_{1:k}) \\
\frac{P\Big(u_t(b_i)=1|Z_{1:t}\Big)}{|b_i|} &\to D_{t|t}(\mathbf{x}_t|Z_{1:t}) \\
\frac{P\Big(u_{t+1}(b_j)=1|Z_{1:k}\Big)}{|b_j|} &\to D_{t+1|k}(\mathbf{x}_{t+1}|Z_{1:k}) \\
\frac{P\Big(u_{t+1}(b_j)=1|Z_{1:t}\Big)}{|b_j|} &\to D_{t+1|t}(\mathbf{x}_{t+1}|Z_{1:t})
\end{aligned}$$

where $D_{t|k}(\mathbf{x}_t|Z_{1:k})$ and $D_{t+1|k}(\mathbf{x}_{t+1}|Z_{1:k})$ are the smoothed PHD at time t and $t+1$, respectively, $D_{t|t}(\mathbf{x}_t|Z_{1:t})$ is the filtered PHD at time t, and $D_{t+1|t}(\mathbf{x}_{t+1}|Z_{1:t})$ is the predicted PHD at time $t+1$. In addition, the state transition probability

$\frac{P\left\{u_{t+1}(b_j)=1|u_t(b_i)=1\right\}}{|b_j|}$ approaches to state transition density $f_{t+1|t}(\mathbf{x}_{t+1}|\mathbf{x}_t)$, the survival probability $P_s\left\{u_t(b_i)=1\right\} \rightarrow P_s(\mathbf{x}_t)$, and the bin (hyper)volume $|b_j| \rightarrow d\mathbf{x}_{t+1}$. The substitution of these limiting values into (5.4) leads to the following continuous backward PHD smoothing equation:

$$D_{t|k}(\mathbf{x}_t|Z_{1:k}) = D_{t|t}(\mathbf{x}_t|Z_{1:t}) \times \left[P_s(\mathbf{x}_t)\int \frac{D_{t+1|k}(\mathbf{x}_{t+1}|Z_{1:k})f_{t+1|t}(\mathbf{x}_{t+1}|\mathbf{x}_t)}{D_{t+1|t}(\mathbf{x}_{t+1}|Z_{1:t})}d\mathbf{x}_{t+1} + (1-P_s(\mathbf{x}_t))\right] \quad (5.5)$$

The smoothed PHD at time t is function of the filtered PHD time t and the smoothed PHD at time $t+1$ as shown below.

$$\begin{array}{ccccccccccc} \cdots & \rightarrow & D_{t|t} & \xrightarrow{\text{prediction}} & D_{t+1|t} & \xrightarrow{\text{Update}} & D_{t+1|t+1} & \rightarrow & \cdots & \rightarrow & D_{k|k} \\ & & \searrow & & \swarrow & & \downarrow & & & & \downarrow \\ \cdots & \leftarrow & & D_{t|k} & & \xleftarrow{\text{smoothing}} & D_{t+1|k} & \leftarrow & \cdots & \leftarrow & D_{k|k} \end{array}$$

Here, the backward recursion is initialized with the filtering results at the present time k and stopped at time $k-L$, where L is the time lag of the smoothing algorithm. SMC implementation of this backward recursion is given in following section.

5.2 SMC Backward PHD Smoothing Iteration

In SMC PHD smoothing, the smoothed particle weights at time $k-L$ are evaluated from filter outputs $\left\{w_{t|t}^{(p)}, \mathbf{x}_t^{(p)}\right\}_{p=1}^{L_t}$ for $t = k-L, \ldots, k$ using backward iterations. From (5.5), the smoothed weights are computed as follows:

- For $t = k-1, \ldots, k-L$ and $p = 1, \ldots, L_t$

$$w_{t|k}^{(p)} = w_{t|t}^{(p)} \left[P_s\left(\mathbf{x}_t^{(p)}\right) \sum_{q=1}^{L_{t+1}} \frac{w_{t+1|k}^{(q)} p_{t+1|t}(\mathbf{x}_{t+1}^{(q)}|\mathbf{x}_t^{(p)})}{\mu_{t+1|t}^{(q)}} + \left(1 - P_s(\mathbf{x}_t^{(p)})\right) \right] \quad (5.6)$$

where

$$\mu_{t+1|t}^{(q)} = \gamma_{t+1}(\mathbf{x}_{t+1}^{(q)}) + \sum_{r=1}^{L_t} w_{t|t}^{(r)} \left\{ P_s(\mathbf{x}_t^{(r)}) p_{t+1|t}(\mathbf{x}_{t+1}^{(q)}|\mathbf{x}_t^{(r)}) + b_{t+1|t}(\mathbf{x}_{t+1}^{(q)}|\mathbf{x}_t^{(r)}) \right\}$$

At the end of the iteration, the particle set $\left\{ w_{k-L|k}^{(p)}, \mathbf{x}_{k-L}^{(p)} \right\}_{p=1}^{L_{k-L}}$ is used to find smoothing outputs: namely, smoothed expected number of targets and corresponding target state estimates.

5.2.1 Fast Smoothing Iteration

The iteration (5.6) requires $\mathcal{O}(L_t L_{t+1})$ operations to evaluate the smoothed weights. The bottleneck is due to evaluation of the sum of kernels, i.e.,

$$w_{t|k}^{(p)} = \sum_{q=1}^{L_{t+1}} \tilde{w}_{t+1|k}^{(q)} f_{t+1|t}(\mathbf{x}_{t+1}^{(q)}|\mathbf{x}_t^{(p)}) \quad (5.7)$$

The computational cost can be reduced using the fast method proposed in [52] that uses the N-body algorithm [34]. For this method, the transition model should be a similarity kernel defined on a metric space, that is $p_{t+1|t}(\mathbf{x}_{t+1}|\mathbf{x}_t) = \kappa(d(\mathbf{x}_{t+1}, \mathbf{x}_t))$, where $d(\cdot)$ denotes distance. The conversion of state space into a metric space is discussed in Section 5.3. This space can be partitioned into KD-tree [52], which groups the particles into subgroups in its leaf nodes. Having created the trees for particles at time t and $t+1$ with the pre-specified error tolerance, evaluation of

the above sum can be approximated by querying leaf nodes of the dual-tree instead of querying between particles [52]. This N-body simulation method requires only $\mathcal{O}(L_t \log L_{t+1})$ operations to evaluate approximate sum-kernel.

5.3 Simulation

In this section, the results of the simulation studies using the novel PHD smoothing algorithm are presented. In this study, a two-dimensional scenario with surveillance region of $[-200,\ 200] \times [-200,\ 200]$ is considered. The number of targets in the region is time-varying due to possible target appearance and disappearance at any time. Spontaneous target birth is assumed to follow a Poisson point process with intensity function $\gamma_k = 0.1\mathcal{N}(\cdot|\mathbf{x}_b, Q_b)$, where $\mathcal{N}(\cdot|\mathbf{x}_b, Q_b)$ denotes a normal distribution with mean $\mathbf{x}_b = [0\ 3\ 0\ -3]^T$ and covariance $Q_b = \text{diag}([10\ 1\ 10\ 1])$. The target state at time k, $\mathbf{x}_k = [x_k\ \dot{x}_k\ y_k\ \dot{y}_k]^T$, consists of position, $[x_k\ y_k]^T$, and velocity, $[\dot{x}_k\ y_k]^T$ of the target. Here, $[\cdot]^T$ represents the transpose of a matrix. Each existing target survives with probability $P_s = 0.95$, which is state-independent. The target state has the following linear-Gaussian target dynamics:

$$\mathbf{x}_k = F_k \mathbf{x}_{k-1} + \mathbf{v}_{k-1} \tag{5.8}$$

where the target transition matrix F_k is given by

$$F_k = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

and the process noise $\mathbf{v}_{k-1}$ has a normal distribution with mean zero and covariance matrix Q_{k-1}, which is given by

$$Q_{k-1} = \sigma^2 \begin{bmatrix} T^3/3 & T^2/2 & 0 & 0 \\ T^2/2 & T & 0 & 0 \\ 0 & 0 & T^3/3 & T^2/2 \\ 0 & 0 & T^2/2 & T \end{bmatrix}$$

Here, T is the sampling period and equal to 1. The process noise standard deviation, σ, is equal to 0.2. For this discretized continuous time kinematic model [7], Q_{k-1} is a full rank matrix, which is essential for computing the transition prior in the smoothing formula (5.6). As mentioned in Section 5.2.1, the target state needs to be converted such that the transition prior is defined on metric space for the fast smoothing method. The conversion is as follows:

$$\begin{aligned} p_{k|k-1}(\mathbf{x}_k|\mathbf{x}_{k-1}) &= \frac{1}{c_k} \exp\left(-\left(\mathbf{x}_k - F_k\mathbf{x}_{k-1}\right)^T Q_{k-1}^{-1}\left(\mathbf{x}_k - F_k\mathbf{x}_{k-1}\right)/2\right) \\ &= \frac{1}{c_k} \exp\left(-\|\mathbf{y}_k - \mathbf{y}_k^*\|^2/2\right) \end{aligned} \tag{5.9}$$

where Q_{k-1}^{-1} is the inverse of covariance matrix Q_{k-1}, $c_k = \sqrt{\det(2\pi Q_{k-1})}$, $\mathbf{y}_k = Q_{k-1}^{-1/2}\mathbf{x}_k$, $\mathbf{y}_k^* = Q_{k-1}^{-1/2}F_k\mathbf{x}_{k-1}$, $\|\mathbf{z}_1 - \mathbf{z}_2\|$ is the Euclidean distance between vectors $\mathbf{z}_1$ and $\mathbf{z}_2$, and $Q_{k-1}^{-1/2}$ is the Cholesky factor of Q_{k-1}^{-1}. The position ground truth of three tracks over 100 scans are displayed in Figure 5.1. The individual plots for x and y components of each track against time show the start and finish times of the tracks in Figure 5.2. The sensor is located at $[0 \ -100]^T$ and measures the range r_k and

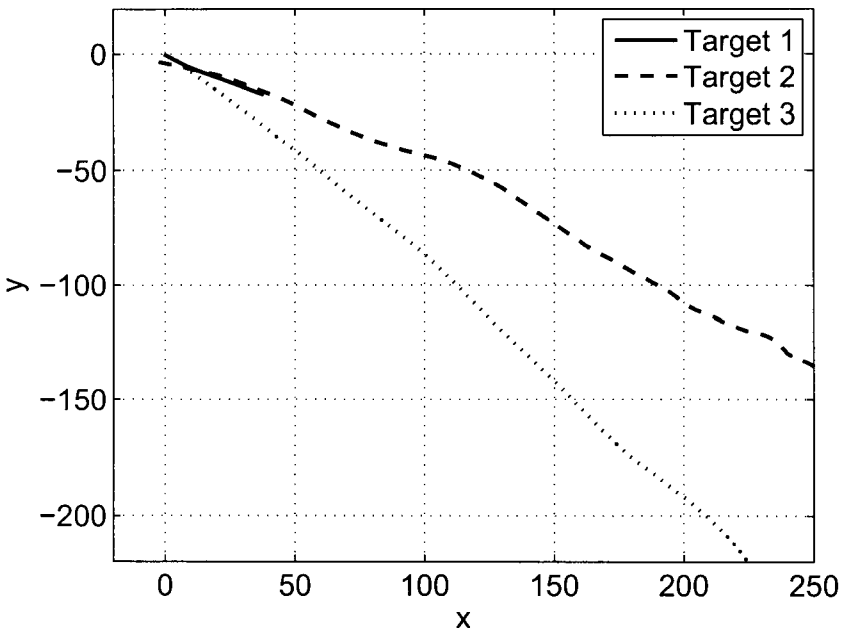


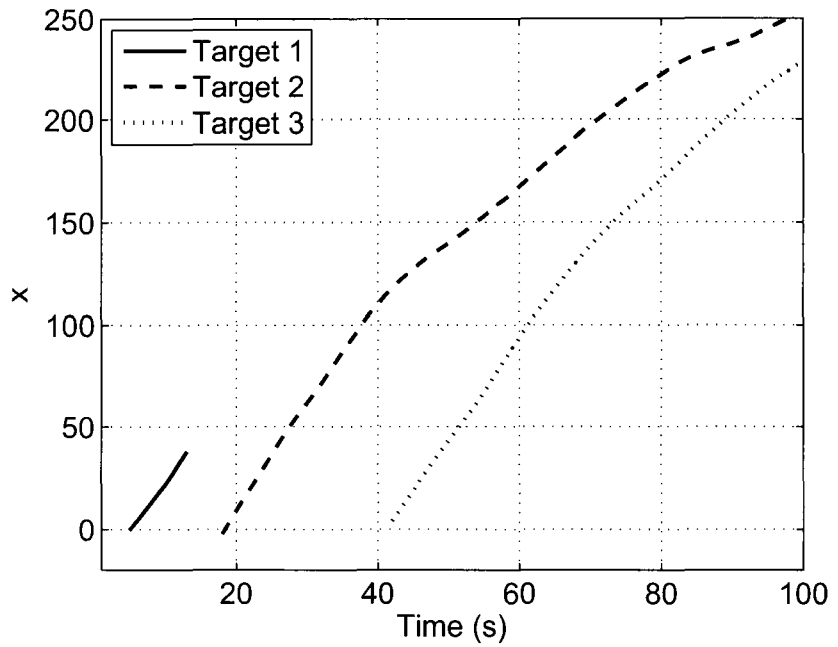
Figure 5.1: Ground truth: position plots of 3 true tracks

the bearing θ_k of a target that are related to the target state as follows:

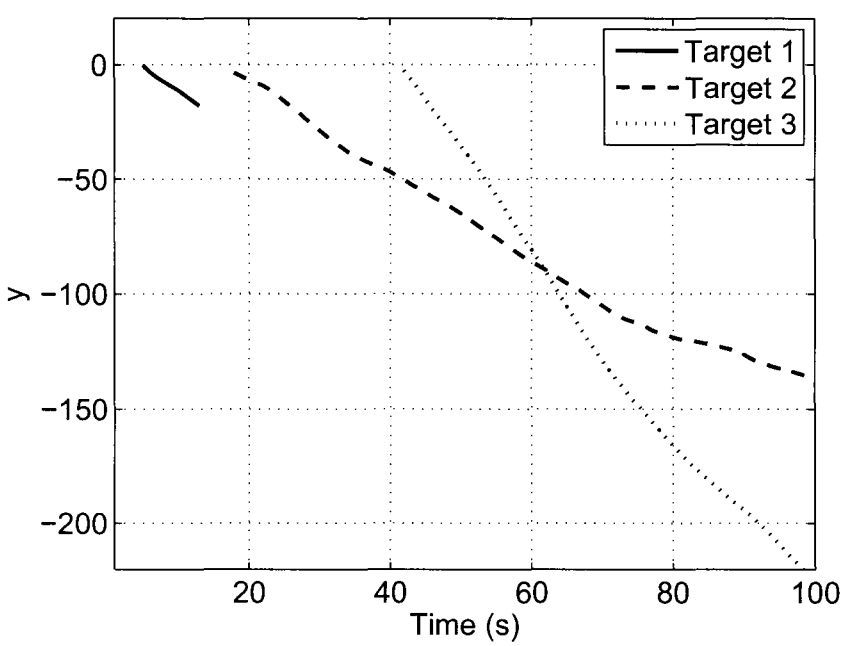
$$r_k = \left\| \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{x}_k - \begin{bmatrix} 0 \\ -100 \end{bmatrix} \right\| + w_{1,k} \tag{5.10}$$

$$\theta_k = \arctan\left(\frac{\begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \mathbf{x}_k}{\begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{x}_k + 100} \right) + w_{2,k} \tag{5.11}$$

where $w_{1,k}$ and $w_{2,k}$ are the independent zero-mean Gaussian noises with standard deviations 2 and 0.05, respectively. The above measurements are reported by the sensor with probability of detection $P_d = 0.9$, which is also state-independent. Clutter is uniformly distributed over the sensor field of view $[0\ 300] \times [-\pi/2\ \pi/2]$. The measurement set for a typical run is shown in Figure 5.3.



(a)



(b)

Figure 5.2: Ground truth: plots of x and y components of the 3 true tracks against time

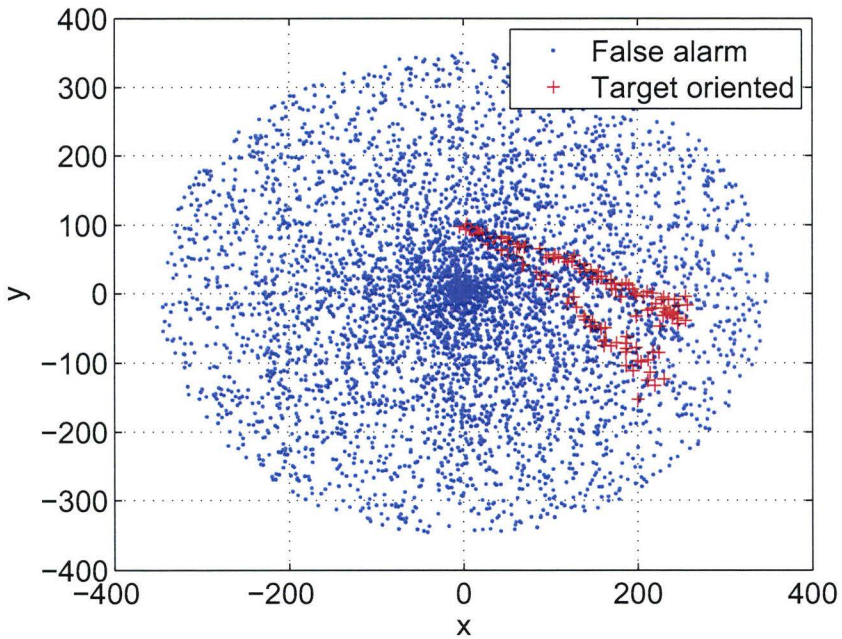


Figure 5.3: Measurements for a typical run

In this simulation, the smoothed states are obtained by backward PHD smoothing with different lags. The SMC method is implemented with 1000 particles per track, while tracks are initialized with 2000 particles. The extraction of point estimates from particle approximation is obtained using K-mean clustering method [36]. To demonstrate the improvement performance, Wasserstein distance [41] is used, which is a multitarget miss-distance generalizing the standard root mean square error of a single target problem to a multitarget problem. Results from 100 Monte Carlo runs are discussed below.

Figure 5.4 shows multitarget miss-distances of the estimates of filter and smoothers with different lags for the scenario with average rate of 20 cutter returns per scan. Similar results for the scenario with average rate of 50 clutter returns per scan are given in Figure 5.5. Both demonstrate improved performance of the smoothing method

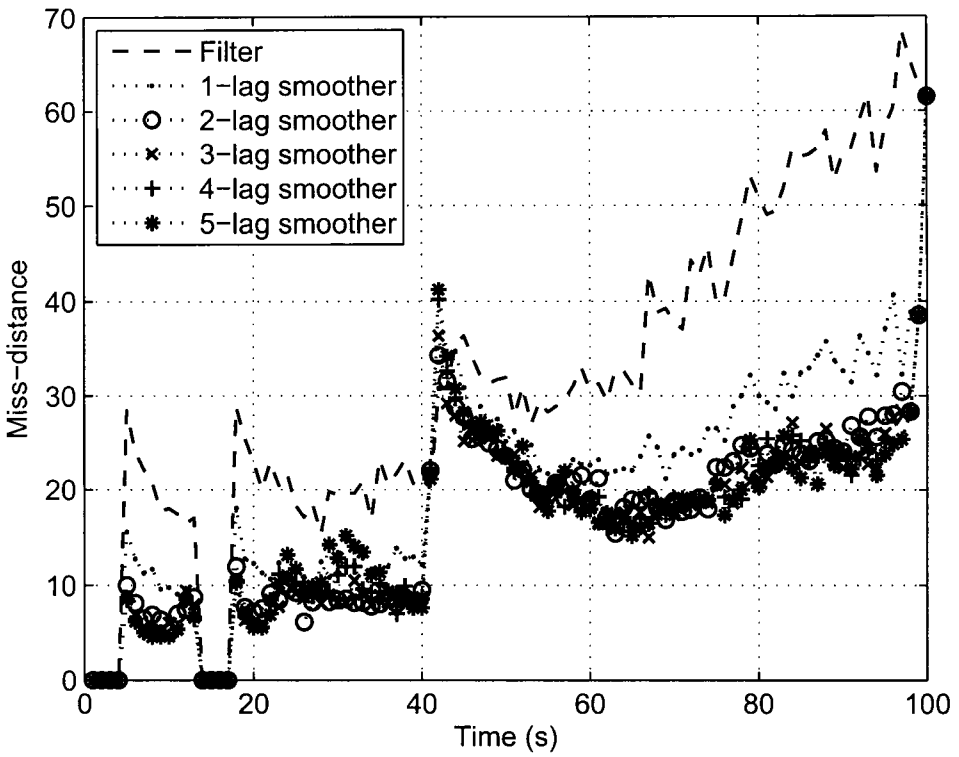


Figure 5.4: Multitarget miss-distance (The average rate of clutter returns per scan = 20)

over the filter. Increasing the clutter density degrades the performance of the filter, and hence that of the smoother. It can also be observed that the higher the lag we use for smoothing the better the estimates we obtain. However, this tendency saturates after the lag of 3. This suggests that the future measurements with long lag do not improve the current state as the information from them becomes irrelevant with lag. The improved performance of the smoothing method is achieved at the cost of additional computational load (i.e., the proposed smoothing method with the lag of 3 requires 1.95 seconds while the standard PHD filter requires 1.31 seconds for a single step of MATLAB implementation on a Pentium IV 3 GHz processor).

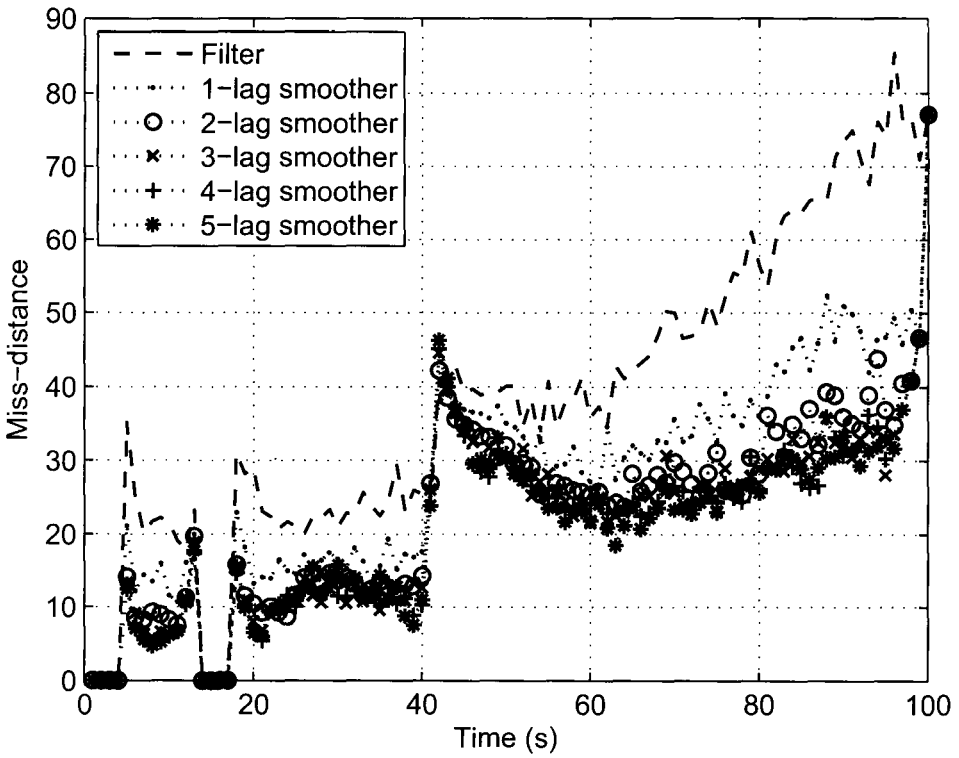


Figure 5.5: Multitarget miss-distance (The average rate of clutter returns per scan = 50)

Chapter 6

MMPHD Smoothing

In maneuvering target tracking problems, detection and tracking of changes in the target motion model are crucial to get accurate state estimates. For such problems, multiple model approaches have been shown to be highly effective. A PHD-based multiple model approach for multitarget tracking was proposed in [77]. This chapter provides a smoothing algorithm for the MMPHD approach, which is a natural extension of PHD smoothing algorithm for maneuvering targets. In Section 6.1, MMPHD filter is reviewed. The smoothing algorithm for maneuvering targets is given in Section 6.2. The particle implementation of MMPHD filter is provided in Section 6.3, while particle implementation of novel MMPHD smoothing is developed in Section 6.4. In Section 6.5, simulation studies are provided.

6.1 MMPHD Filter

In MMPHD filtering, the mode-dependent densities $D_{k|k}(\mathbf{x}_k, r_k = u|Z_{1:k}), \quad u = 1, \ldots, N_r$ are propagated recursively through three steps: namely, mixing, prediction and update [77]. Here, N_r is the number of models and r_k is the model index

parameter governed by an underlying Markov process with the model transition probability

$$p_{k|k-1}(r_k = u|r_{k-1} = v) = h_{vu} \tag{6.1}$$

In the mixing stage, each mode-matched filter is fed with a different density that is a combination of the previous mode-dependent densities. The initial density $\tilde{D}_{k|k-1}(\mathbf{x}_{k-1}, r_k = u|Z_{1:k-1})$ fed to the PHD filter, which is matched to the target model u, is calculated on the basis of Markovian model transition probability matrix $[h_{vu}]$ and mode-dependent prior $D_{k-1|k-1}(\mathbf{x}_{k-1}, r_{k-1} = v|Z_{1:k-1})$. That is, for $u = 1, \ldots, N_r$

$$\tilde{D}_{k|k-1}(\mathbf{x}_{k-1}, r_k = u|Z_{1:k-1}) = \sum_{v=1}^{N_r} D_{k-1|k-1}(\mathbf{x}_{k-1}, r_{k-1} = v|Z_{1:k-1})h_{vu} \tag{6.2}$$

In the prediction stage, having obtained the initial density for the PHD filter that is matched to the model u, the mode-dependent density is calculated as

$$\begin{aligned} D_{k|k-1}(\mathbf{x}_k, r_k = u|Z_{1:k-1}) &= \gamma_k(\mathbf{x}_k, r_k = u) \\ &\quad + \int \left[P_s(\mathbf{x}_{k-1})p_{k|k-1}(\mathbf{x}_k|\mathbf{x}_{k-1}, r_k = u)\right. \\ &\quad \left. + b_{k|k-1}(\mathbf{x}_k|\mathbf{x}_{k-1}, r_k = u)\right] \\ &\quad \times \tilde{D}_{k|k-1}(\mathbf{x}_{k-1}, r_k = u|Z_{1:k-1})d\mathbf{x}_{k-1} \end{aligned} \tag{6.3}$$

Finally, with the availability of measurements, the update stage results in the mode-dependent updated density is given by

$$D_{k|k}(\mathbf{x}_k, r_k = u|Z_{1:k}) \cong \left[\sum_{\mathbf{z}_k \in Z_k} \frac{P_d(\mathbf{x}_k)p_k(\mathbf{z}_k|\mathbf{x}_k, r_k = u)}{\lambda_k c_k(\mathbf{z}_k) + \psi_k(\mathbf{z}_k|Z_{k-1})} + (1 - P_d(\mathbf{x}_k))\right] \times D_{k|k-1}(\mathbf{x}_k, r_k = u|Z_{1:k-1}) \tag{6.4}$$

where the likelihood function $\psi(\cdot)$ is given by

$$\psi_k(\mathbf{z}_k|Z_{k-1}) = \int P_d(\mathbf{x}_k)p_k(\mathbf{z}_k|\mathbf{x}_k, r_k = u)D_{k|k-1}(\mathbf{x}_k, r_k = u|Z_{1:k-1})d\mathbf{x}_k \tag{6.5}$$

This update stage implicitly incorporates the mode probability update. The expected number of targets can be found by summing up all the integrals of the updated mode-dependent PHDs.

6.2 MMPHD Smoother

Having reviewed the MMPHD filter in Section 6.1, the PHD smoothing formula can now be extended (5.5) for maneuvering targets. The smoothed mode-dependent density can be found with following backward recursion

$$\begin{aligned} &D_{t|k}(\mathbf{x}_t, r_t = u|Z_{1:k}) = \\ &\quad D_{t|t}(\mathbf{x}_t, r_t = u|Z_{1:t}) \\ &\quad \times \left[P_s(\mathbf{x}_t) \int \sum_{v=1}^{N_r} \frac{D_{t+1|k}(\mathbf{x}_{t+1}, r_{t+1} = v|Z_{1:k})p_{t+1|t}(\mathbf{x}_{t+1}|\mathbf{x}_t, r_t = u)h_{vu}}{D_{t+1|t}(\mathbf{x}_{t+1}, r_{t+1} = v|Z_{1:t})} d\mathbf{x}_{t+1}\right. \\ &\qquad \left. +(1 - P_s(\mathbf{x}_t))\right] \end{aligned} \tag{6.6}$$

where the normalizing constant, $D_{t+1|t}(\mathbf{x}_{t+1}, r_{t+1} = v|Z_{1:t})$, is given by

$$\begin{aligned} &D_{t+1|t}(\mathbf{x}_{t+1}, r_{t+1} = v|Z_{1:t}) = \\ &\quad \gamma_{t+1}(\mathbf{x}_{t+1}, r_{t+1} = v) + \int \sum_{u=1}^{N_r} \left[P_s(\mathbf{x}_t) p_{t+1|t}(\mathbf{x}_{t+1}|\mathbf{x}_t, r_t = u)\right. \\ &\quad \left. + b_{t+1|t}(\mathbf{x}_{t+1}|\mathbf{x}_t, r_t = u)\right] h_{vu} D_{t|t}(\mathbf{x}_t, r_t = u|Z_{1:t}) d\mathbf{x}_t \end{aligned} \tag{6.7}$$

6.3 SMC MMPHD Filter

In the SMC implementation of the MMPHD filter [77], the PHD surface is represented by a set of particles $\left\{w_{k-1|k-1}^{(j)}, \mathbf{x}_{k-1}^{(j)}, r_{k-1}^{j}\right\}_{j=1}^{L_{k-1}}$, where $\mathbf{x}_{k-1}^{(j)}$ and r_{k-1}^{j} are the target state and the target mode of the jth particle, $w_{k-1|k-1}^{(j)}$ is the associated weight and L_{k-1} is the number of particles. Therefore the PHD at time $k-1$ can be written as

$$D_{k-1|k-1}\left(\mathbf{x}_{k-1}, r_{k-1}|Z_{1:k-1}\right) = \sum_{j=1}^{L_{k-1}} w_{k-1|k-1}^{(j)} \delta\left(\mathbf{x}_{k-1} - \mathbf{x}_{k-1}^{(j)}, r_{k-1} - r_{k-1}^{(j)}\right) \tag{6.8}$$

where $\delta(\cdot)$ is the Dirac Delta function. The recursion at time k is summarized as follows:

- Mixing or model prediction: For $j = 1, \ldots, L_{k-1}$

$$\begin{aligned} r_{k|k-1}^{(j)} &\sim \pi_k\left(r_k|r_{k-1}^{(j)}\right) \\ \tilde{w}_{k|k-1}^{(j)} &= \frac{f_{k|k-1}\left(r_{k|k-1}^{(j)}|r_{k-1}^{(j)}\right)}{\pi_k\left(r_{k|k-1}^{(j)}|r_{k-1}^{(j)}\right)} w_{k-1|k-1}^{(j)} \end{aligned}$$

  where $\pi_k(r_k|r_{k-1})$ is the proposal mass function for mode transition.

- Prediction:

– Samples for existing and spawning targets: For $j = 1, \ldots, L_{k-1}$,

$$\begin{aligned}
\mathbf{x}_{k|k-1}^{(j)} &\sim q_k\left(\mathbf{x}_k|\mathbf{x}_{k-1}^{(j)}, r_{k|k-1}^{(j)}, Z_k\right) \\
w_{k|k-1}^{(j)} &= \frac{P_s\left(\mathbf{x}_{k|k-1}^{(j)}\right) f_{k|k-1}\left(\mathbf{x}_{k|k-1}^{(j)}|\mathbf{x}_{k-1}^{(j)}, r_{k|k-1}^{(j)}\right) + b_{k|k-1}\left(\mathbf{x}_{k|k-1}^{(j)}|\mathbf{x}_{k-1}^{(j)}, r_{k|k-1}^{(j)}\right)}{q_k\left(\mathbf{x}_{k|k-1}^{(j)}|\mathbf{x}_{k-1}^{(j)}, r_{k|k-1}^{(j)}, Z_k\right)} \tilde{w}_{k|k-1}^{(j)}
\end{aligned}$$

– Samples for new-born targets: For $j = L_{k-1} + 1, \ldots, L_{k-1} + J_k$,

$$\begin{aligned}
r_{k|k-1}^{(j)} &\sim \beta_k(r_k) \\
\mathbf{x}_{k|k-1}^{(j)} &\sim p_k\left(\mathbf{x}_k|r_{k|k-1}^{(j)}, Z_k\right) \\
w_{k|k-1}^{(j)} &= \frac{\gamma_k(\mathbf{x}_{k|k-1}^{(j)})\theta_k\left(r_{k|k-1}^{(j)}\right)}{p_k\left(\mathbf{x}_{k|k-1}^{(j)}|Z_k\right)\beta_k\left(r_{k|k-1}^{(j)}\right)} \frac{1}{J_k}
\end{aligned}$$

where $\theta_k(\cdot)$ and $\theta_k(\cdot)$ are state density and model mass function for target birth, while $p_k(\cdot)$ and $\beta_k(\cdot)$ are corresponding proposal density and mass function.

- Update:

– updated particle weights for $j = 1, \ldots, L_{k-1} + J_k$

$$w_{k|k}^{*(j)} = \left[\left(1 - P_d(\mathbf{x}_{k|k-1}^{(j)})\right) + \sum_{i=1}^{N_k^Z} \frac{P_d(\mathbf{x}_{k|k-1}^{(j)}) f_k\left(\mathbf{z}_k^i|\mathbf{x}_{k|k-1}^{(j)}, r_{k|k-1}^{(j)}\right)}{\lambda_k c_k(\mathbf{z}_k^i) + \Psi_k(\mathbf{z}_k^i)}\right] w_{k|k-1}^{(j)}$$

where

$$\Psi_k(\mathbf{z}_k^i) = \sum_{s=1}^{L_{k-1}+J_k} P_d(\mathbf{x}_{k|k-1}^{(s)}) f_k\left(\mathbf{z}_k^i|\mathbf{x}_{k|k-1}^{(s)}, r_{k|k-1}^{(s)}\right) w_{k|k-1}^{(s)}$$

- Resampling:

  - Find the expected number of targets

$$\hat{N}_{k|k} = \sum_{j=1}^{L_{k-1}+J_k} w_{k|k}^{*(j)}$$

  - Resample $\left\{w_{k|k}^{*(j)}/n_{k|k}, \mathbf{x}_{k|k-1}^{(j)}\right\}_{j=1}^{L_{k-1}+J_k}$ to get $\left\{w_{k|k}^{(j)}/\hat{N}_{k|k}, \mathbf{x}_{k|k}^{(j)}\right\}_{j=1}^{L_k}$

At the end of the recursion, the estimate of the number of targets is given by $\hat{N}_k = \text{round}(\hat{N}_{k|k})$. Then the state estimates of $\hat{N}_k$ targets can be extracted from the particles approximating the posterior PHD function from suitable clustering techniques or EM algorithm.

6.4 SMC MMPHD Smoother

In SMC MMPHD smoothing, the smoothed particle weights at time $k-L$ are evaluated using backward iterations using the filter outputs $\left\{w_{t|t}^{(j)}, \mathbf{x}_t^{(j)}, r_t^{(j)}\right\}_{j=1}^{L_t}$ $t = k-L, \ldots, k$. The smoothed weights are computed as follows:

- For $l = k-1, \ldots, k-L$, $j = 1, \ldots, L_t$

$$w_{t|k}^{(j)} = w_{t|t}^{(j)} \left[P_s(\mathbf{x}_t^{(j)}) \sum_{i=1}^{L_{t+1}} \frac{w_{t+1|k}^{(i)} p_{t+1|t}(\mathbf{x}_{t+1}^{(i)}|\mathbf{x}_t^{(j)}) h_{r_{t+1}^{(i)} r_t^{(j)}}}{\mu_{t+1|t}^{(i)}} + \left(1 - P_s(\mathbf{x}_t^{(j)})\right) \right] \quad (6.9)$$

where

$$\mu_{t+1|t}^{(i)} = \gamma_{t+1}(\mathbf{x}_{t+1}^{(i)}) + \sum_{l=1}^{L_t} w_{t|t}^{(l)} h_{r_{t+1}^{(i)} r_t^{(l)}} \left[P_s(\mathbf{x}_t^{(l)}) f_{t+1|t}(\mathbf{x}_{t+1}^{(i)}|\mathbf{x}_t^{(l)}) + b_{t+1|t}(\mathbf{x}_{t+1}^{(i)}|\mathbf{x}_t^{(l)}) \right]$$

At the end of the iteration, the particle set $\left\{w_{k-L|k}^{(j)}, \mathbf{x}_{k-L}^{(j)}, r_{k-L}^{(j)}\right\}_{j=1}^{L_{k-L}}$ is used to find smoothing outputs: namely, smoothed expected number of targets, corresponding target state estimates and modes. In this case, the fast method discussed in Section 5.2.1 cannot be used since the new sum of kernels has discrete density h_{vu} given by

$$w^{(j)} = \sum_{i=1}^{N} \tilde{w}^{(i)} f(\mathbf{x}_{t+1}^{(i)}|\mathbf{x}_t^{(j)}) h_{r_{t+1}^{(i)} r_t^{(j)}} \tag{6.10}$$

However, for a special case where the targets have two models with symmetric transition probabilities, the fast method can still be used. That is, the model transition matrix has the form

$$[h_{vu}] = \begin{bmatrix} a & 1-a \\ 1-a & a \end{bmatrix} \tag{6.11}$$

After converting $f(\cdot)$ into metric Gaussian kernel, also discussed in Section 5.3, the sum of kernels in (6.10) can be rewritten as follows:

$$w^{(p)} = \sum_{q=1}^{N} \tilde{w}^{(q)} \frac{1}{\tilde{c}} \exp\left(-\|\tilde{\mathbf{y}}_{t+1} - \tilde{\mathbf{y}}_{t+1}^{*}\|^2/2\right) \tag{6.12}$$

where $\tilde{\mathbf{y}}_{t+1} = [\mathbf{y}_{t+1}^T \ b_{t+1}]^T$, $\tilde{\mathbf{y}}_{t+1}^{*} = [\mathbf{y}_{t+1}^{*T} \ b_t]^T$, $\mathbf{y}_{t+1}$ and $\mathbf{y}_{t+1}^{*}$ are the states in the metric space corresponding to $\mathbf{x}_{t+1}$ and $\mathbf{x}_t$. Here, b_t is given by

$$b_t = \begin{cases} 0 & u = 1 \\ \sqrt{2\ln(a/1-a)} & u = 2 \end{cases} \tag{6.13}$$

Now, the sum of kernels in (6.12) is in the required metric space, hence the fast method can be used.

6.5 Simulation

In the simulation study, we consider a two-dimensional scenario with two maneuvering targets, namely, target 1 and target 2. With initial position at (25, 10) km, target 1 moves westward for 50s at a nearly constant velocity with velocity of 320 ms^{-1}, before executing a 1.8^0/s coordinated turn in the counter-clockwise for 50s. Then it moves southward for another 50s, followed by a clockwise 1.8^0/s coordinated turn for 50s. Target 2, starting from (25, -25) km with initial velocity of 300 ms^{-1}, executes a coordinated turn of 1.8^0/s in the counter-clockwise for 50s, followed by a nearly constant velocity motion for 50s. Then it executes a clockwise coordinated turn of 1.8^0/s, followed by a nearly constant velocity motion for 50s. The target trajectories are shown in Figure 6.1. The sensor is located at the origin, and provides range and bearing measurements with measurement noise standard deviations of 100m and 0.02 rad. The measurements are available at discrete sampling interval $T = 5$s with probability of detection $P_d = 0.9$. Clutter is uniformly distributed over sensor field of view [0 30] km $\times$ $[-\pi\ \pi]$ rad with average rate of 8 returns per scan. The MMPHD filter consists of two models with symmetric transition probability matrix as follows:

$$[h_{vu}] = \begin{bmatrix} 1 - \frac{T}{\tau} & \frac{T}{\tau} \\ \frac{T}{\tau} & 1 - \frac{T}{\tau} \end{bmatrix} \tag{6.14}$$

Here, the sojourn time $\tau = 100$s. The first model is a constant velocity with $\sigma^2 = 1\ \text{m}^2\text{s}^{-3}$. That is,

$$\mathbf{x}_k = F_{1,k}\mathbf{x}_{k-1} + \mathbf{v}_{1,k-1} \tag{6.15}$$

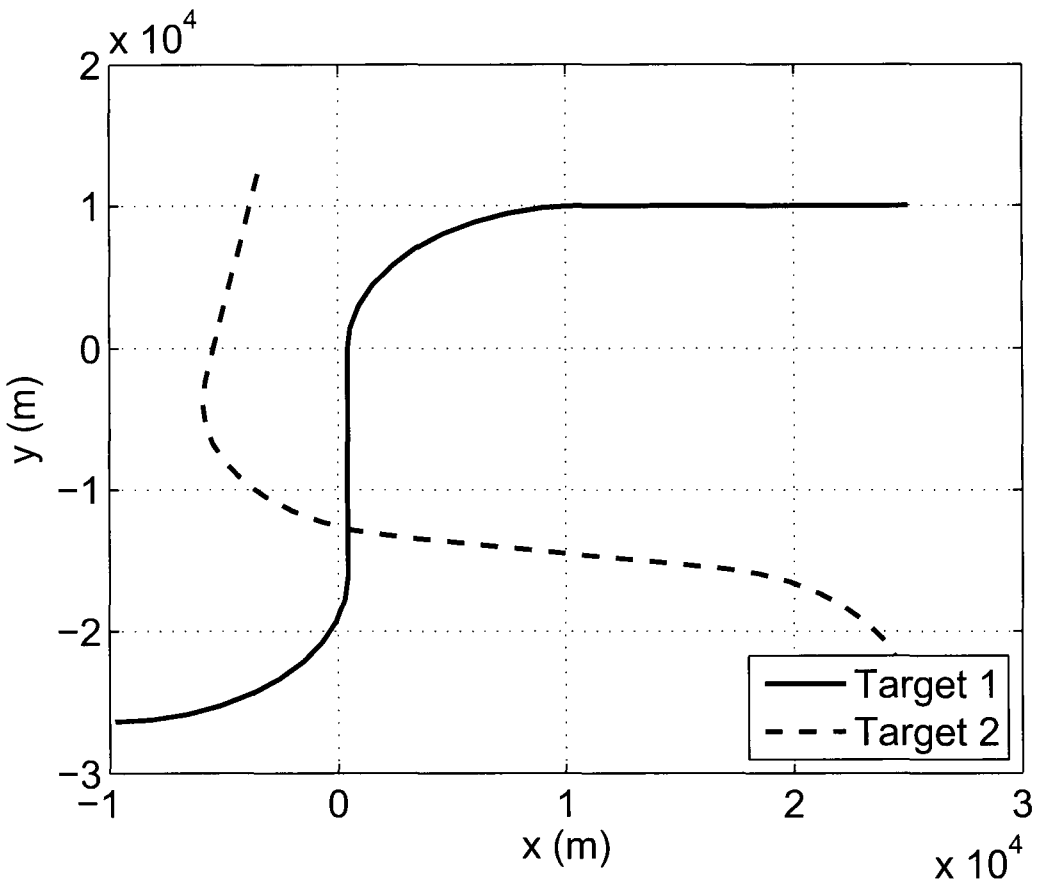


Figure 6.1: Ground truth of the maneuvering target scenario

where $\mathbf{x}_k = [x_k \ \dot{x}_k \ y_k \ \dot{y}_k]^T$ and the target transition matrix $F_{1,k}$ is given by

$$F_{1,k} = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

and the process noise $\mathbf{v}_{1,k-1}$ has a normal distribution with mean zero and covariance matrix $Q_{1,k-1}$, which is given by

$$Q_{1,k-1} = \sigma^2 \begin{bmatrix} T^3/3 & T^2/2 & 0 & 0 \\ T^2/2 & T & 0 & 0 \\ 0 & 0 & T^3/3 & T^2/2 \\ 0 & 0 & T^2/2 & T \end{bmatrix}$$

The second one is coordinated turn model with variable turn rate and given by

$$\mathbf{x}_k = F_{2,k}\mathbf{x}_{k-1} + \mathbf{v}_{2,k-1} \tag{6.16}$$

where $\mathbf{x}_k = [x_k \ \dot{x}_k \ y_k \ \dot{y}_k \ \Omega_k]^T$ is the augmented state vector, which consists of target position $[x_k, \ y_k]^T$, target velocity $[\dot{x}_k, \dot{y}_k]^T$ and target turn rate Ω_k at time step, and

target transition matrix $F_{2,k}$ is given by

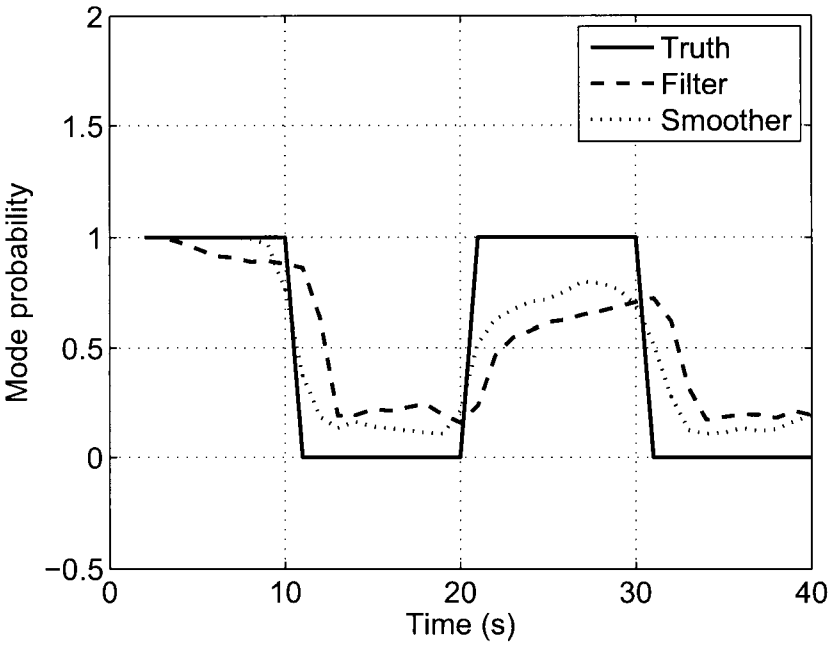
$$F_{2,k} = \begin{bmatrix} 1 & \frac{\sin \Omega_{k-1} T}{\Omega_{k-1}} & 0 & -\frac{1-\cos \Omega_{k-1} T}{\Omega_{k-1}} & 0 \\ 0 & \cos \Omega_{k-1} T & 0 & -\sin \Omega_{k-1} T & 0 \\ 0 & \frac{1-\cos \Omega_{k-1} T}{\Omega_{k-1}} & 1 & \frac{\sin \Omega_{k-1} T}{\Omega_{k-1}} & 0 \\ 0 & \sin \Omega_{k-1} T & 0 & \cos \Omega_{k-1} T & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

and the process noise $\mathbf{v}_{2,k-1}$ has a normal distribution with mean zero and covariance matrix $Q_{2,k-1}$, which is given by

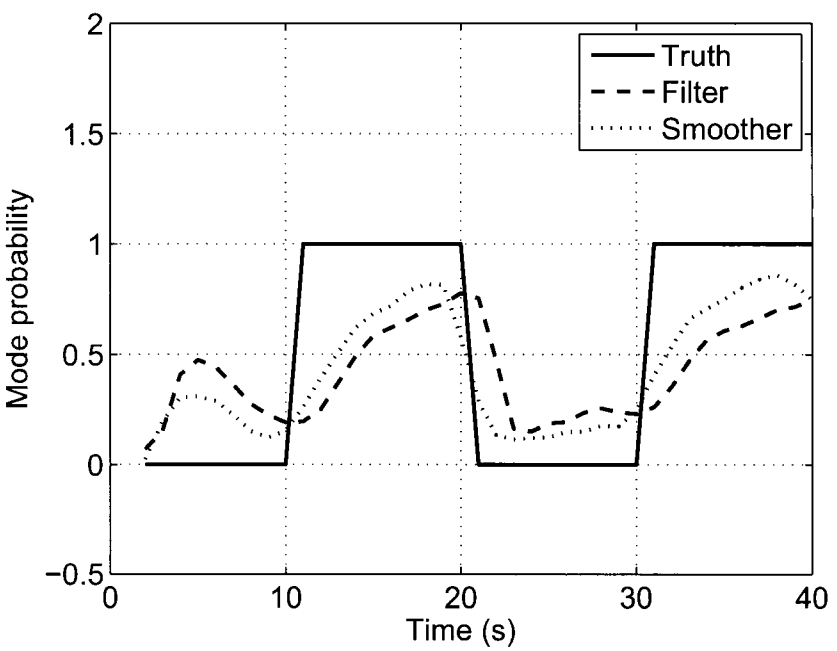
$$Q_{2,k-1} = \begin{bmatrix} \sigma^2 T^3/3 & \sigma^2 T^2/2 & 0 & 0 & 0 \\ \sigma^2 T^2/2 & \sigma^2 T & 0 & 0 & 0 \\ 0 & 0 & \sigma^2 T^3/3 & \sigma^2 T^2/2 & 0 \\ 0 & 0 & \sigma^2 T^2/2 & \sigma^2 T & 0 \\ 0 & 0 & 0 & 0 & \sigma_\Omega^2 T \end{bmatrix}$$

where $\sigma^2 = 1$ m^{2}s^{-3} and $\sigma_\Omega^2 = 1 \times 10^{-4}$ rad^{2}s^{-1}

Results from 100 Monte Carlo runs are discussed below. Figure 6.2 shows the model-switching property of the MMPHD filter and smoother for both targets. It clearly demonstrates improved model-switching of MMPHD smoother over the MMPHD filter. The RMSE for position, velocity and turn rate are provided in Figure 6.3, 6.4 and 6.5 respectively. The comparisons of these RMSE values also confirm the improved performance of the smoother over the filter.

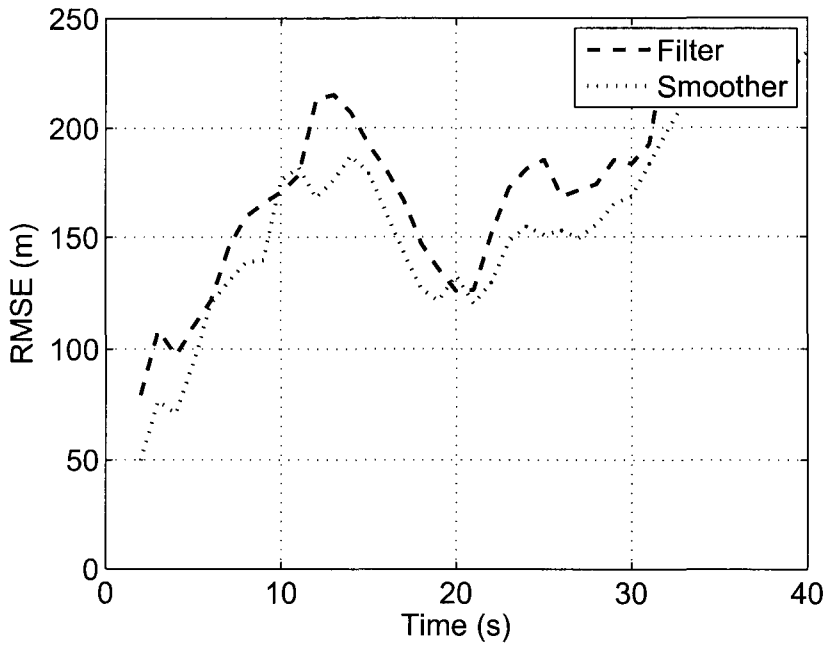


(a) Target 1

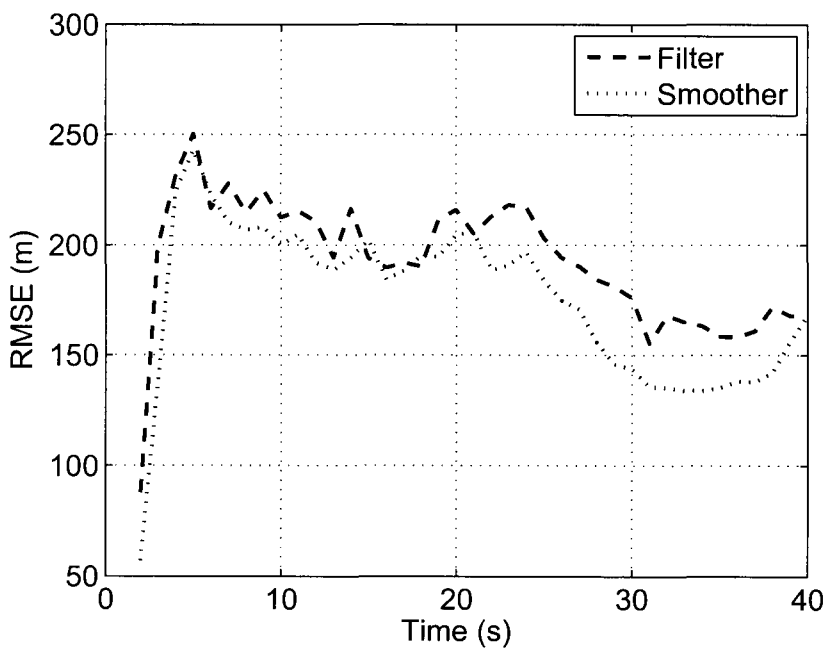


(b) Target 2

Figure 6.2: Model switching

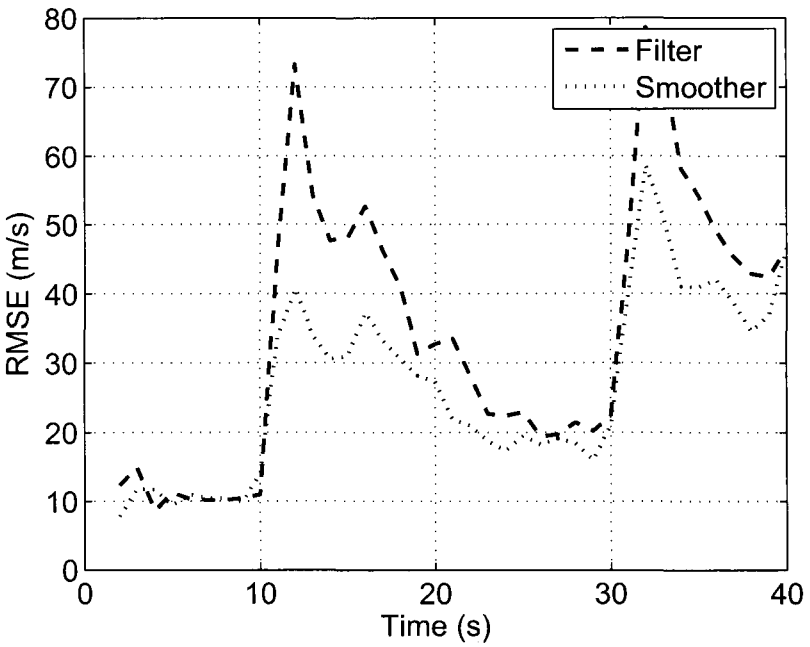


(a) Target 1

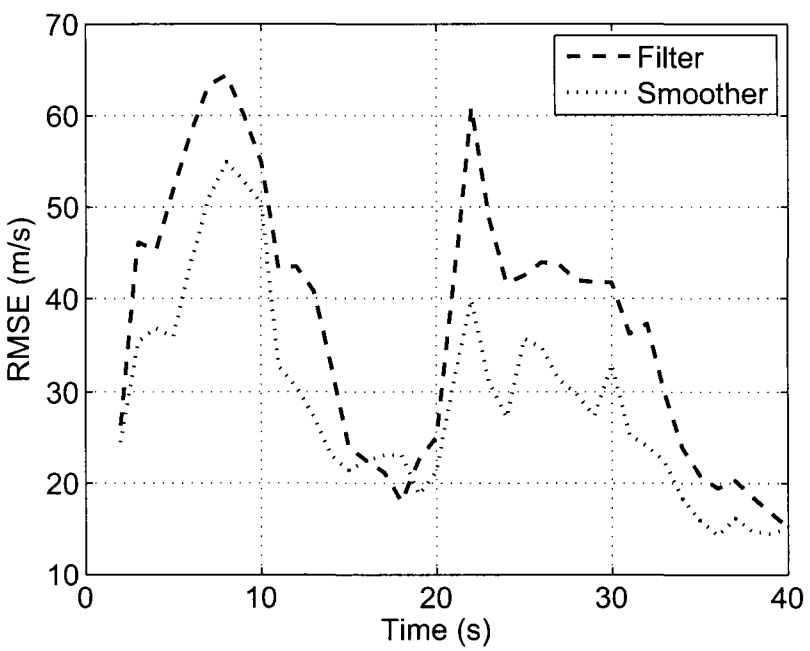


(b) Target 2

Figure 6.3: Position RMSE

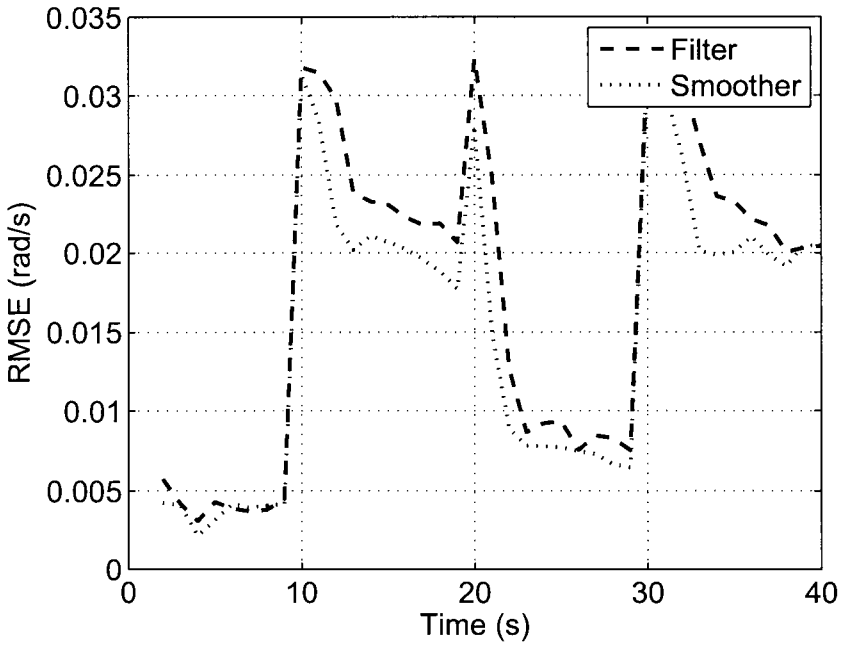


(a) Target 1

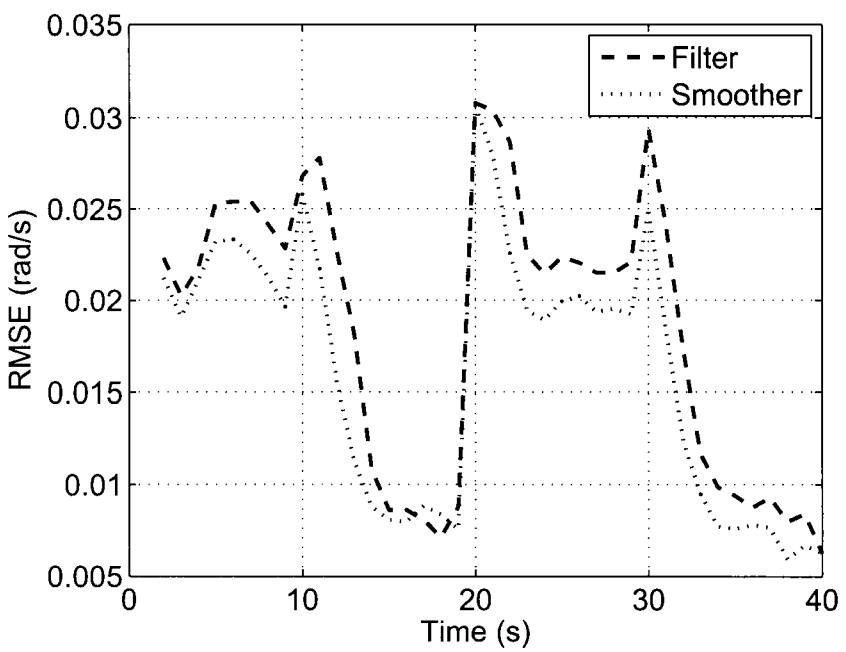


(b) Target 2

Figure 6.4: Velocity RMSE



(a) Target 1



(b) Target 2

Figure 6.5: Turn rate RMSE

Chapter 7

GMPHD Smoothing

In this chapter, we propose a smoothing method for GMPHD state estimator using multiple sensors. In the case of multisensor systems, a sequential update of the PHD has been suggested in the literature. The posterior is sequentially updated using measurements from one sensor at a time, while assuming that the posterior updated by the previous sensor is approximately Poisson [63]. In Section 7.4, we demonstrate that this sequential update is susceptible to the imperfections in the last sensor, which was also observed in [26]. To mitigate this behavior of the PHD filter, we propose different strategies to improve the GMPHD filter using measurements from multiple sensors simultaneously: namely, the all-sequence update and the parallel update (Section 7.1). In all-sequence update, we get average of the posterior PHD surfaces resulting from all possible sequences of update by considering permutation of the order of update. In the parallel update, the posterior PHD surface is found by averaging the updated posterior PHD surfaces from all sensors. The resulting filter outputs are further improved using a backward smoothing (Section 7.2). An approximate closed-form solution is found for the backward smoothing recursion using Gaussian

mixture implementation. Resulting algorithm provides delayed but better estimates for the target states. Further, we propose a Gaussian mixture implementation of the smoothing algorithm for the Multiple Model PHD (MMPHD) approach (Section 6.2), which is a natural extension of PHD smoothing algorithm for maneuvering targets. Improved performance is demonstrated using simulation studies with multiple bearing only sensors in Section 7.4.

7.1 Multisensor PHD Update

In the case of multisensor systems, the following sequential update was suggested in [63]

$$\begin{aligned} D^1_{k|k}(\mathbf{x}_k|Z_{1:k}) &\cong \Psi_k^{[j_1]} D_{k|k-1}(\mathbf{x}_k|Z_{1:k-1}) \qquad (7.1) \\ &\vdots \\ D^i_{k|k}(\mathbf{x}_k|Z_{1:k}) &\cong \Psi_k^{[j_i]} D^{i-1}_{k|k}(\mathbf{x}_k|Z_{1:k}) \\ &\vdots \\ D^{\chi_j}_{k|k}(\mathbf{x}_k|Z_{1:k}) &\cong \Psi_k^{[j_S]} D^{S-1}_{k|k}(\mathbf{x}_k|Z_{1:k}) \qquad (7.2) \end{aligned}$$

where S is the number of sensors, $\chi_j = \{j_1, \ldots, j_S\}$ is a sequence of sensor indices, and $\Psi_k^{[j_i]}$ is the update operator (Section 3.2.2) of the j_ith sensor. In each step of the above update (7.1)–(7.2), it is assumed that the posterior updated by previous sensor is approximately Poisson [63]. As demonstrated in Section 7.4.1, these approximations make the performance of the PHD filter sensitive to the order of updates. That is, it is shown that different update sequences result in different filter results. By considering permutations of the sensor index sequence, the end result will be $S!$ different PHD

surfaces. To improve the PHD filter with multiple sensor, the average of all possible surfaces can be obtained. That is,

$$D_{k|k}(\mathbf{x}_k|Z_{1:k}) = \frac{1}{S!}\sum_{j=1}^{S!} D_{k|k}^{\chi_j}(\mathbf{x}_k|Z_{1:k}) \tag{7.3}$$

Hereafter, this update will be known as all-sequence update, which provides better results than the sequential update in [63]. However, it is computationally demanding for problems with a large number of sensors. Using a similar argument, the average of the PHD surfaces obtained by individual sensor updates from the predicted PHD can be considered. That is, the proposed parallel update is given by

$$D_{k|k}(\mathbf{x}_k|Z_{1:k}) = \frac{1}{S}\sum_{i=1}^{S} \Psi_k^{[i]} D_{k|k-1}(\mathbf{x}_k|Z_{1:k-1}) \tag{7.4}$$

It will be shown in Section 7.4 that the performance of this method is similar to that of the all-sequence method, while its computational load is similar to that of the sequential method of [63].

For the Gaussian mixture implementation, the update procedure in (3.35) is repeatedly applied with measurements from each sensor according to the sequential update step (7.1)–(7.2). For all-sequence update in (7.3), the updated mixtures of all possible sequences are combined and each component is reweighted by a factor of $\frac{1}{S!}$. For the parallel update in (7.4), the updated mixtures from all sensors are combined with reweighting by a factor of $\frac{1}{S}$.

7.2 Gaussian Mixture PHD Smoothing

In this section, a backward smoothing recursion for PHD-based state estimator using Gaussian mixture implementation is proposed. An approximate closed-form solution is derived for the backward recursion. In the backward smoothing recursion at time t $(k - L \leq t < k)$, we evaluate the smoothed PHD at time t using the filtered PHD at time t and the smoothed PHD at time $t + 1$, which are Gaussian mixtures. That is, they are given by

$$D_{t|t}(\mathbf{x}_t|Z_{1:t}) = \sum_{i=1}^{J_{t|t}} w_{t|t}^i \mathcal{N}\left(\mathbf{x}_t; \mathbf{m}_{t|t}^i, P_{t|t}^i\right) \tag{7.5}$$

$$D_{t+1|k}(\mathbf{x}_{t+1}|Z_{1:k}) = \sum_{i=1}^{J_{t+1|k}} w_{t+1|k}^i \mathcal{N}\left(\mathbf{x}_{t+1}; \mathbf{m}_{t+1|k}^i, P_{t+1|k}^i\right) \tag{7.6}$$

The smoothed PHD at time t can be found by substituting (7.5) and (7.6) into (5.5) as follows:

$$\begin{aligned} D_{t|k}(\mathbf{x}_t|Z_{1:k}) = & \sum_{i=1}^{J_{t|t}} \sum_{j=1}^{J_{t+1|k}} P_s w_{t|t}^i w_{t+1|k}^j \mathcal{N}\left(\mathbf{x}_t; \mathbf{m}_{t|t}^i, P_{t|t}^i\right) \\ & \times \int \frac{\mathcal{N}\left(\mathbf{x}_{t+1}; \mathbf{m}_{t+1|k}^j, P_{t+1|k}^j\right) \mathcal{N}\left(\mathbf{x}_{t+1}; F_t\mathbf{x}_t, F_t P_{t|t}^i F_t^T + Q_t\right)}{D_{t+1|t}(\mathbf{x}_{t+1}|Z_{1:t})} d\mathbf{x}_{t+1} \\ & + (1 - P_s) \sum_{i=1}^{J_{t|t}} w_{t|t}^i \mathcal{N}\left(\mathbf{x}_t; \mathbf{m}_{t|t}^i, P_{t|t}^i\right) \end{aligned} \tag{7.7}$$

In the above, the term in the second line, which has the normalization factor $D_{t+1|t}(\mathbf{x}_{t+1}|Z_{1:t})$ (a Gaussian mixture), does not allow an exact closed-form solution. However, one can consider the above expression as interaction of the components of the Gaussian mixtures corresponding to the filtered PHD at time t and the smoothed

PHD at time $t+1$. Each interaction can be viewed as Kalman smoother with corresponding means and covariances of the components considered. Therefore, the proposed smoothed PHD can be written as

$$\begin{aligned} D_{t|k}(\mathbf{x}_t|Z_{1:k}) &= \sum_{i=1}^{J_{t|t}} \sum_{j=1}^{J_{t+1|k}} w_{t|k}^{i,j} \mathcal{N}\left(\mathbf{x}_{t+1}; \mathbf{m}_{t|k}^{i,j}, P_{t|k}^{i,j}\right) \\ &\quad + (1-P_s) \sum_{i=1}^{J_{t|t}} w_{t|t}^{i} \mathcal{N}\left(\mathbf{x}_t; \mathbf{m}_{t|t}^{i}, P_{t|t}^{i}\right) \end{aligned} \tag{7.8}$$

where the mean and the covariance are given by the standard backward Kalman smoothing equations as follows:

$$\mathbf{m}_{t|k}^{i,j} = \mathbf{m}_{t|t}^{i} + A_t^i \left(\mathbf{m}_{t+1|k}^{j} - \mathbf{m}_{t+1|t}^{i}\right) \tag{7.9}$$

$$P_{t|k}^{i,j} = P_{t|t}^{i} + A_t^i \left(P_{t+1|k}^{j} - P_{t+1|t}^{i}\right) A_t^i \tag{7.10}$$

$$A_t^i = P_{t|t}^{i} F_t^T {P_{t+1|t}^{i}}^{-1} \tag{7.11}$$

The smoothed weight $w_{t|k}^{i,j}$ in (7.8) needs to be determined. Using the standard Kalman smoothing recursion [44] the Gaussian component in the first summation of (7.8) can be written as

$$\begin{aligned} \mathcal{N}\left(\mathbf{x}_{t+1}; \mathbf{m}_{t|k}^{i,j}, P_{t|k}^{i,j}\right) &= \mathcal{N}\left(\mathbf{x}_t; \mathbf{m}_{t|t}^{i}, P_{t|t}^{i}\right) \\ &\quad \times \int \frac{\mathcal{N}\left(\mathbf{x}_{t+1}; F\mathbf{x}_t, Q\right) \mathcal{N}\left(\mathbf{x}_{t+1}; \mathbf{m}_{t+1|k}^{j}, P_{t+1|k}^{j}\right)}{\mathcal{N}\left(\mathbf{x}_{t+1}; \mathbf{m}_{t+1|t}^{i}, P_{t+1|t}^{i}\right)} d\mathbf{x}_{t+1} \end{aligned} \tag{7.12}$$

Using (7.12), (7.8) can be rewritten as

$$\begin{aligned} D_{t|k}(\mathbf{x}_t|Z_{1:k}) &= \sum_{i=1}^{J_{t|t}} \sum_{j=1}^{J_{t+1|k}} w_{t|k}^{i,j} \mathcal{N}\left(\mathbf{x}_t; \mathbf{m}_{t|t}^i, P_{t|t}^i\right) \\ &\quad \times \int \frac{\mathcal{N}\left(\mathbf{x}_{t+1}; F\mathbf{x}_t, Q\right) \mathcal{N}\left(\mathbf{x}_{t+1}; \mathbf{m}_{t+1|k}^j, P_{t+1|k}^j\right)}{\mathcal{N}\left(\mathbf{x}_{t+1}; \mathbf{m}_{t+1|t}^i, P_{t+1|t}^i\right)} d\mathbf{x}_{t+1} \\ &\quad + (1 - P_s) \sum_{i=1}^{J_{t|t}} w_{t|t}^i \mathcal{N}\left(\mathbf{x}_t; \mathbf{m}_{t|t}^i, P_{t|t}^i\right) \end{aligned} \tag{7.13}$$

By comparing the actual smoothed PHD in (7.7) and the proposed smoothed PHD in (7.13), the error in the proposed smoothed PHD is given by

$$\begin{aligned} e_{D_{t|k}(\mathbf{x}_t|Z_{1:k})} &= \sum_{i=1}^{J_{t|t}} \sum_{j=1}^{J_{t+1|k}} \int \mathcal{N}\left(\mathbf{x}_t; \mathbf{m}_{t|t}^i, P_{t|t}^i\right) \mathcal{N}\left(\mathbf{x}_{t+1}; F\mathbf{x}_t, Q\right) \mathcal{N}\left(\mathbf{x}_{t+1}; \mathbf{m}_{t+1|k}^j, P_{t+1|k}^j\right) \\ &\quad \times \left[\frac{w_{t|k}^{i,j}}{\mathcal{N}\left(\mathbf{x}_{t+1}; \mathbf{m}_{t+1|t}^i, P_{t+1|t}^i\right)} - \frac{P_s w_{t|t}^i w_{t+1|k}^j}{\sum_{n=1}^{J_{t+1|t}} w_{t+1|t}^n \mathcal{N}\left(\mathbf{x}_{t+1}; \mathbf{m}_{t+1|t}^n, P_{t+1|t}^n\right)} \right] \\ &\quad \times d\mathbf{x}_{t+1} \end{aligned} \tag{7.14}$$

The above can be rewritten as follows:

$$\begin{aligned} e_{D_{t|k}(\mathbf{x}_t|Z_{1:k})} &= \sum_{i=1}^{J_{t|t}} \sum_{j=1}^{J_{t+1|k}} \int \mathcal{N}\left(\mathbf{x}_t; \mathbf{m}_{t|t}^i + A_t^i(\mathbf{x}_{t+1} - \mathbf{m}_{t+1|t}^i), P_{t|t}^i - A_t^i P_{t+1|t}^i A_t^{iT}\right) \\ &\quad \times \mathcal{N}\left(\mathbf{x}_{t+1}; \mathbf{m}_{t+1|t}^i, P_{t+1|t}^i\right) \mathcal{N}\left(\mathbf{x}_{t+1}; \mathbf{m}_{t+1|k}^j, P_{t+1|k}^j\right) \\ &\quad \times \left[\frac{w_{t|k}^{i,j}}{\mathcal{N}\left(\mathbf{x}_{t+1}; \mathbf{m}_{t+1|t}^i, P_{t+1|t}^i\right)} - \frac{P_s w_{t|t}^i w_{t+1|k}^j}{\sum_{n=1}^{J_{t+1|t}} w_{t+1|t}^n \mathcal{N}\left(\mathbf{x}_{t+1}; \mathbf{m}_{t+1|t}^n, P_{t+1|t}^n\right)} \right] \\ &\quad \times d\mathbf{x}_{t+1} \end{aligned} \tag{7.15}$$

where identity [44]

$$\mathcal{N}\left(\mathbf{x}_t; \mathbf{m}^i_{t|t}, P^i_{t|t}\right)\mathcal{N}\left(\mathbf{x}_{t+1}; F\mathbf{x}_t, Q\right) = \mathcal{N}\left(\mathbf{x}_t; \mathbf{m}^i_{t|t} + A^i_t(\mathbf{x}_{t+1} - \mathbf{m}^i_{t+1|t}), P^i_{t|t} - A^i_t P^i_{t+1|t} A^{i^T}_t\right)\mathcal{N}\left(\mathbf{x}_{t+1}; \mathbf{m}^i_{t+1|t}, P^i_{t+1|t}\right)$$

is used. Here, the predicted mean $\mathbf{m}^i_{t+1|t}$ and its covariance $P^i_{t+1|t}$ are defined in (2.18) and (2.19), respectively. The matrix A^i_t is defined by (4.7). Now consider the second line in (7.15): the first term is the predicted Gaussian function and the second term smoothed Gaussian function. Since the covariance of the smoothed Gaussian function is usually much smaller than that of predicted Gaussian function, the second term can be approximated with a δ-function. That is,

$$\mathcal{N}\left(\mathbf{x}_{t+1}; \mathbf{m}^j_{t+1|k}, P^j_{t+1|k}\right) \approx \delta\left(\mathbf{x}_{t+1} - \mathbf{m}^j_{t+1|k}\right) \tag{7.16}$$

Using this approximation, the error function in (7.15) can be simplified as follows:

$$\begin{aligned} e_{D_{t|k}(\mathbf{x}_t|Z_{1:k})} = \sum_{i=1}^{J_{t|t}}\sum_{j=1}^{J_{t+1|k}} & \mathcal{N}\left(\mathbf{x}_t; \mathbf{m}^i_{t|t} + A^i_t(\mathbf{m}^j_{t+1|k} - \mathbf{m}^i_{t+1|t}), P^i_{t|t} - A^i_t P^i_{t+1|t} A^{i^T}_t\right) \\ & \times\mathcal{N}\left(\mathbf{m}^j_{t+1|k}; \mathbf{m}^i_{t+1|t}, P^i_{t+1|t}\right)\left[\frac{w^{i,j}_{t|k}}{\mathcal{N}\left(\mathbf{m}^j_{t+1|k}; \mathbf{m}^i_{t+1|t}, P^i_{t+1|t}\right)}\right. \\ & \left. - \frac{P_s w^i_{t|t} w^j_{t+1|k}}{\sum_{n=1}^{J_{t+1|t}} w^n_{t+1|t}\mathcal{N}\left(\mathbf{m}^j_{t+1|k}; \mathbf{m}^n_{t+1|t}, P^n_{t+1|t}\right)}\right] \end{aligned} \tag{7.17}$$

The above error function can be minimized by setting the whole term in the square bracket to zero that leads to required smoothed weight as follows:

$$w_{t|k}^{i,j} = \frac{P_s w_{t+1|k}^{j} w_{t|t}^{i} \mathcal{N}\left(\mathbf{m}_{t+1|k}^{j}; \mathbf{m}_{t+1|t}^{i}, P_{t+1|t}^{i}\right)}{\sum_{l=1}^{J_{t+1|t}} w_{t+1|t}^{l} \mathcal{N}\left(\mathbf{m}_{t+1|k}^{j}; \mathbf{m}_{t+1|t}^{l}, P_{t+1|t}^{l}\right)} \tag{7.18}$$

Similar to the forward filtering, the backward smoothing suffers from increasing number of components, which can be mitigated by using Gaussian mixture reduction methods.

7.3 GMPHD Smoothing for Maneuvering Target Tracking

For maneuvering target tracking problems, a multiple model approach for PHD-based multitarget states estimator was proposed in [77]. In [99], a Gaussian mixture implementation for the multiple model approach was provided. This section provides a Gaussian mixture implementation of the smoothing algorithm for the Multiple Model PHD (MMPHD) approach, which is a natural extension of the PHD smoothing algorithm for maneuvering targets. In Section 7.3.1, Gaussian Mixture implementation of MMPHD is reviewed. A new Gaussian mixture implementation of MMPHD smoother is proposed in Section 7.3.2.

7.3.1 Gaussian Mixture Implementation of MMPHD Filter

In Gaussian mixture implementation of the MMPHD filter, the filtered mode-dependent density at time $k-1$, $D_{k-1|k-1}(\mathbf{x}_{k-1}, r_{k-1}|Z_{1:k-1})$ is represented by the sum of Gaussian components as follows:

$$D_{k-1|k-1}(\mathbf{x}_{k-1}, r_{k-1} = v|Z_{1:k-1}) = \sum_{i=1}^{J_{k-1}(v)} w^i_{k-1|k-1}(v)\mathcal{N}\left(\mathbf{x}; \mathbf{m}^i_{k-1|k-1}(v), P^i_{k-1|k-1}(v)\right)$$

The propagation of these Gaussian components is similar to the PHD filter recursion in Section 3.5 by replacing PHD by the above mode dependent PHD.

7.3.2 Gaussian Mixture MMPHD Smoothing

In Gaussian mixture MMPHD smoothing, the smoothed Gaussian components at time $k-L$ are found using backward recursion (6.6) as follows:

$$\begin{aligned} D_{t|k}(\mathbf{x}_t, r_t = v|Z_{1:k}) \;=\; & \sum_{u=1}^{N_r} \sum_{i=1}^{J_{t|t}(v)} \sum_{j=1}^{J_{t+1|k}(u)} w^{i,j}_{t|k}(v,u)\mathcal{N}\left(\mathbf{x}_{t+1}; \mathbf{m}^{i,j}_{t|k}(v,u), P^{i,j}_{t|k}(v,u)\right) \\ & + (1-P_s) \sum_{i=1}^{J_{t|t}(v)} w^i_{t|t}(v)\mathcal{N}\left(\mathbf{x}_t; \mathbf{m}^i_{t|t}(v), P^i_{t|t}(v)\right) \end{aligned} \tag{7.19}$$

where $\mathbf{m}^{i,j}_{t|k}(v,u)$ and $P^{i,j}_{t|k}(v,u)$ are given by (7.9) and (7.10) with appropriate models. The smoothed component weight $w^{i,j}_{t|k}(v,u)$ is given by

$$w^{i,j}_{t|k}(v,u) = \frac{P_s w^j_{t+1|k}(u) w^i_{t|t}(v) h_{vu}\mathcal{N}\left(\mathbf{m}^j_{t+1|k}(u); \mathbf{m}^i_{t+1|t}(v), P^i_{t+1|t}(v)\right)}{\sum_{l=1}^{J_{t+1|t}(v)} w^l_{t+1|t}(v)\mathcal{N}\left(\mathbf{m}^j_{t+1|k}(u); \mathbf{m}^l_{t+1|t}(v), P^l_{t+1|t}(v)\right)} \tag{7.20}$$

At the end of the iteration, the smoothed Gaussian components at time $k-L$ are used to find smoothing outputs: namely, smoothed expected number of targets, corresponding target state estimates and modes.

7.4 Simulation

In this section, the results of the simulation studies for the new multiple sensor PHD update method (Section 7.1) and GMPHD smoothing method (Section 7.2) are presented. Section 7.4.1 provides results for various multiple sensor update methods, while Section 7.4.2 demonstrates the improved performance of the GMPHD smoothing method. In this work, we consider a multiple bearing only sensor system with a 2D target scenario. As shown in Figure 7.1, it consists of three sensors, which are located at [0 -2] km, [-2 0] km, and [2 0] km, respectively. The number of targets in the region is time-varying due to target appearance and disappearance at any time. Spontaneous target birth is assumed to have Gaussian mixture intensity as follows:

$$\gamma_k(\mathbf{x}_k) = 0.05\mathcal{N}(\mathbf{x}_k|\mathbf{x}_b^1, Q_b) + 0.05\mathcal{N}(\mathbf{x}_k|\mathbf{x}_b^2, Q_b) \tag{7.21}$$

where $\mathbf{x}_b^1 = [-1200\ 20\ -1700\ 15]^T$, $\mathbf{x}_b^2 = [-1200\ -20\ -1700\ 15]^T$, and $Q_b = \text{diag}([2500\ 250\ 2500\ 250])$. The target state at time k, $\mathbf{x}_k = [x_k\ \dot{x}_k\ y_k\ \dot{y}_k]^T$, consists of the position, $[x_k\ y_k]^T$, and the velocity, $[\dot{x}_k\ \dot{y}_k]^T$ of the target. Each existing target survives with probability of $P_{s,k} = 0.95$, which is state-independent. It has the following linear Gaussian target dynamics [7]:

$$\mathbf{x}_k = F\mathbf{x}_{k-1} + \mathbf{v}_{k-1} \tag{7.22}$$

where target transition matrix F is given by

$$F = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

and the process noise $\mathbf{v}_{k-1}$ has a normal distribution with mean zero and covariance Q, which is given by

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

Here, T is the sampling period equal to 1s. The process noise standard deviation, σ, is equal to 1 ms^{-1}. Position ground truth of two tracks over 100 scans are displayed in Figure 7.1. The individual plots for x and y components of each track against time show the start and finish times of the tracks in Figure 7.2. Each sensor provides bearing measurement which is related to target state as follows:

$$\theta_k^i = \arctan\left(\frac{y_k - y_s^i}{x_k - x_s^i}\right) + w_k^i \tag{7.23}$$

where $[x_s^i \ y_s^i]^T$ is the position of the ith sensor and w_k^i is a zero mean Gaussian noise with standard deviation of 0.5 rad. Other sensor parameters (the probability of detection P_d, the average rate of clutter returns per scan λ, and the field of view) are provided in Table 7.1. Simulation results given in this paper are based on 100 Monte

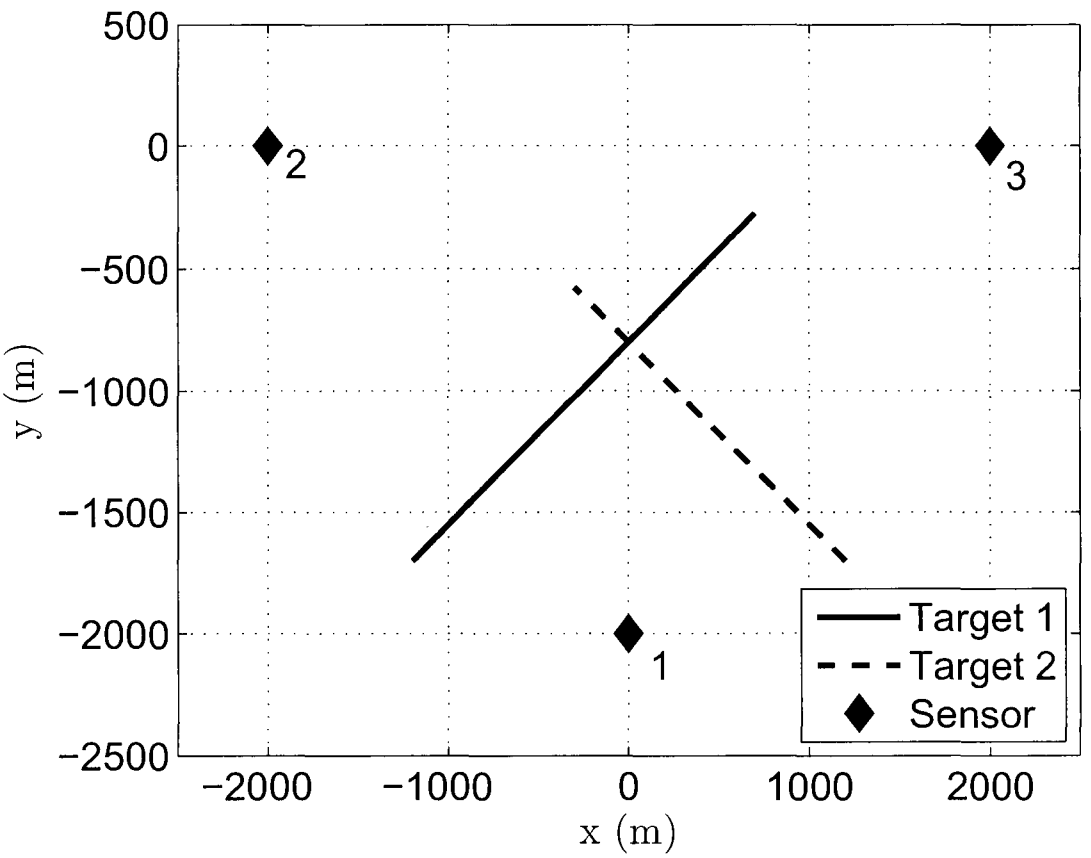
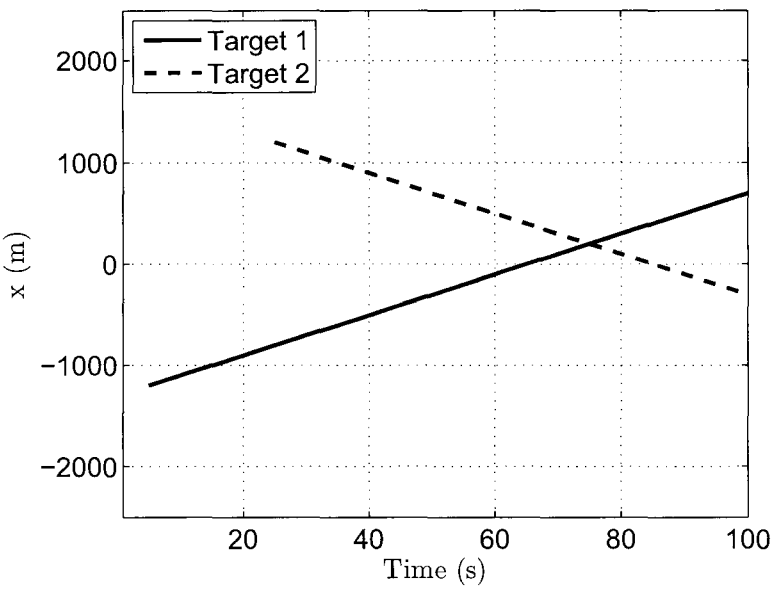
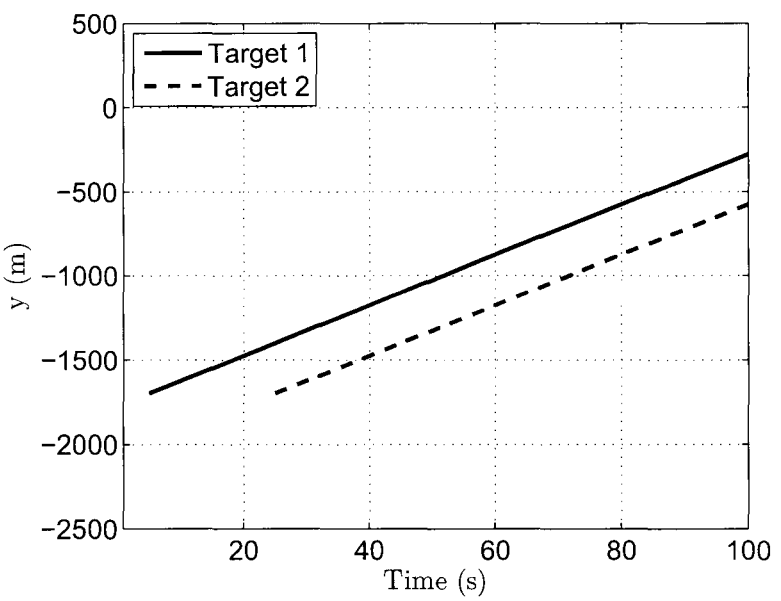


Figure 7.1: Ground truth: position plots of two true tracks



(a)



(b)

Figure 7.2: Ground truth: plots of x and y components of the 2 true tracks against time

Sensor	P_d	λ	FOV
1	0.95	1	$[0 \quad \pi]$
2	0.6	20	$[-\pi/2 \quad \pi/2]$
3	0.6	20	$[\pi/2 \quad 3\pi/2]$

Table 7.1: Sensor parameters

Carlo runs.

7.4.1 Filtering

In this section, the results of different update methods from Section 7.1 are compared. For the sequential update, different sequences update by placing the perfect sensor (sensor 1) at different places are considered as follows:

- Sequence 1: {1, 2, 3}
- Sequence 2: {3, 1, 2}
- Sequence 3: {2, 3, 1}

Figures 7.3 – 7.5 show the estimate of the number of targets and its one-sigma gates for each of the above update sequences. Wasserstein distance [41], which is a multi-target miss-distance generalization of the standard root mean square error of a single target problem to a multitarget problem, is given in Figure 7.6. Sequence 3 yields better results than the other two sequences. Accuracy of the last sensor influences the accuracy of estimation results, which is in contradiction with intuition and observations from other standard tracking algorithms [6], where it is recommended to update first with the most accurate sensor. Also the above results demonstrate that the update order in the multiple sensor PHD filtering determines the final estimates. That is, different sequences of update result in different PHD surfaces. Therefore,

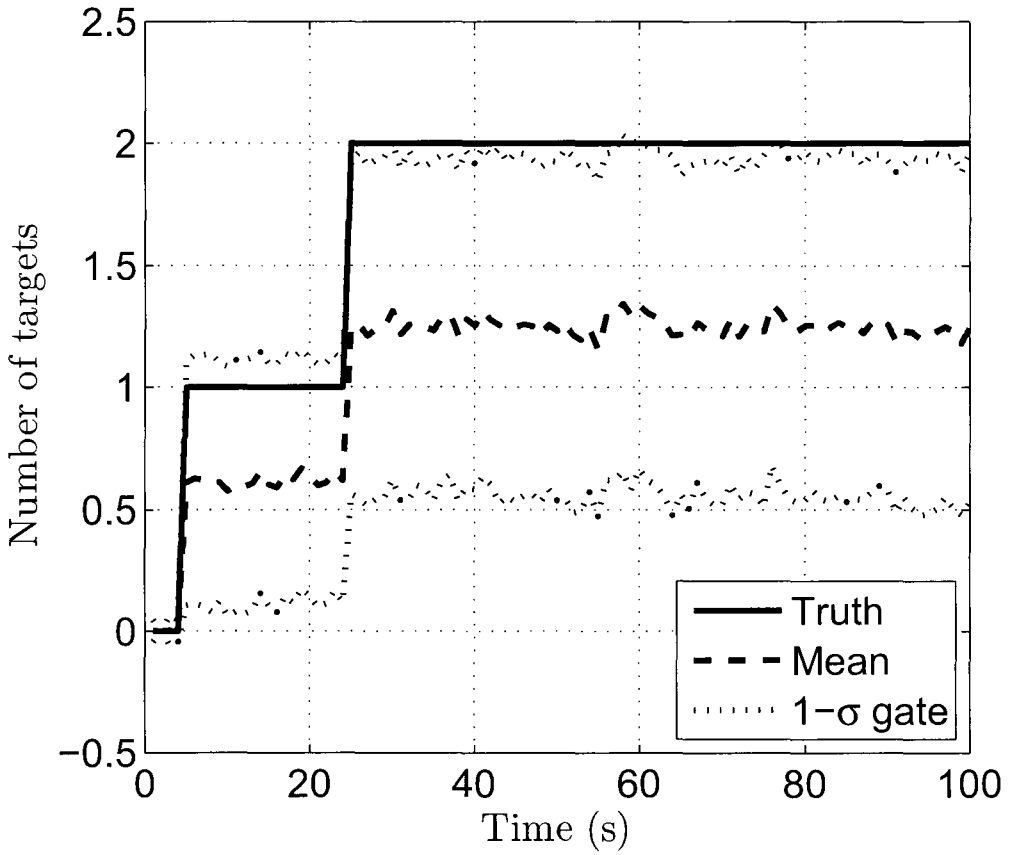


Figure 7.3: Number of targets for Sequence 1.

in the all-sequence method, we obtain the average of PHD surfaces of all possible sequences. This all-sequence method provides better results (Figures 7.7 and 7.9), but it requires more computational load for problems with a large number of sensors. On the other hand, the average of updated PHD surfaces from all sensors are obtained for the parallel method. This requires computational load similar to that of the single sequence method while having performance similar to the all-sequence method (Figures 7.8 and 7.9).

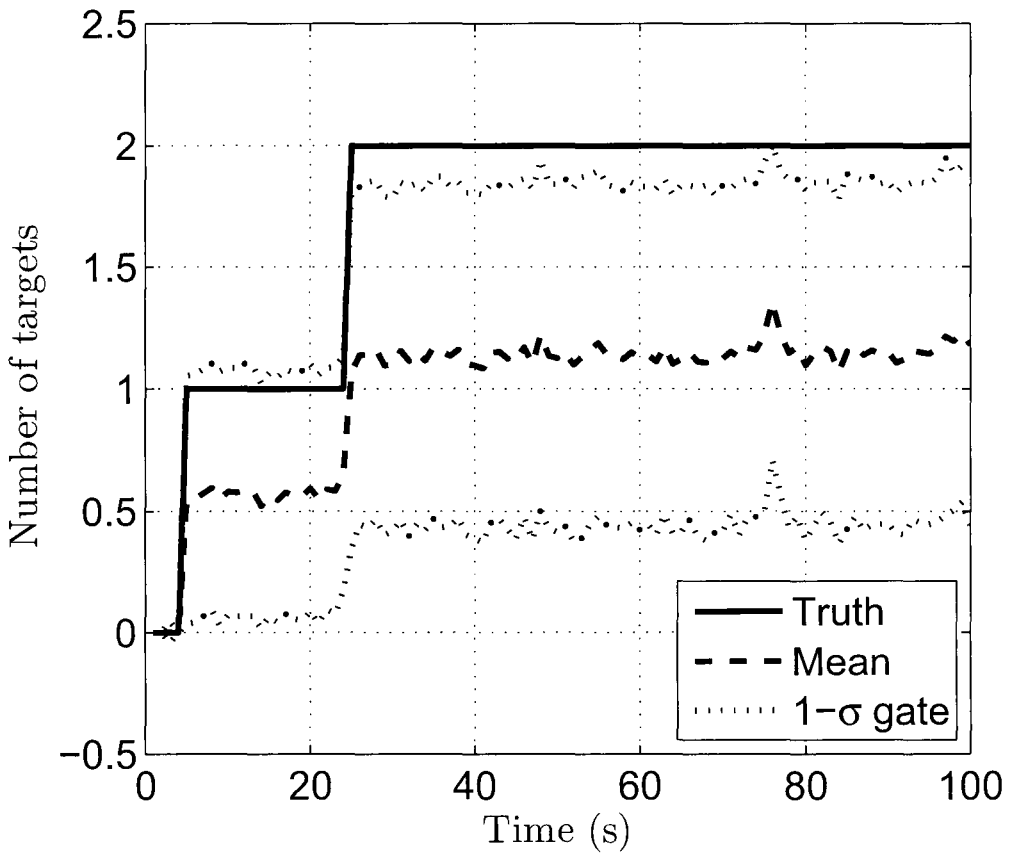


Figure 7.4: Number of targets for Sequence 2

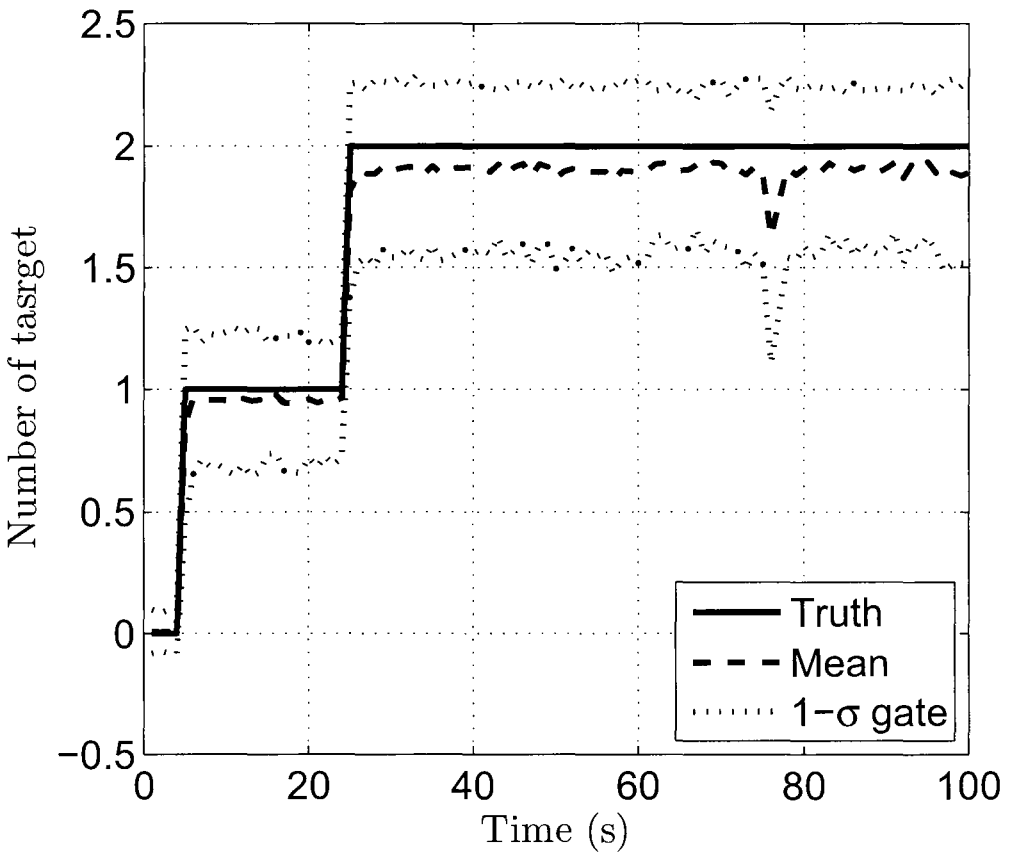


Figure 7.5: Number of targets for Sequence 3

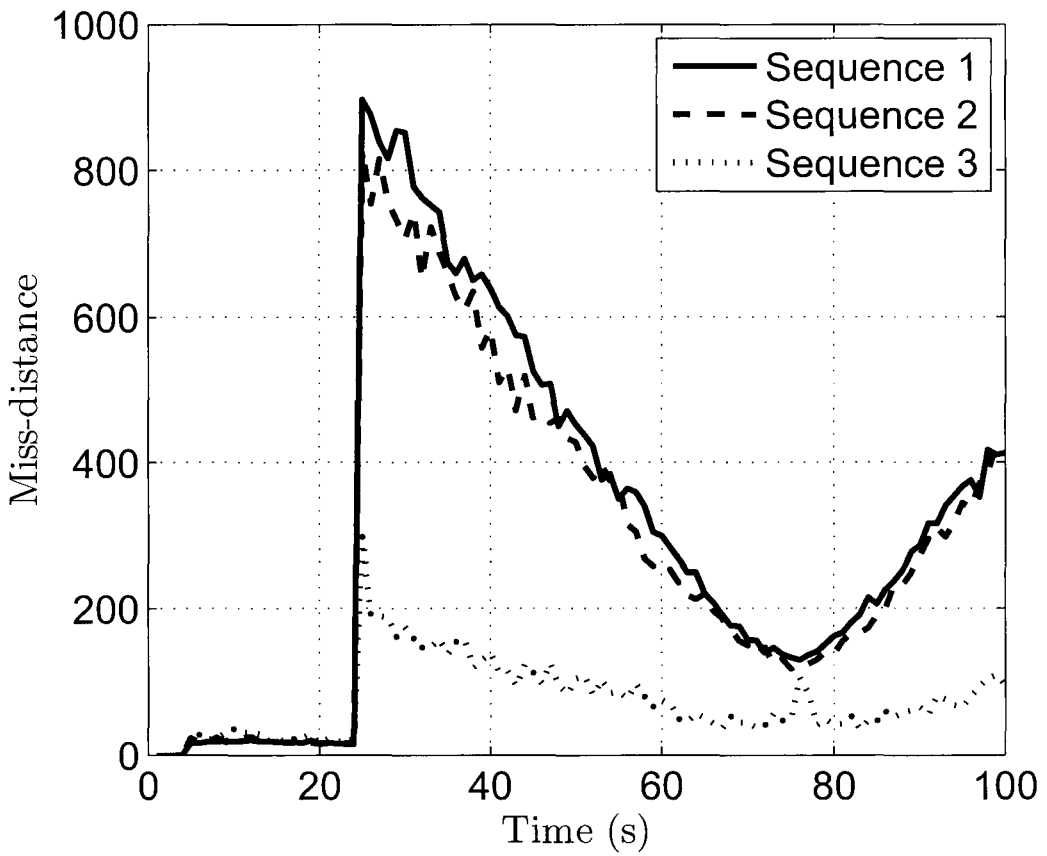


Figure 7.6: Multitarget miss-distance

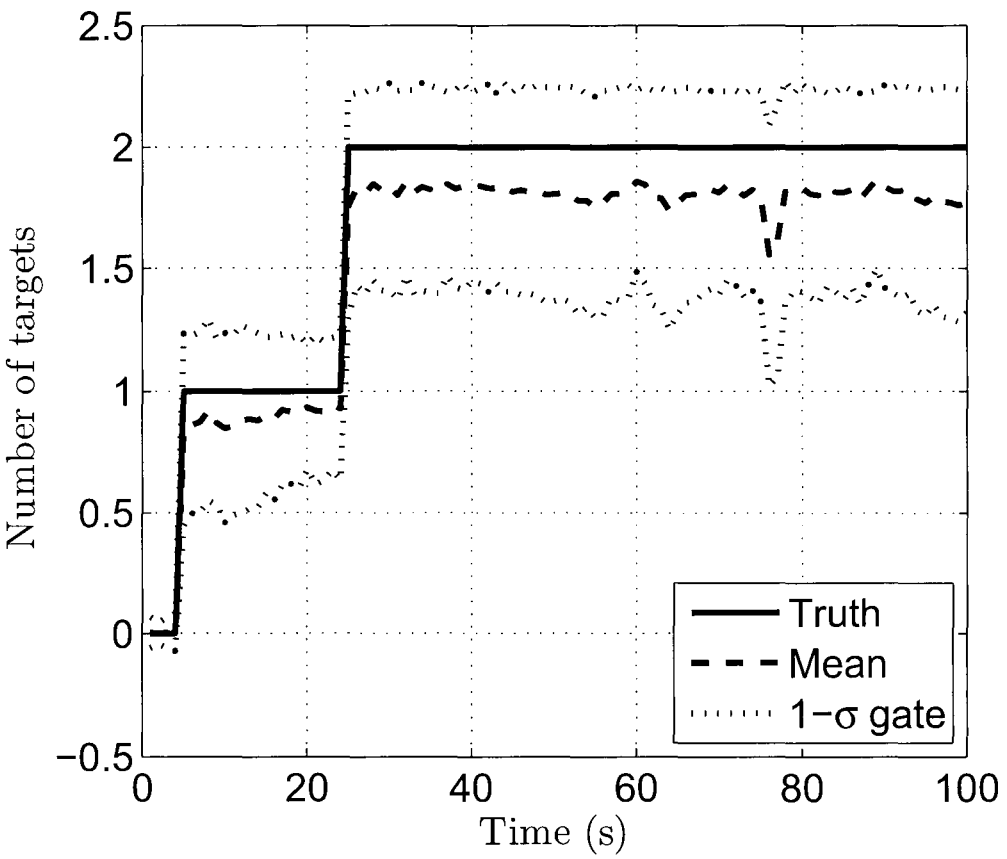


Figure 7.7: Number of targets for all-sequence update

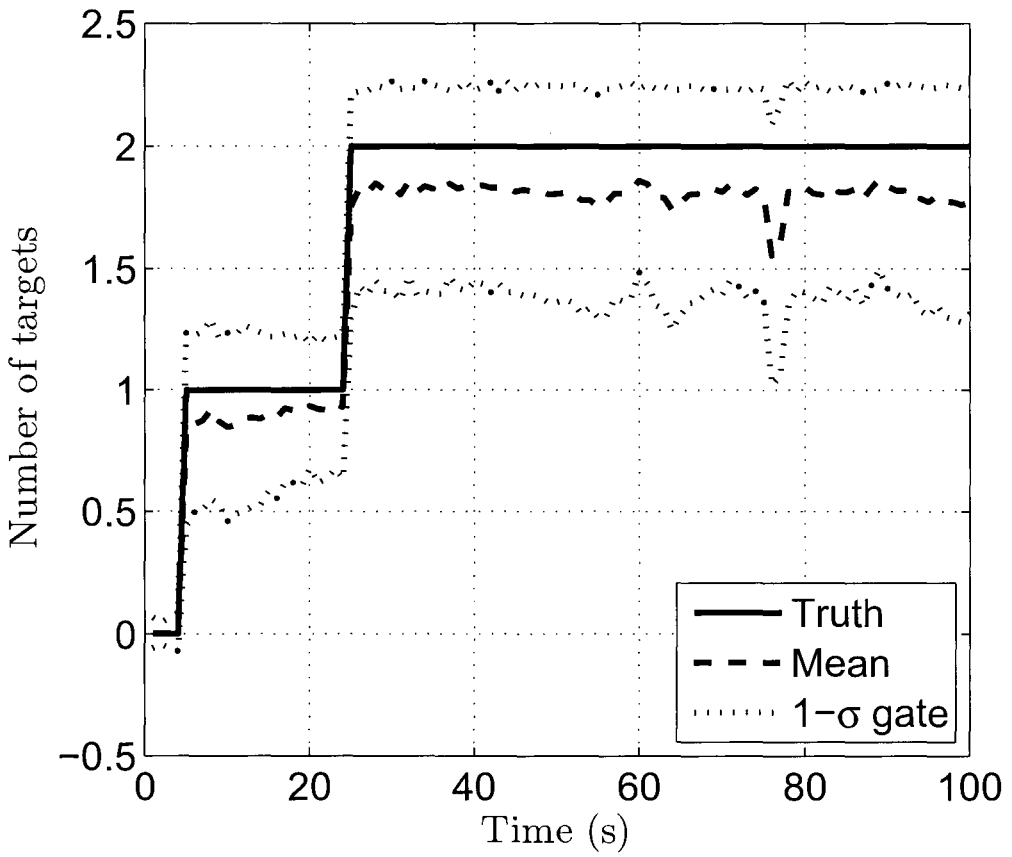


Figure 7.8: Number of targets for parallel update

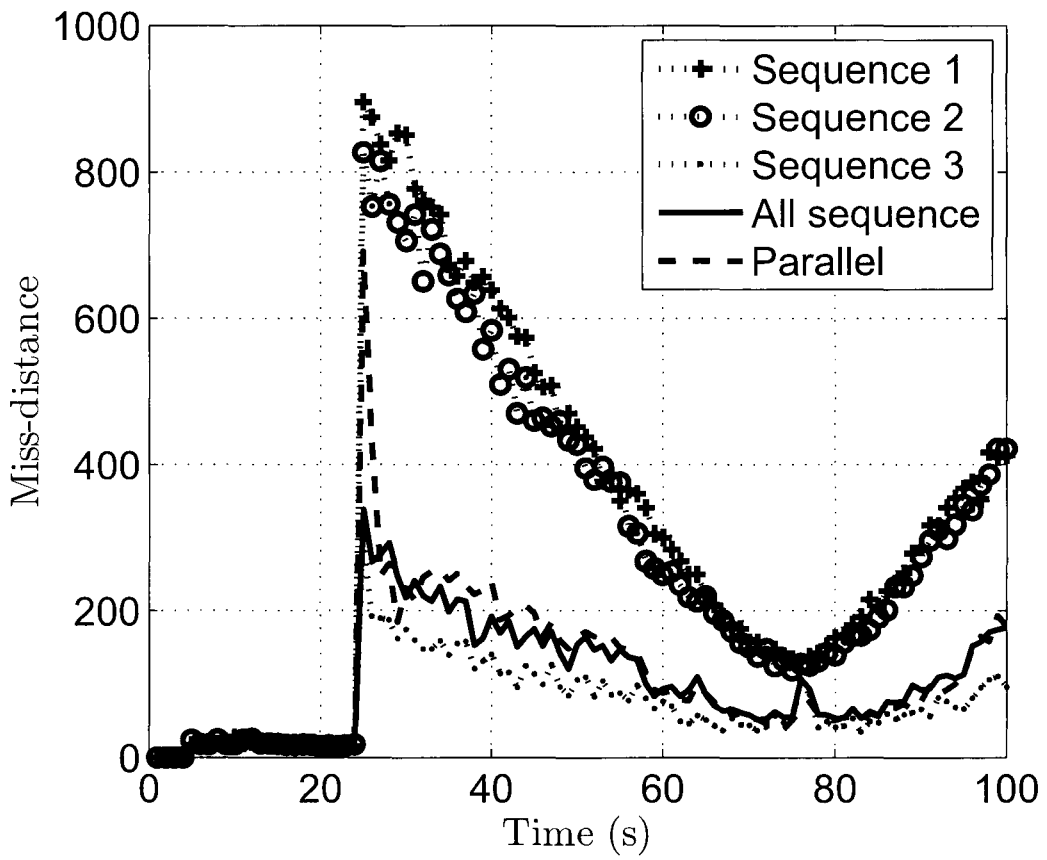


Figure 7.9: Multitarget miss-distance

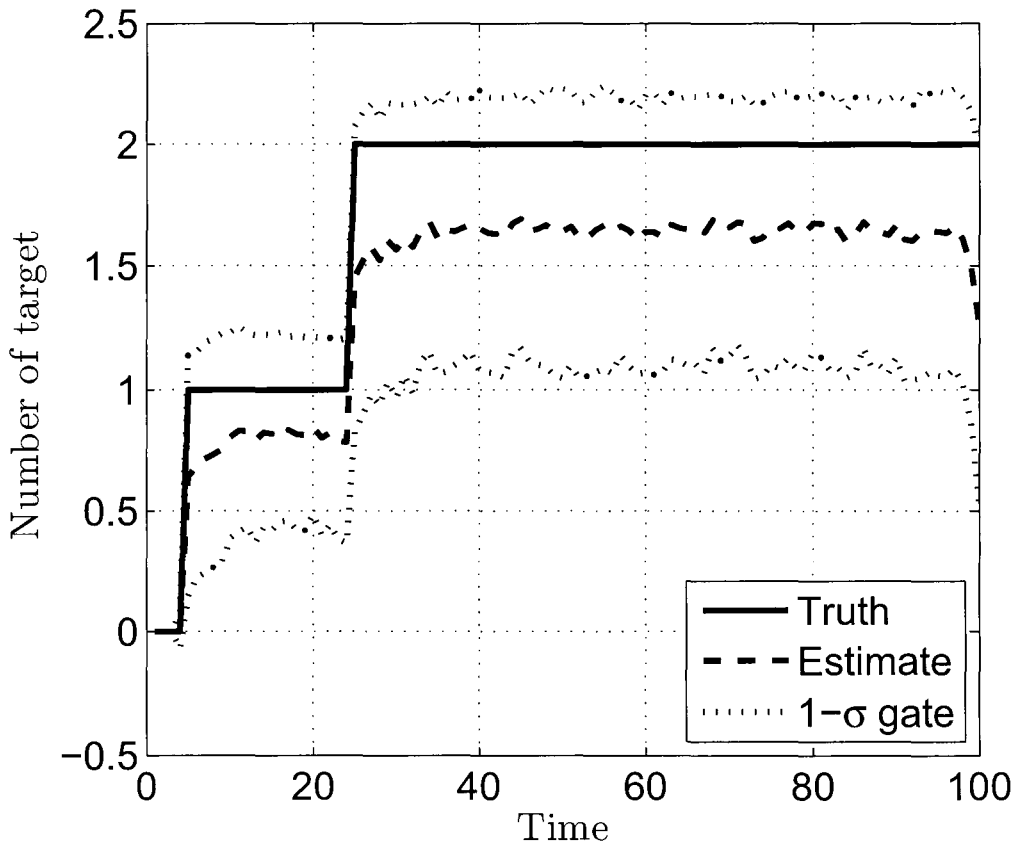


Figure 7.10: Number of targets for Sequence 1

7.4.2 Smoothing

In this section, results for the novel GMPHD smoothing algorithm are provided. Here, the smoothed states are obtained by backward PHD smoothing with a lag of 4 time steps. The estimates of the number of targets for various update methods are provided in Figures 7.10 – 7.14. Multitarget miss-distances of the estimates of both filter and smoother are shown in Figures 7.15 – 7.19. These results clearly demonstrate improved performance by the smoother over the filter.

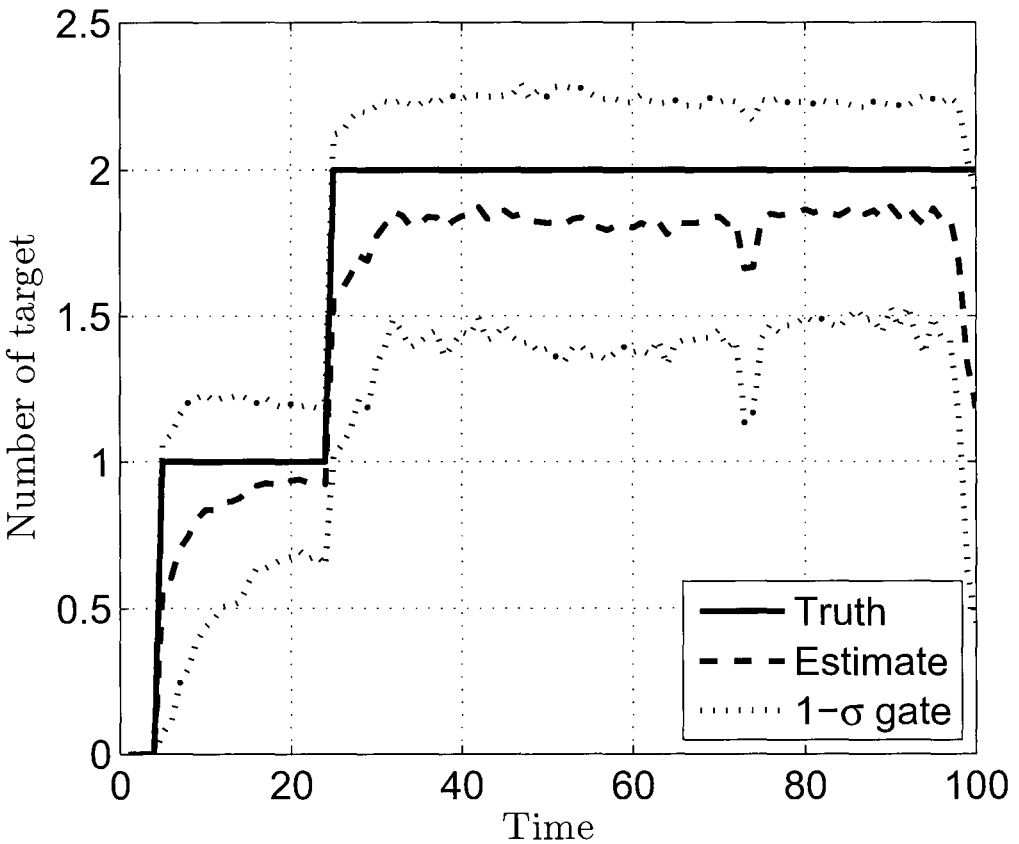


Figure 7.11: Number of targets for Sequence 2

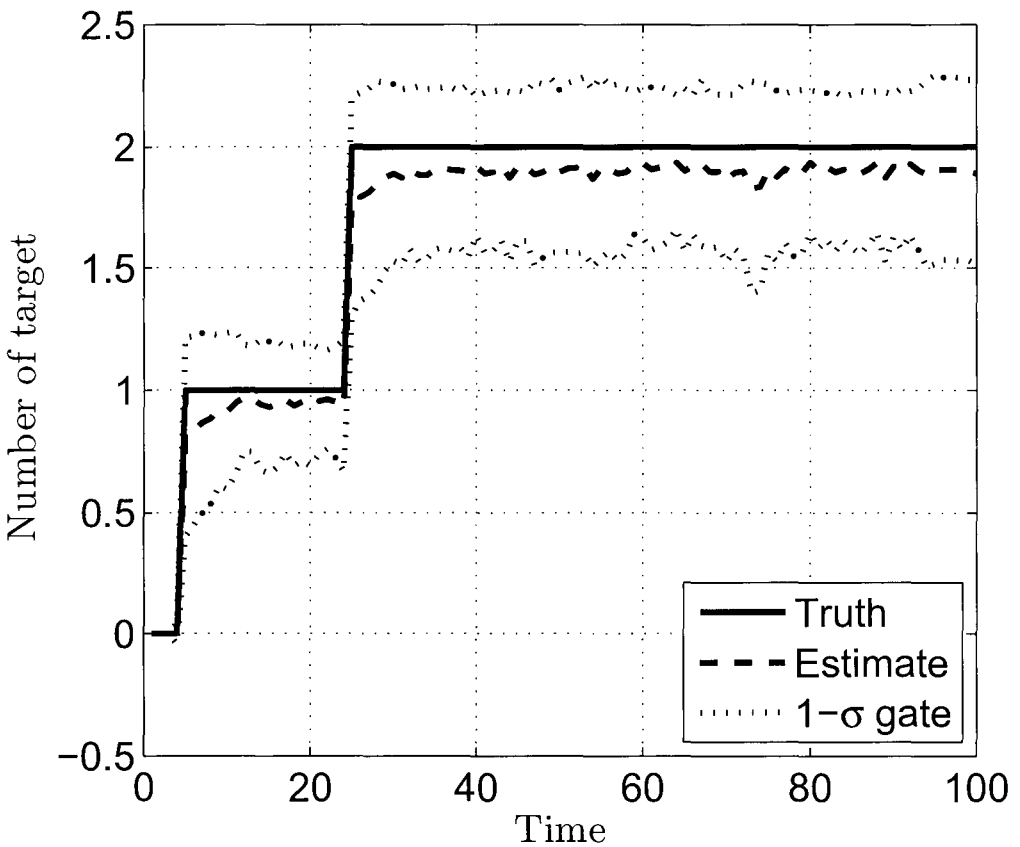


Figure 7.12: Number of targets for Sequence 3

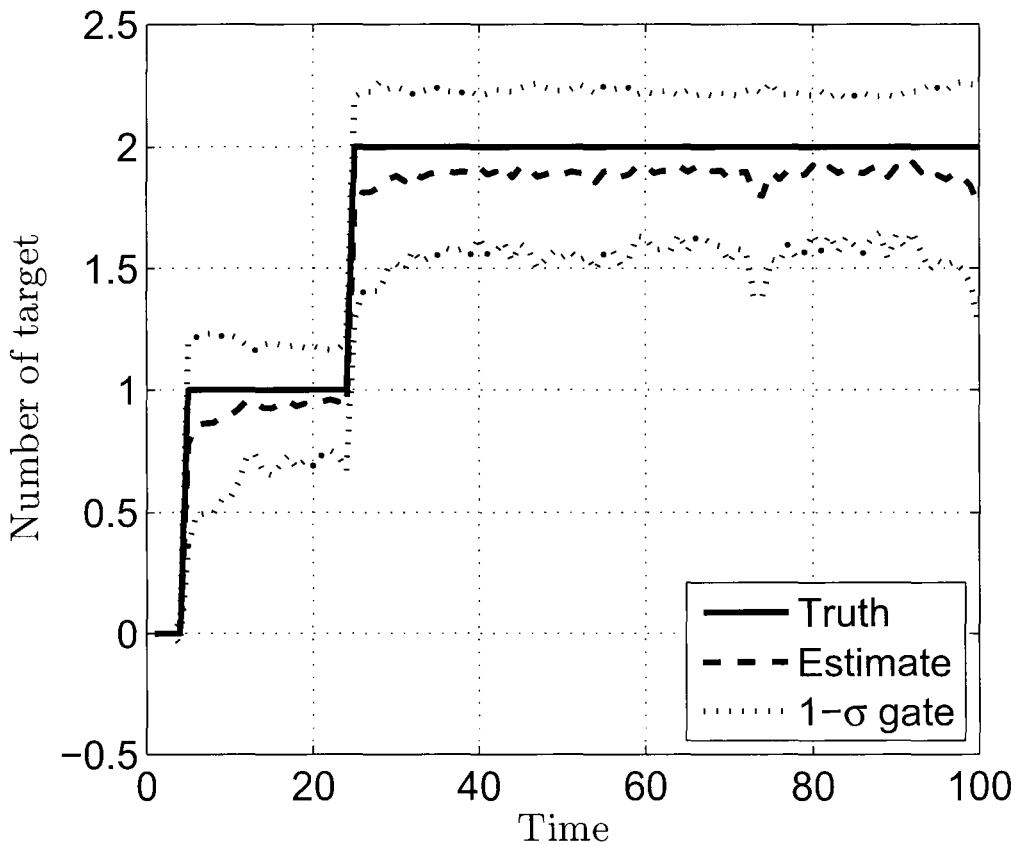


Figure 7.13: Number of targets for all-sequence update

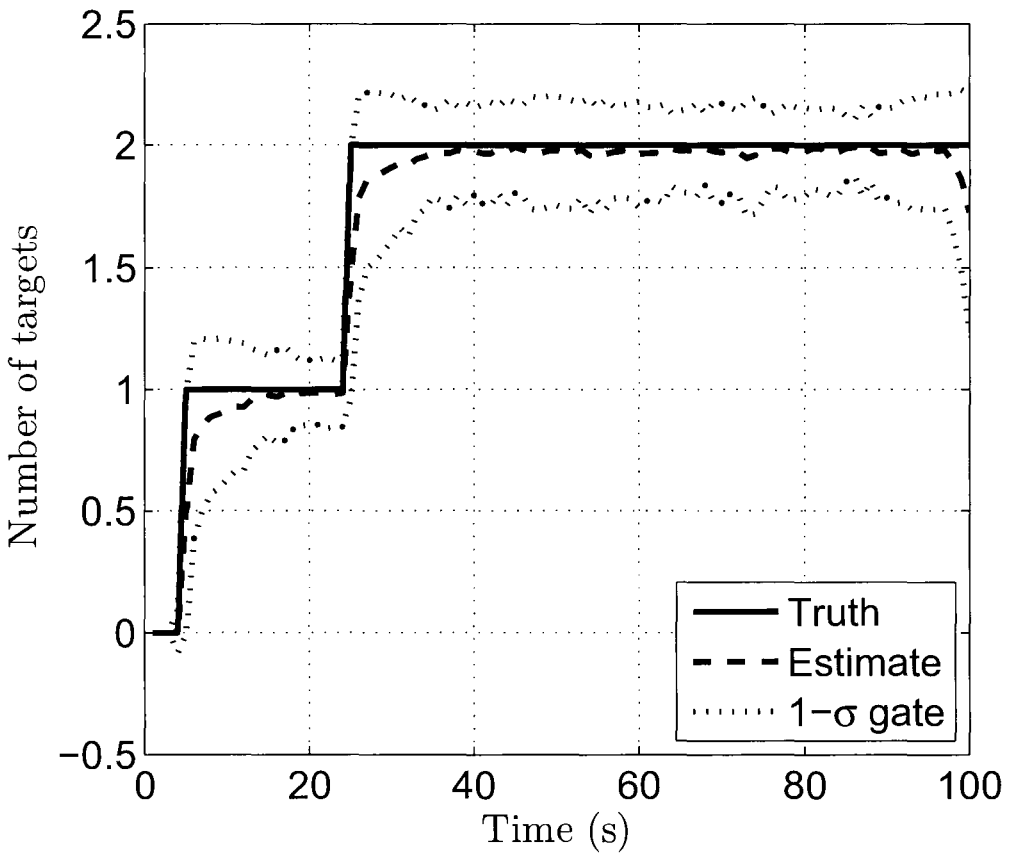


Figure 7.14: Number of targets for parallel update

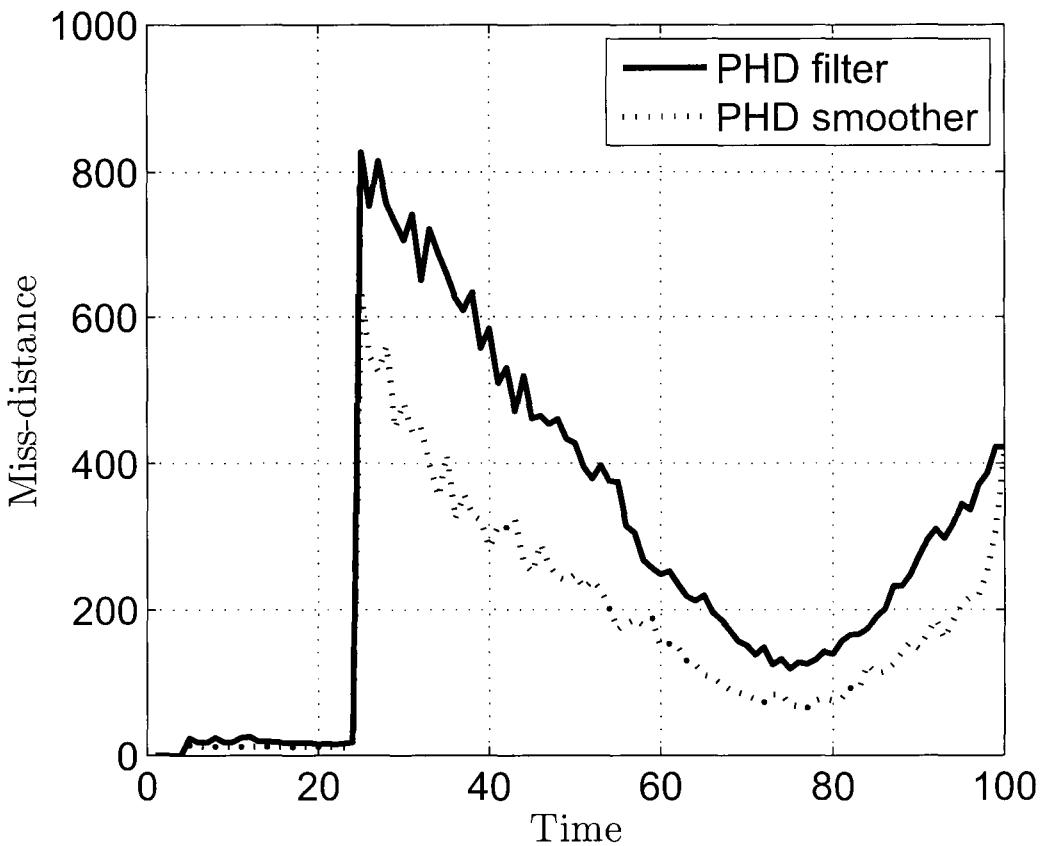


Figure 7.15: Multitarget miss-distance for Sequence 1

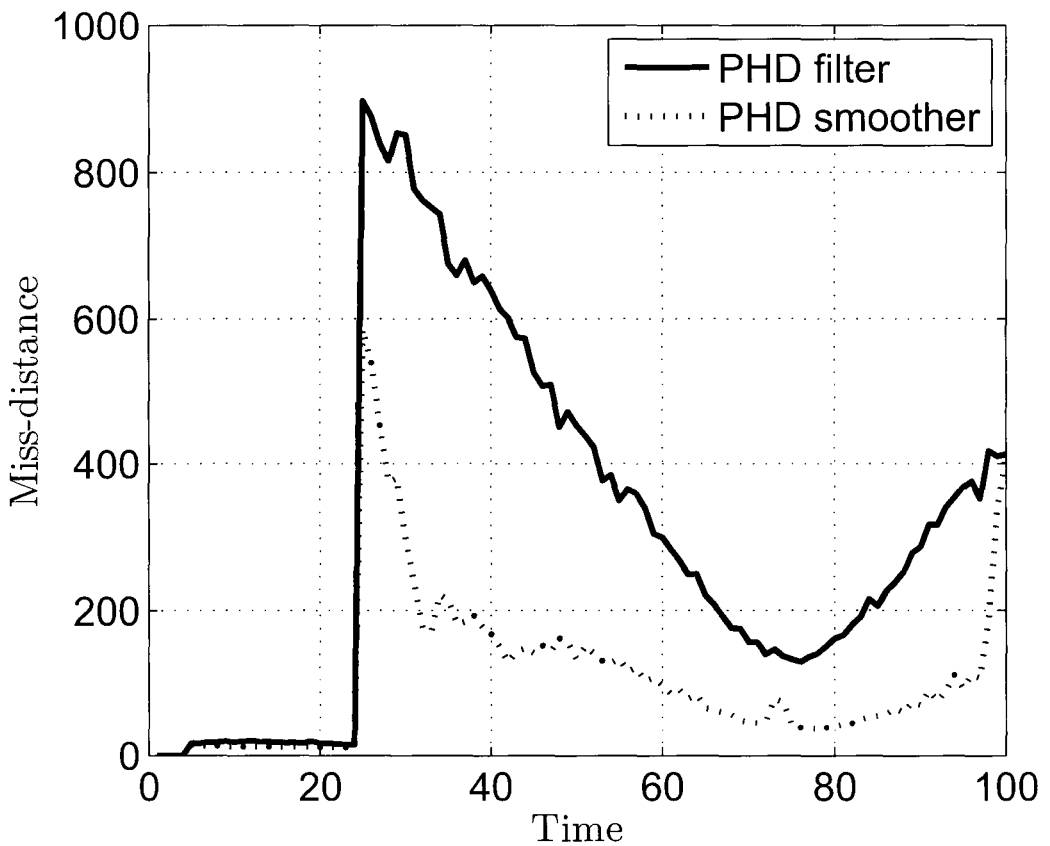


Figure 7.16: Multitarget miss-distance for Sequence 2

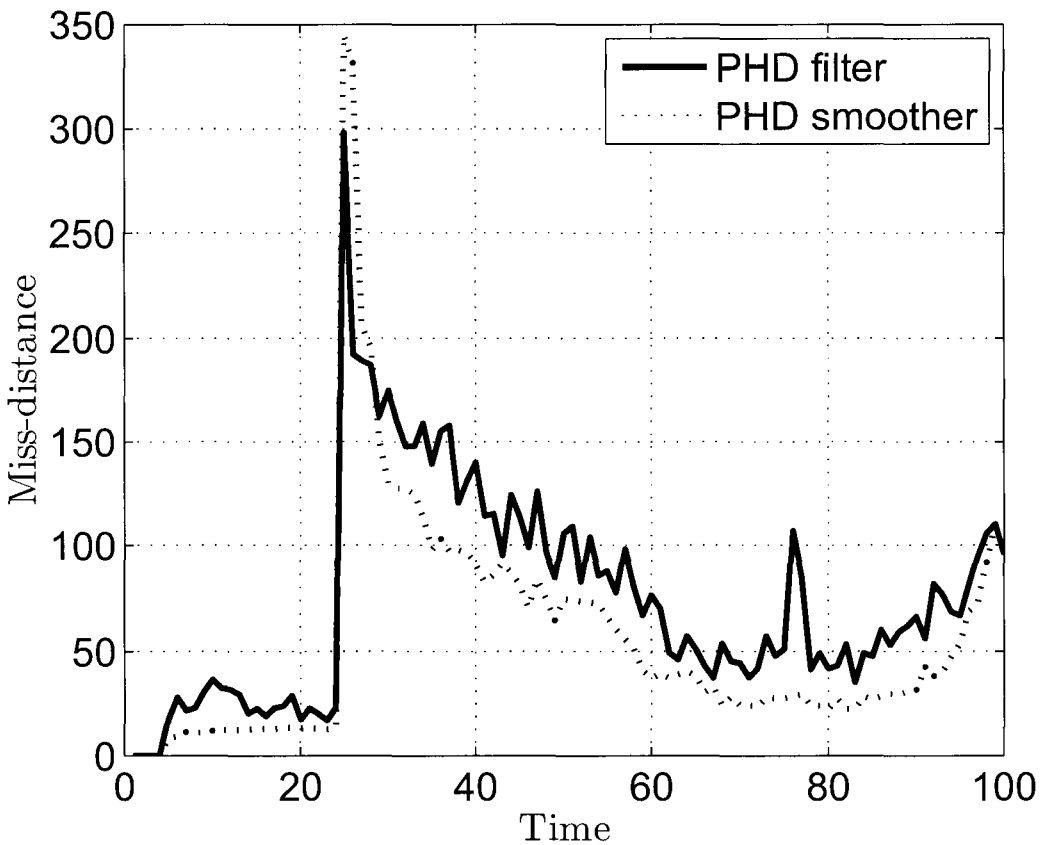


Figure 7.17: Multitarget miss-distance for Sequence 3

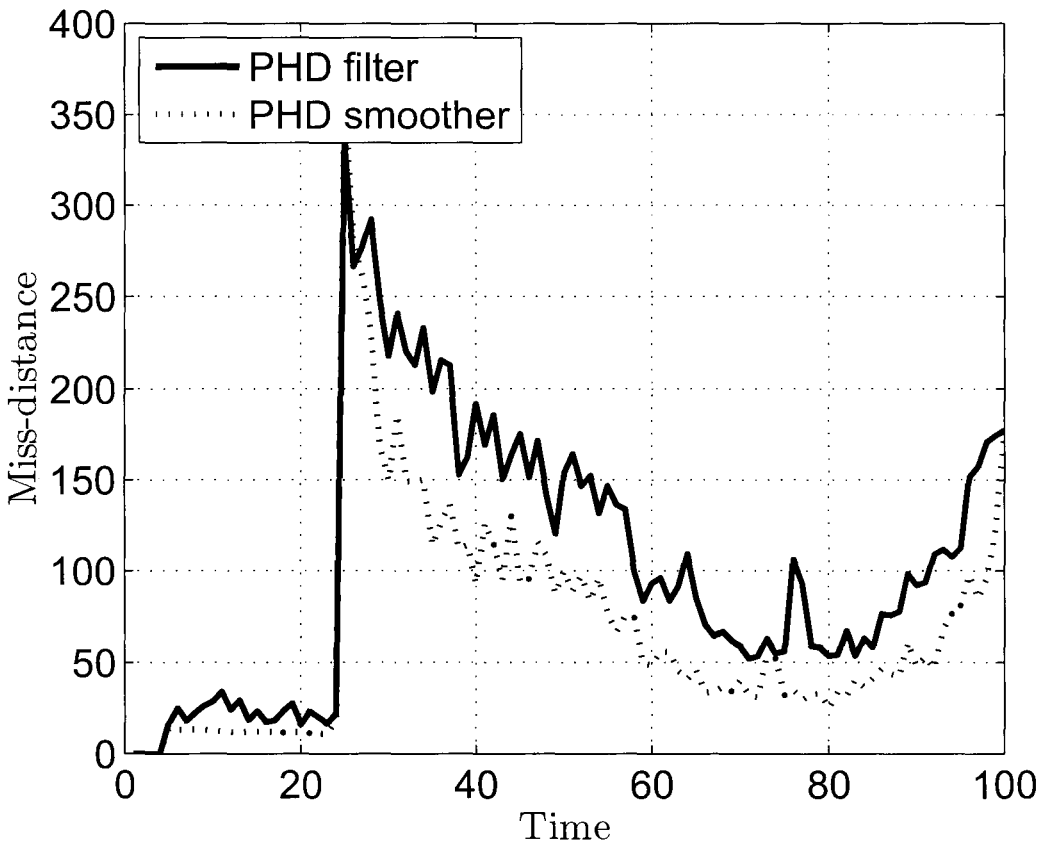


Figure 7.18: Multitarget miss-distance for all-sequence update

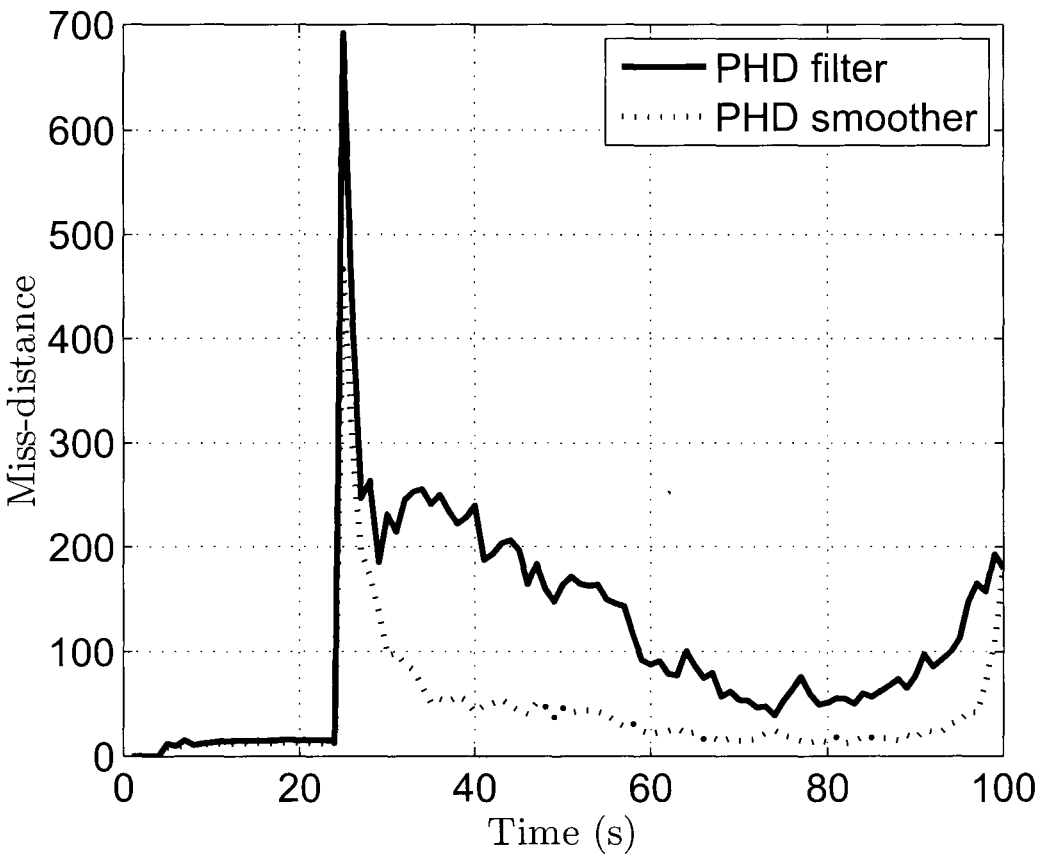


Figure 7.19: Multitarget miss-distance for parallel update.

Chapter 8

Interacting Multiple Model Smoothing

In this chapter, a novel smoothing method for Kalman based IMM estimator is proposed for tracking agile target. This RTS-type method involves forward filtering followed by backward smoothing while maintaining the fundamental spirit of the IMM. The forward filtering is performed using the standard IMM recursion (Section 2.1.3), while the backward smoothing is performed using a novel interacting smoothing recursion. This backward recursion, which is derived in Section 8.1, mimics the IMM estimator in the backward direction, where each mode conditioned smoother uses the standard Kalman smoothing recursion. The proposed algorithm is compared with existing augmented IMM smoothing method [16]. However, this augmented method explicitly assumes that there is no mode jumps within the lag [45][57]. Further, the proposed method does not require the existence of the inverse of the target dynamic as in [12][39]. Both the proposed method and augmented methods are summarized in Section 8.2. The simulation results are provided in Section 8.3.

8.1 Derivation of IMM Smoother

In backward smoothing, we are interested in smoothed density, $p\left(\mathbf{x}_t|\mathbf{Z}_1^k\right)$. Using total probability theorem, it can be written as follows:

$$p\left(\mathbf{x}_t|\mathbf{Z}_1^k\right) = \sum_{j=1}^{N_r} p\left(\mathbf{x}_t|r_t^j, \mathbf{Z}_1^k\right) P\left\{r_t^j|\mathbf{Z}_1^k\right\} \tag{8.1}$$

The rest of this section shows that the above density can be approximated as a mixture of mode conditioned smoothing densities (i.e., $p\left(\mathbf{x}_t|r_t^j, \mathbf{Z}_1^k\right)$ is a Gaussian density that represents the smoothed density given mode r_t^j, and $P\left\{r_t^j|\mathbf{Z}_1^k\right\}$ is the smoothed mode probability). Note that, the notations in Section 2.1.3 are extensively used in the following derivations. Using the total probability theorem and the Bayes' rule, the density $p\left(\mathbf{x}_t|r_t^j, \mathbf{Z}_1^k\right)$ can be simplified as follows:

$$\begin{aligned} p\left(\mathbf{x}_t|r_t^j, \mathbf{Z}_1^k\right) &= \int p\left(\mathbf{x}_t|\mathbf{x}_{t+1}, r_t^j, \mathbf{Z}_1^k\right) p\left(\mathbf{x}_{t+1}|r_t^j, \mathbf{Z}_1^k\right) d\mathbf{x}_{t+1} \\ &= \int \frac{p\left(\mathbf{x}_{t+1}|\mathbf{x}_t, r_t^j, \mathbf{Z}_1^t\right) p\left(\mathbf{x}_t|r_t^j, \mathbf{Z}_1^t\right)}{p\left(\mathbf{x}_{t+1}|r_t^j, \mathbf{Z}_1^t\right)} p\left(\mathbf{x}_{t+1}|r_t^j, \mathbf{Z}_1^k\right) d\mathbf{x}_{t+1} \\ &= p\left(\mathbf{x}_t|r_t^j, \mathbf{Z}_1^t\right) \int \frac{p\left(\mathbf{x}_{t+1}|r_t^j, \mathbf{Z}_1^k\right) p\left(\mathbf{x}_{t+1}|\mathbf{x}_t, r_t^j, \mathbf{Z}_1^t\right)}{p\left(\mathbf{x}_{t+1}|r_t^j, \mathbf{Z}_1^t\right)} d\mathbf{x}_{t+1} \end{aligned} \tag{8.2}$$

In the above, $p\left(\mathbf{x}_t|r_t^j, \mathbf{Z}_1^t\right)$ is the filtered mode conditioned density which is a Gaussian. The term $p\left(\mathbf{x}_{t+1}|\mathbf{x}_t, r_t^j, \mathbf{Z}_1^t\right)$ is a Gaussian density as it is corresponding to state transition density of model r_t^j where we used Markov property of state transition model. The term in the denominator is the normalizing constant. The term $p\left(\mathbf{x}_{t+1}|r_t^j, \mathbf{Z}_1^k\right)$ can be written as follows:

$$p\left(\mathbf{x}_{t+1}|r_t^j, \mathbf{Z}_1^k\right) = \sum_{i=1}^{N_r} p\left(\mathbf{x}_{t+1}|r_{t+1}^i, r_t^j, \mathbf{Z}_1^k\right) p\left\{r_{t+1}^i|r_t^j, \mathbf{Z}_1^k\right\} \tag{8.3}$$

Here, the condition on r_t^j in the first term can be ignored due to Markov property. Hence, the first term is the smoothed density at time $t+1$ given the mode r_{t+1}^i that is a Gaussian. The term $P\left\{r_{t+1}^i | r_t^j, \mathbf{Z}_1^k\right\}$ is the mixing probability which is found in the sequel. Therefore, (8.3) refers a Gaussian mixture which can be approximated to a single Gaussian using moment matching method. By replacing this approximated Gaussian density in (8.2), we will end up with the standard Kalman smoothing equation that will give a Gaussian density for $p\left(\mathbf{x}_t | r_t^j, \mathbf{Z}_1^k\right)$ as required in (8.1). Using the Bayes' rule, the mixing probability $P\left\{r_{t+1}^i | r_t^j, \mathbf{Z}_1^k\right\}$ can be written as follows:

$$P\left\{r_{t+1}^i | r_t^j, \mathbf{Z}_1^k\right\} = \frac{1}{d_j} P\left\{r_t^j | r_{t+1}^i, \mathbf{Z}_1^k\right\} P\left\{r_{t+1}^i | \mathbf{Z}_1^k\right\} \tag{8.4}$$

where $P\left\{r_t^j | r_{t+1}^i, \mathbf{Z}_1^k\right\}$ is the backward mode transition probability, $P\left\{r_{t+1}^i | \mathbf{Z}_1^k\right\}$ is the smoothed mode probability at time $t+1$, and d_j is the normalizing constant and given by

$$d_j = \sum_{h=1}^{N_r} P\left\{r_t^j | r_{t+1}^h, \mathbf{Z}_1^k\right\} P\left\{r_{t+1}^h | \mathbf{Z}_1^k\right\} \tag{8.5}$$

The backward mode transition probability can be written as follows:

$$\begin{aligned} P\left\{r_t^j | r_{t+1}^i, \mathbf{Z}_1^k\right\} &= P\left\{r_t^j | r_{t+1}^i, \mathbf{Z}_1^t\right\} \\ &= \frac{1}{e_i} P\left\{r_{t+1}^i | r_t^j, \mathbf{Z}_1^t\right\} P\left\{r_t^j | \mathbf{Z}_1^t\right\} \end{aligned} \tag{8.6}$$

In the above, the first line is due to Markov property of mode transition. Here, $P\left\{r_{t+1}^i | r_t^j, \mathbf{Z}_1^t\right\}$ is the forward mode transition probability, $P\left\{r_t^j | \mathbf{Z}_1^t\right\}$ is the filtered

mode probability at time t, and the normalizing constant e_i is given by

$$e_i = \sum_{h=1}^{N_r} P\left\{r_{t+1}^i | r_t^h, \mathbf{Z}_1^t\right\} P\left\{r_t^h | \mathbf{Z}_1^t\right\} \tag{8.7}$$

The smoothed mode probability $P\left\{r_t^j | \mathbf{Z}_1^k\right\}$ in (8.1) can be found as follows:

$$\begin{aligned} P\left\{r_t^j | \mathbf{Z}_1^k\right\} &\simeq P\left\{r_t^j | \mathcal{M}_{t+1|k}, \mathbf{Z}_1^t\right\} \\ &= \frac{1}{f} P\left\{\mathcal{M}_{t+1|k} | r_t^j, \mathbf{Z}_1^t\right\} P\left\{r_t^j | \mathbf{Z}_1^t\right\} \end{aligned} \tag{8.8}$$

The approximation in the first line is due to replacement of the mixture $\mathcal{M}_{t+1|k}$ which can be considered as sufficient statistics for the measurement set $\mathbf{Z}_{t+1}^k$. The normalizing constant f is given by

$$f = \sum_{i=1}^{N_r} P\left\{\mathcal{M}_{t+1|k} | r_t^i, \mathbf{Z}_1^t\right\} P\left\{r_t^i | \mathbf{Z}_1^t\right\} \tag{8.9}$$

The first term in (8.8) can be written as follows:

$$\begin{aligned} P\left\{\mathcal{M}_{t+1|k} | r_t^j, \mathbf{Z}_1^t\right\} &= \sum_{i=1}^{N_r} P\left\{\mathcal{M}_{t+1|k} | r_{t+1}^i, r_t^j, \mathbf{Z}_1^t\right\} P\left\{r_{t+1}^i | r_t^j, \mathbf{Z}_1^t\right\} \\ &= \sum_{i=1}^{N_r} P\left\{\mathcal{M}_{t+1|k} | r_{t+1}^i, r_t^j, \mathcal{M}_{t|t}\right\} P\left\{r_{t+1}^i | r_t^j\right\} \\ &= \sum_{i=1}^{N_r} P\left\{\mathcal{M}_{t+1|k}^i | \mathcal{M}_{t|t}^j\right\} P\left\{r_{t+1}^i | r_t^j\right\} \end{aligned} \tag{8.10}$$

Here, we again used the Markov property and sufficient statistic argument.

8.2 IMM Smoother Algorithms

Finally, we summarize the novel backward smoothing method for multiple model approach in Section 8.2.1. The results of this proposed method are compared with that of existing augmented smoothing method [16], which is reviewed in Section 8.2.2.

8.2.1 Backward IMM Smoother Algorithm

In the backward smoothing, we find Gaussian mixture that represents the smoothed density at time t given measurement up to time k, where $k - L \leq t \leq k$ and L is the time lag. Given the filtered mixture $\mathcal{M}_{t|t}$ at time t and smoothed mixture $\mathcal{M}_{t+1|k}$ at time $t+1$, the backward recursion can be summarized as follows:

- Calculation of backward transition probability: Using (8.6), the backward mode transition probability can be written as

$$b_{ij} = \frac{1}{e_i} p_{ji} \mu_{t|t}^{j} \quad i, j = 1, \ldots, N_r \tag{8.11}$$

  where $e_i = \sum_{l=1}^{N_r} p_{li} \mu_{t|t}^{l}$

- Calculation of the backward mixing probability: The mixing probability is given by

$$\mu_{t+1|k}^{i|j} = \frac{1}{d_j} b_{ij} \mu_{t+1|k}^{i} \quad i, j = 1, \ldots, N_r \tag{8.12}$$

  where $d_j = \sum_{l=1}^{N_r} b_{lj} \mu_{t+1|k}^{l}$

- Mixing: The mean and the covariance matrix for the j^{th} mode-matched smoother

are given by

$$\mathbf{m}_{t+1|k}^{0j} = \sum_{i=1}^{N_r} \mu_{t+1|k}^{i|j} \mathbf{m}_{t+1|k}^{i} \tag{8.13}$$

$$P_{t+1|k}^{0j} = \sum_{i=1}^{N_r} \mu_{t+1|k}^{i|j} \left\{ P_{t+1|k}^{i} + \left[\mathbf{m}_{t+1|k}^{i} - \mathbf{m}_{t+1|k}^{0j} \right] \left[\mathbf{m}_{t+1|k}^{i} - \mathbf{m}_{t+1|k}^{0j} \right]^T \right\} \tag{8.14}$$

- Mode-matched smoothing: The mean and the covariance in (8.13) and (8.14) are used as input to the mode-matched smoother r_t^j. The smoothed mode conditioned mean and covariance matrix are given by

$$\mathbf{m}_{t|k}^{j} = \mathbf{m}_{t|t}^{j} + A_{t|k}^{j} \left(\mathbf{m}_{t+1|k}^{0j} - \mathbf{m}_{t+1|t}^{j} \right) \tag{8.15}$$

$$P_{t|k}^{j} = P_{t|t}^{j} - A_{t|k}^{j} \left(P_{t+1|k}^{0j} - P_{t+1|t}^{j} \right) {A_{t|k}^{j}}^T \tag{8.16}$$

where $\mathbf{m}_{t+1|t}^{j}$ and $P_{t+1|t}^{j}$ are the predicted mean and covariance matrix corresponding to j^{th} mode, and $A_{t|k}^{j}$ is the smoothing gain and given by [79]

$$A_{t|k}^{j} = P_{t|t}^{j} F_{jk}^T \left(P_{t+1|t}^{j} \right)^{-1} \tag{8.17}$$

Here, F_{jk}^T is the state transition matrix of the j^{th} model.

- Mode probability smoothing: The smoothed mode probability is given by

$$\mu_{t|k}^{j} = \frac{1}{f} \Lambda_{t|k}^{j} \mu_{t|t}^{j} \tag{8.18}$$

where the normalizing constant $f = \sum_{i=1}^{N_r} \Lambda_{t|k}^{i} \mu_{t|t}^{i}$ and

$$\Lambda_{t|k}^{j} = \sum_{i=1}^{N_r} p_{ji} \mathcal{N} \left(\mathbf{m}_{t+1|k}^{i}; \mathbf{m}_{t+1|t}^{j}, P_{t+1|t}^{j} \right) \tag{8.19}$$

- Estimate: Finally the estimate and corresponding covariance matrix are found using moment matching. That is,

$$\hat{\mathbf{x}}_{t|k} = \sum_{i=1}^{N_r} \mu_{t|k}^{j} \mathbf{m}_{t|k}^{j} \tag{8.20}$$

$$P_{t|k} = \sum_{i=1}^{N_r} \mu_{t|k}^{j} \left\{ P_{t|k}^{j} + \left[\mathbf{m}_{t|k}^{j} - \hat{\mathbf{x}}_{t|k} \right] \left[\mathbf{m}_{t|k}^{j} - \hat{\mathbf{x}}_{t|k} \right]^{T} \right\} \tag{8.21}$$

The algorithm is also summarized in Figure 8.1, which resembles IMM filter in the backward direction (Figure 2.1).

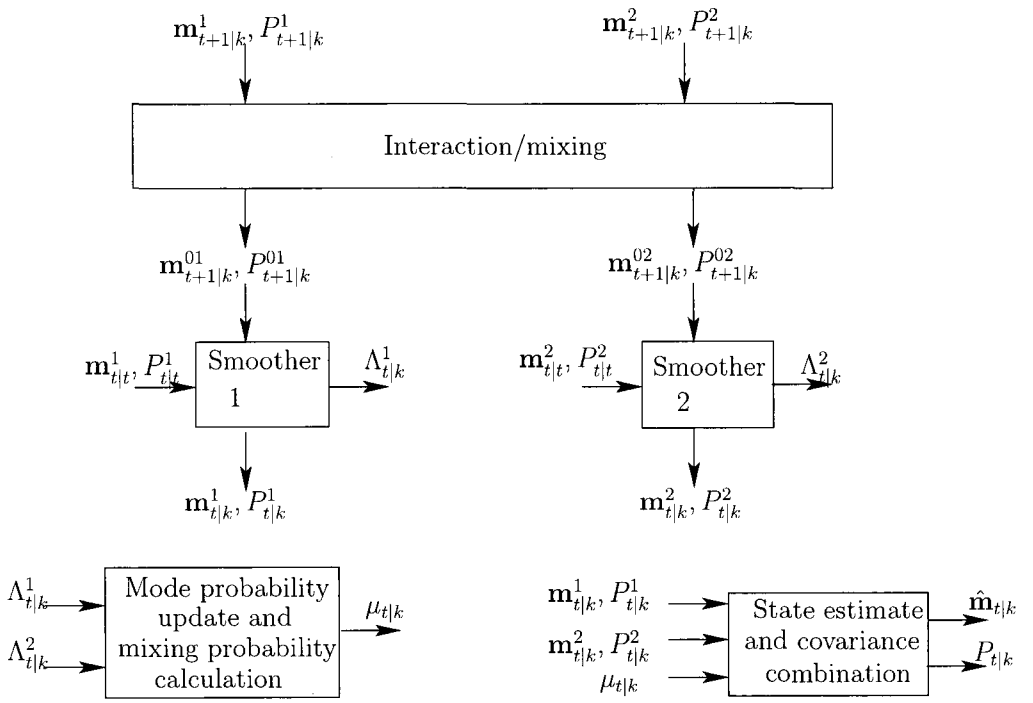


Figure 8.1: IMM smoother with two models

8.2.2 Augmented IMM Smoother Algorithm

In the augmented method [16], an augmented state vector is considered as follows:

$$\tilde{\mathbf{x}}_k = \left[\tilde{\mathbf{x}}_k^{(0)^T} \ \tilde{\mathbf{x}}_k^{(1)^T} \ \dots \ \tilde{\mathbf{x}}_k^{(L)^T}\right]^T \tag{8.22}$$

where

$$\tilde{\mathbf{x}}_k^{(0)} = \mathbf{x}_k, \ \tilde{\mathbf{x}}_k^{(1)} = \mathbf{x}_{k-1}, \ \dots, \ \tilde{\mathbf{x}}_k^{(L)} = \mathbf{x}_{k-L}$$

The associated covariance matrix is defined as

$$\tilde{P}_{k|k} = \begin{bmatrix} \tilde{P}_{k|k}^{(0,0)} & \tilde{P}_{k|k}^{(0,1)} & \dots & \tilde{P}_{k|k}^{(0,L)} \\ \tilde{P}_{k|k}^{(1,0)} & \tilde{P}_{k|k}^{(1,1)} & \dots & \tilde{P}_{k|k}^{(1,L)} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{P}_{k|k}^{(L,0)} & \tilde{P}_{k|k}^{(L,1)} & \dots & \tilde{P}_{k|k}^{(L,L)} \end{bmatrix}$$

where

$$\tilde{P}_{k|k}^{(i,j)} = \mathbb{E}\left[\left(\tilde{\mathbf{x}}_k^{(i)} - \hat{\tilde{\mathbf{x}}}_{k|k}^{(i)}\right)\left(\tilde{\mathbf{x}}_k^{(j)} - \hat{\tilde{\mathbf{x}}}_{k|k}^{(j)}\right)^T\right] \tag{8.23}$$

with

$$\hat{\tilde{\mathbf{x}}}_{k|k}^{(i)} = \mathbb{E}\left[\tilde{\mathbf{x}}_k^{(i)} | \mathbf{z}_{1:k}\right] = \hat{\mathbf{x}}_{k-i|k} \tag{8.24}$$

which is the smoothed estimate at time $k-1$ given measurements up to time k. The augmented system is defined as follows:

$$\begin{bmatrix} \tilde{\mathbf{x}}_k^{(0)} \\ \tilde{\mathbf{x}}_k^{(1)} \\ \tilde{\mathbf{x}}_k^{(2)} \\ \vdots \\ \tilde{\mathbf{x}}_k^{(L)} \end{bmatrix} = \begin{bmatrix} F_k(r_k) & 0 & \dots & 0 & 0 \\ I & 0 & \dots & 0 & 0 \\ 0 & I & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & I & 0 \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}}_{k-1}^{(0)} \\ \tilde{\mathbf{x}}_{k-1}^{(1)} \\ \tilde{\mathbf{x}}_{k-1}^{(2)} \\ \vdots \\ \tilde{\mathbf{x}}_{k-1}^{(L)} \end{bmatrix} + \begin{bmatrix} \mathbf{v}_{k-1}(r_k) \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

where $F_k(r_k)$ is the state transition matrix currently in effect (2.20). The augment measurement model is given by

$$\mathbf{z}_k = [H_k(r_k)\ 0\ \dots\ 0\ 0] \begin{bmatrix} \tilde{\mathbf{x}}_k^{(0)} \\ \tilde{\mathbf{x}}_k^{(1)} \\ \tilde{\mathbf{x}}_k^{(2)} \\ \vdots \\ \tilde{\mathbf{x}}_k^{(L)} \end{bmatrix} + \mathbf{n}_k(r_k)$$

where $H_k(r_k)$ is the measurement matrix currently in effect (2.21). The augmented state (8.22) is propagated using IMM recursion in Section 2.1.3. Here, it is assumed that there is no model switching within the augmented block. Finally the smoothed state estimates at time $k-L$ is given by

$$\hat{\mathbf{x}}_{k-L|k} = \hat{\tilde{\mathbf{x}}}_{\mathbf{k}|\mathbf{k}}^{(\mathbf{L})} \tag{8.25}$$

The associated error covariance matrix

$$P_{k-L|k} = \tilde{P}_{k|k}^{(L,L)} \tag{8.26}$$

The smoothed mode probability can be written as

$$\mu_{k-L|k}^{j} = \frac{1}{c_{k-L|k}} \mathcal{N}\left(\hat{\mathbf{x}}_{k-L|k}; \mathbf{m}_{k-L|k-L}^{j}, P_{k-L|k-L}^{j}\right) \mu_{k-L|k-L}^{j} \tag{8.27}$$

where normalizing constant $c_{k-L|k}$ is given by

$$c_{k-L|k} = \sum_{j=1}^{N_r} \mathcal{N}\left(\hat{\mathbf{x}}_{k-L|k}; \mathbf{m}_{k-L|k-L}^{j}, P_{k-L|k-L}^{j}\right) \mu_{k-L|k-L}^{j} \tag{8.28}$$

8.3 Simulation

In this section, the results of the simulation studies using the novel IMM smoothing algorithm are presented. In the simulation study, we consider a two-dimensional scenario with a maneuvering target as shown in Figure 8.2. With initial position at (29.5, 35) km, the target moves southwest direction for 10s at a nearly constant velocity with velocity of 330 ms^{-1}, before executing a $3g$ coordinated turn in the counter-clockwise for 12s. Then it moves southward for 4s, followed by a clockwise $3g$ coordinated turn for 12s. Finally, it moves southwest direction for 12s at a nearly constant velocity with velocity of 330 ms^{-1}. The sensor is located at the origin, and provides range and bearing measurements with measurement noise standard deviations of 15m and 0.002 rad. The measurements are available at discrete sampling interval T = 1s. The IMM filter consists of two models with transition probability

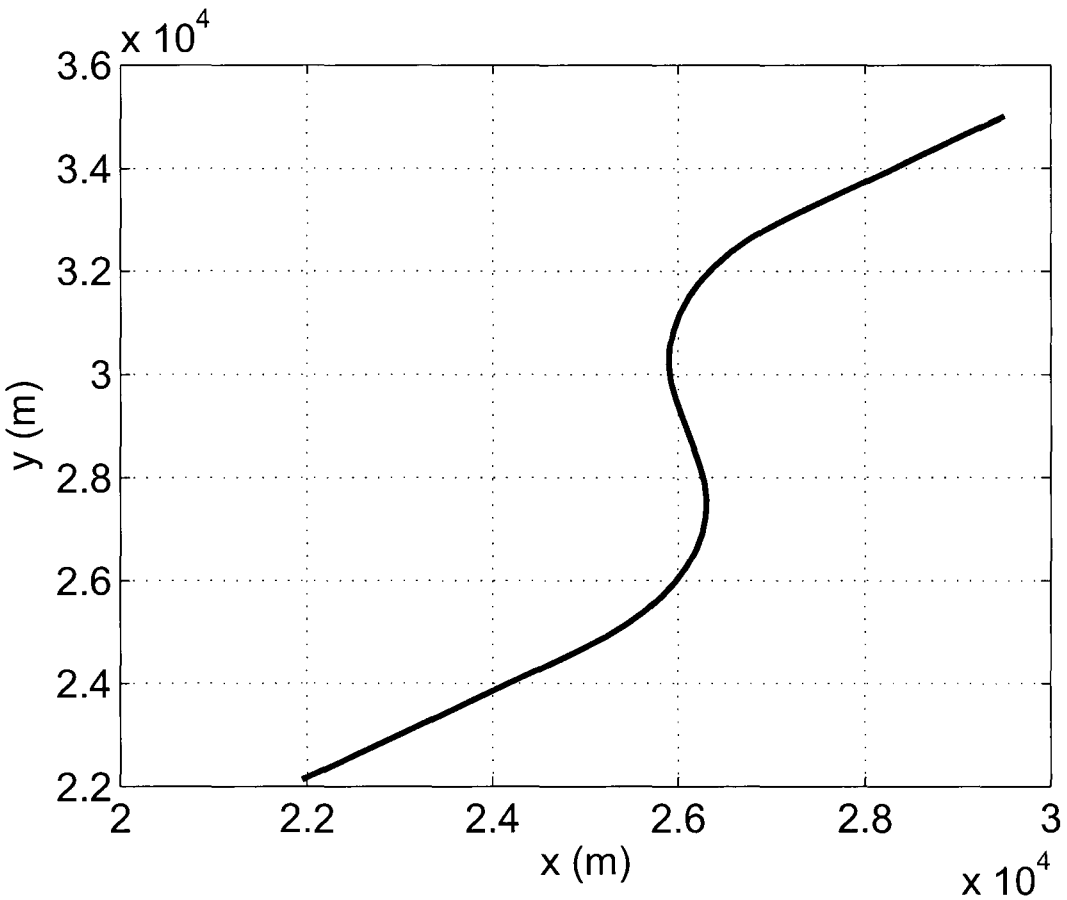


Figure 8.2: Ground truth of the maneuvering target scenario

matrix as follows:

$$[h_{vu}] = \begin{bmatrix} 0.95 & 0.05 \\ 0.1 & 0.9 \end{bmatrix} \tag{8.29}$$

The first model is a constant velocity with $\sigma^2 = 1$ m^{2}s^{-3}. That is,

$$\mathbf{x}_k = F_{1,k}\mathbf{x}_{k-1} + \mathbf{v}_{1,k-1} \tag{8.30}$$

where $\mathbf{x}_k = [x_k \ \dot{x}_k \ y_k \ \dot{y}_k]^T$ and the target transition matrix $F_{1,k}$ is given by

$$F_{1,k} = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

and the process noise $\mathbf{v}_{1,k-1}$ has a normal distribution with mean zero and covariance matrix $Q_{1,k-1}$, which is given by

$$Q_{1,k-1} = \sigma_1^2 \begin{bmatrix} T^3/3 & T^2/2 & 0 & 0 \\ T^2/2 & T & 0 & 0 \\ 0 & 0 & T^3/3 & T^2/2 \\ 0 & 0 & T^2/2 & T \end{bmatrix}$$

where $\sigma_1 = 0.5$ m^{2}s^{-3}. The second one is a constant acceleration model and given by

$$\mathbf{x}_k = F_{2,k}\mathbf{x}_{k-1} + \mathbf{v}_{2,k-1} \tag{8.31}$$

where $\mathbf{x}_k = [x_k \ \dot{x}_k \ \ddot{x}_k \ y_k \ \dot{y}_k \ \ddot{y}_k]^T$ is the augmented state vector, which consists of target position $[x_k, \ y_k]^T$, target velocity $[\dot{x}_k, \dot{y}_k]^T$ and target acceleration $[\ddot{x}_k, \ddot{y}_k]^T$ at time step, and target transition matrix $F_{2,k}$ is given by

$$F_{2,k} = \begin{bmatrix} 1 & T & T^2/2 & 0 & 0 & 0 \\ 0 & 1 & T & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & T & T^2/2 \\ 0 & 0 & 0 & 0 & 1 & T \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

and the process noise $\mathbf{v}_{2,k-1}$ has a normal distribution with mean zero and covariance matrix $Q_{2,k-1}$, which is given by

$$Q_{2,k-1} = \sigma_2^2 \begin{bmatrix} \frac{1}{20}T^5 & \frac{1}{8}T^4 & \frac{1}{6}T^3 & 0 & 0 & 0 \\ \frac{1}{8}T^4 & \frac{1}{3}T^3 & \frac{1}{2}T^2 & 0 & 0 & 0 \\ \frac{1}{6}T^3 & \frac{1}{2}T^2 & T & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{20}T^5 & \frac{1}{8}T^4 & \frac{1}{6}T^3 \\ 0 & 0 & 0 & \frac{1}{8}T^4 & \frac{1}{3}T^3 & \frac{1}{2}T^2 \\ 0 & 0 & 0 & \frac{1}{6}T^3 & \frac{1}{2}T^2 & T \end{bmatrix}$$

where $\sigma_2^2 = 3 \ \mathrm{m^2 s^{-5}}$.

Results from 100 Monte Carlo runs are discussed below. Figure 8.3 shows the model-switching property of the IMM filter and both smoothing methods with time lag of 3. The RMSE for position and velocity are provided in Figures 8.4 and 8.5, respectively. They clearly demonstrate the improved performance of the proposed IMM smoothing method over both IMM filter and augmented IMM smoother. Figure

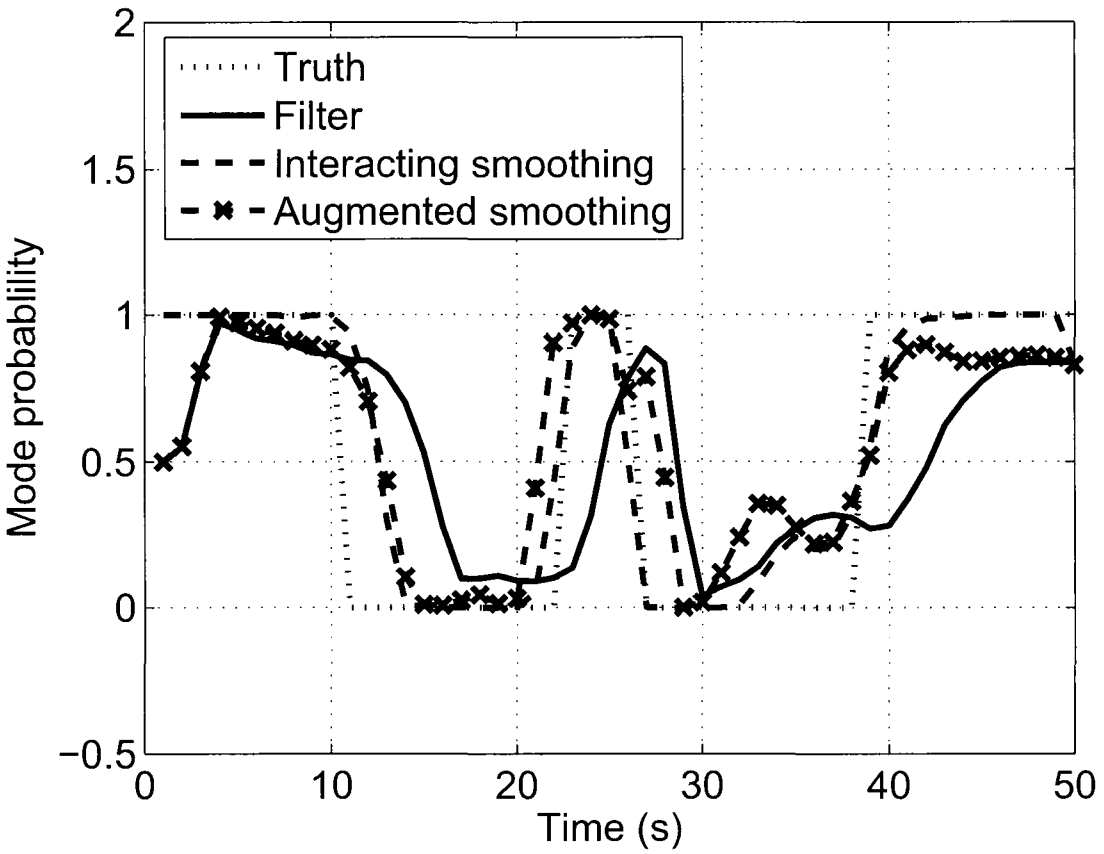


Figure 8.3: Mode switching

8.6 shows Normalized Estimation Error Squared (NEES) for all three methods. The comparisons of the NEES values also confirms that the proposed IMM smoother is more consistent than IMM filter and augmented IMM smoother.

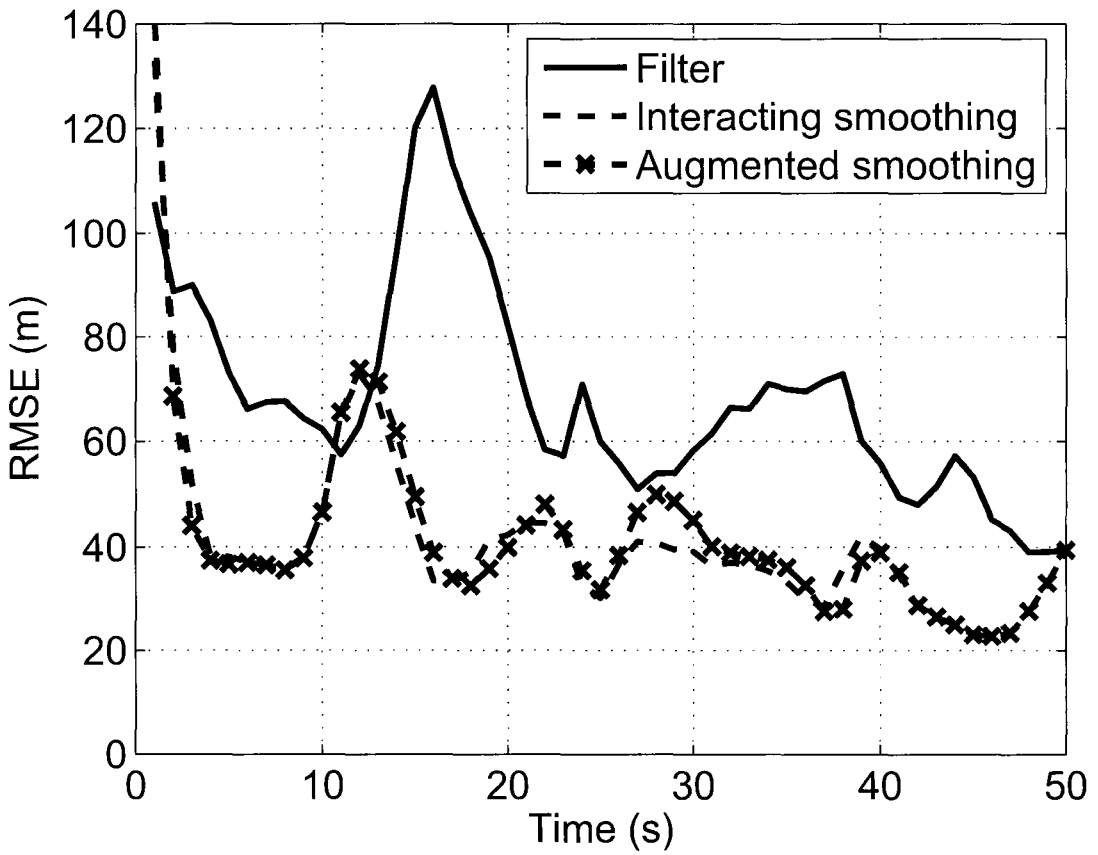


Figure 8.4: Position RMSE

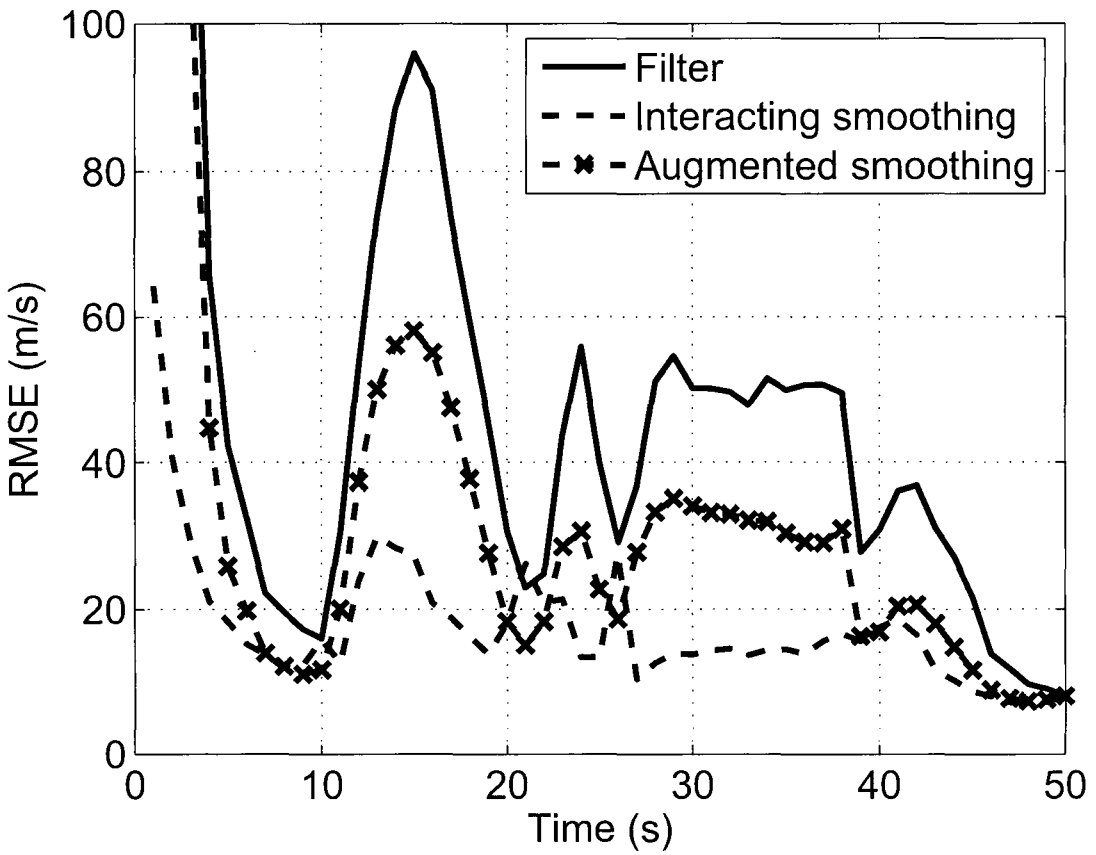


Figure 8.5: Velocity RMSE

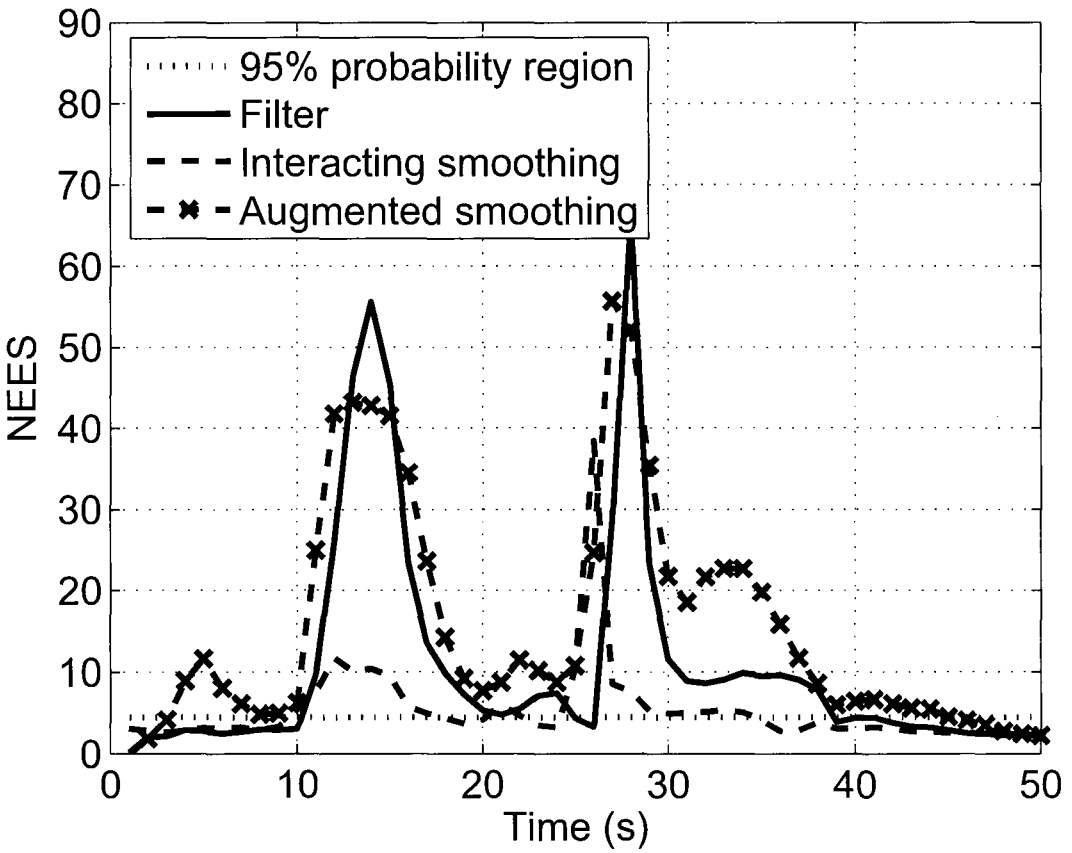


Figure 8.6: NEES

Chapter 9

Conclusions

In this thesis, we have considered the problem of tracking multiple targets using retrodiction (smoothing). The main focus of the thesis was to develop RTS type smoothing methods for various state estimators. Filtering typically produces the best estimates of the target state based on all measurements up to current estimation time. Smoothing, which uses measurements beyond the current estimation time, provides better estimates of target states. This delayed but improved estimates were achieved with additional computational load to incorporate information from measurements beyond the current estimation time to current estimates.

We developed a smoothing algorithm to improve the capability of PHD based state estimator. The proposed method involves a forward multitarget filtering using the standard PHD filter recursion and then a backward smoothing recursion. The PHD filter, which propagates the first order statistical moment of the multitarget state density, is a computationally efficient MTT algorithm. The backward smoothing recursion is performed with a novel recursive formula, which was derived using

the physical-space approach. The resulting backward recursion incorporates intensity for surviving targets as well as disappearing targets. Further, we proposed SMC implementation of the backward smoothing method. It was shown that this SMC implementation requires much computational effort to compute smoothed particle weights. To mitigate this, we introduced a fast implementation, which uses the N-body algorithm. However, this fast method requires that target transition density should be defined in metric space. It was shown that most commonly used transition models can be converted into a function defined in a metric space using simple transformations.

Then, we proposed a smoothing method for MMPHD based state estimator. The MMPHD filter is a natural extension of the PHD filter for maneuvering target tracking. Here, the mode dependent PHD is propagated recursively. The mode dependent filtered PHD is reprocessed in back direction using a new smoothing equation. The SMC implementation of the backward recursion, which involves continuous density as well as probability mass function due to discrete variable for mode, was also proposed. A fast method was provided for a special case where targets switch between two dynamic models and have symmetric mode transition matrix.

Next, we proposed a smoothing method for GMPHD state estimator using multiple sensors. In the Gaussian mixture implementation, the PHD surface is represented by a mixture of Gaussian functions. Under linear Gaussian assumption, the PHD filter can be implemented using closed-form recursion. This can be extended to nonlinear systems by using the EKF or the UKF. The proposed GMPHD smoothing method was applied for multisensor system. In the case of multisensor systems, a sequential update of the PHD has been suggested in the literature. However, this sequential update is susceptible to the imperfections in the last sensor. To mitigate

this, a parallel update for GMPHD filter was proposed. The resulting filter outputs were further improved using a novel closed-form backward smoothing recursion.

Finally, we developed a novel smoothing method for Kalman based IMM estimator for tracking maneuvering target. The proposed method involves forward filtering followed by backward smoothing while maintaining the fundamental spirit of the IMM. The forward filtering is performed using the standard IMM recursion, while the backward smoothing is performed using a novel interacting smoothing recursion. This backward recursion mimics the IMM estimator in the backward direction, where each mode conditioned smoother uses the standard Kalman smoothing recursion. A comparison studies were performed with existing augmented smoothing method.

Appendix A

K-mean Algorithm

The K-mean algorithm is clustering algorithm to partition N objects into K clusters, where $K < N$. It attempts to find the centers of natural clusters in the data [43]. The main idea is to define K centroids, one for each cluster. The centroids are found such that they minimize total intra-cluster variance as follows [37]:

$$V = \sum_{i=1}^{K} \sum_{j=1}^{N} I_{m(\mathbf{x}^{(j)})}(i) ||\mathbf{x}^{(j)} - \mathbf{c}_i||^2 \tag{A.1}$$

where $||\mathbf{x}^{(j)} - \mathbf{c}_i||$ is a chosen distance between a data point $\mathbf{x}^{(j)}$ and the cluster center $\mathbf{c}_i$. The algorithm (Figure A.1) is as follows:

- Initialize centroids $\mathbf{c}_i, i = 1, \ldots, K$ (e.g., random selection of $\{\mathbf{x}^{(j)}\}$)
- For each object $\mathbf{x}^{(j)}$ find membership

$$m(\mathbf{x}^{(j)}) = \arg \min_{i \in \{1,\ldots,K\}} ||\mathbf{x}^{(j)} - \mathbf{c}_i|| \tag{A.2}$$

- While m has changed

- For each $i \in \{1, \ldots, K\}$ compute the centroid $\mathbf{c}_i$ of $\{\mathbf{x}^{(j)} | m(\mathbf{x}^{(j)}) = i\}$
- For each object $\mathbf{x}^{(j)}$ find membership

$$m(\mathbf{x}^{(j)}) = \arg \min_{i \in \{1,\ldots,K\}} ||\mathbf{x}^{(j)} - \mathbf{c}_i|| \tag{A.3}$$

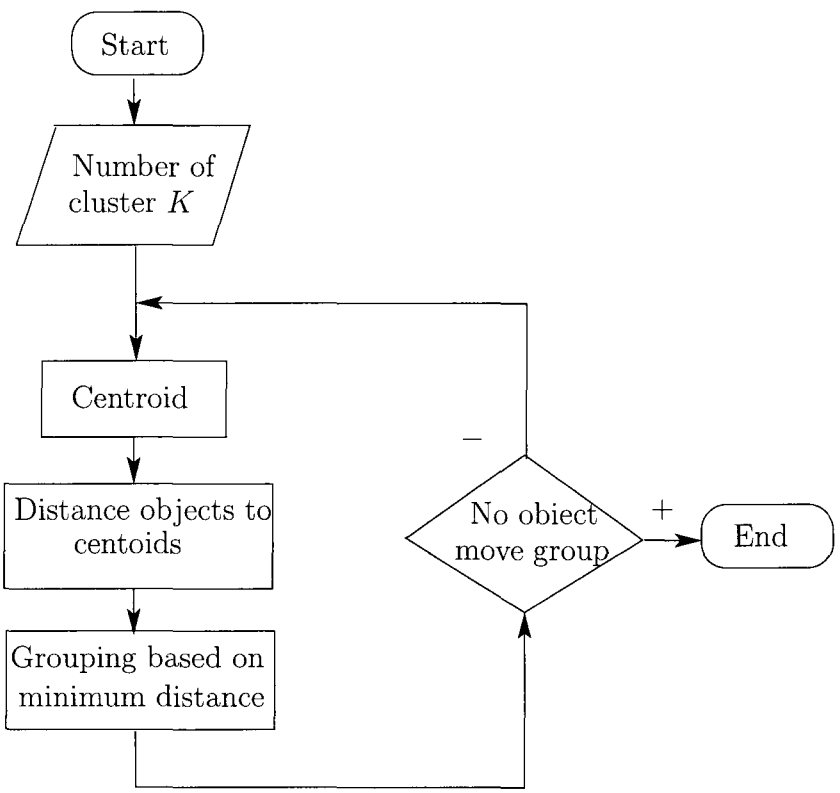


Figure A.1: K-mean algorithm

Appendix B

Gaussian Mixture Reduction

A simple pruning procedure can be used to reduce the number of Gaussian components propagated to the next time step. A good approximation to the Gaussian mixture intensity

$$D(\mathbf{x}) = \sum_{i=1}^{J} w^i \mathcal{N}\left(\mathbf{x}; \mathbf{m}^i, P^i\right) \tag{B.4}$$

can be obtained by truncating components that have weak weights w^i. This can be done by discarding those with weights below some preset threshold, or by keeping only a certain number of components with strongest weights. Moreover, some of the Gaussian components are so close together that they could be accurately approximated by a single Gaussian. Hence, in practice these components can be merged into one. These ideas lead to the simple heuristic pruning algorithm shown below.

Given $\{w^i, \mathbf{m}^i, P^i\}_{i=1}^{J}$, a truncation threshold ϱ, a merging threshold U, and a maximum allowable number of Gaussian terms J_{max}.

Set $l = 0$, and $I = \{i = 1, \ldots, J | w^i > \varrho\}$.

repeat

$l := l + 1$.

$j := \arg\max_{i \in I} w^i$.

$L := \left\{ i \in I | \left(\mathbf{m}^i - \mathbf{m}^j\right)^T \left(P^i\right)^{-1} \left(\mathbf{m}^i - \mathbf{m}^j\right) \leq U \right\}$.

$\tilde{w}^l = \sum_{i \in L} w^i$.

$\tilde{\mathbf{m}}^j = \frac{1}{\tilde{w}^l} \sum_{i \in L} w^i \mathbf{m}^i$.

$\tilde{P}^l = \frac{1}{\tilde{w}^l} \sum_{i \in L} w^i \left(P^i + \left(\tilde{\mathbf{m}}^l - \mathbf{m}^i\right) \left(\tilde{\mathbf{m}}^l - \mathbf{m}^i\right)^T \right)$.

$I := I \setminus L$

until I=ø

if $l > J_{max}$ then replace $\{\tilde{w}^i, \tilde{\mathbf{m}}^i, \tilde{P}^i\}_{i=1}^l$ by those of the J_{max} Gaussian with largest weights.

Output $\{\tilde{w}^i, \tilde{\mathbf{m}}^i, \tilde{P}^i\}_{i=1}^l$ as pruned Gaussian components.

Bibliography

[1] Alspach, D. L. and Sorenson, H. W., "Nonlinear Bayesian estimation using Gaussian sum approximations," *IEEE Trans. Automatic Control,* vol. AC-17, no. 4, pp. 439-448, 1972.

[2] Anderson, B. and Moore, J., *Optimal Filtering.* Englewood, NJ: Prentice Hall, 1979.

[3] Arulampalam, M. S. and Ristic, B., "Comparison of the particle filter with range-parameterised and modified polar EKFs for angle-only tracking," in *Proc. SPIE on Signal and Data Processing of Small Targets*, vol. 4048, pp. 288–299, Orlando, FL, Apr. 2000.

[4] Arulampalam, M. S., Maskell, S., Gordon, N., and Clapp, T., "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Processing,* vol. 50, no. 2, pp. 174–188, Feb. 2002.

[5] Bar-Shalom, Y. and Fortmann, T. E., *Tracking and Data Association.* Boston/London: Academic Press, 1988.

[6] Bar-Shalom, Y. and Li, X. R., *Multitarget-Multisensor Tracking: Principles and Techniques.* Storrs: YBS Publishing, 1995.

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

[8] Blackman, S. and Popoli, R., *Design and Analysis of Modern Tracking Systems.* Boston, MA: Artech House, 1999.

[9] Blackman, S., "Multiple hypothesis tracking for multiple target tracking," *IEEE A & E Systems Magazine,* vol. 19, no. 1, pp. 5–18, Jan. 2004.

[10] Blom, H. A. P., "A sophisticated tracking algorithm for air traffic control surveillance radar data," in *Proc. Int. Conf. Radar,* pp. 393–398, Paris, France, May 1984.

[11] Blom, H. A. P. and Bar-Shalom, Y., "The interacting multiple model algorithm for systems with Markovian switching coefficients," *IEEE Trans. Automatic Control,* vol. 33, no. 8, pp. 780–783, Aug. 1988.

[12] Blom, H. A. P., and Bar-Shalom, Y., "Time-reversion of a hybrid state stochastic difference system with a jumb-linear smoothing application", *IEEE Trans. Information Theory*, vol. 36, no. 4, pp. 836–847, Jul. 1990.

[13] Bresler, Y., "Two-filter formula for discrete-time non-linear Bayesian smoothing," *International Journal of Control,* vol. 43, no. 2, pp. 629-641, 1986.

[14] Briers, M., Doucet, A., and Maskell, S., "Smoothing algorithms for state-space models," Dept. Eng., Univ. Cambridge, Cambridge, U.K., Tech. Rep. TR-CUEDF-INFENG 498, 2004.

[15] Brown, R. G. and Hwang, P. Y. C., *Introduction to Random Signals and Applied Kalman Filtering.* New York: John Wiley & Sons, 1997.

[16] Chen, B., and Tugnait, J. K., "Interacting multiple model fixed-lag smoothing algorithm for Markovian switching systems", *IEEE Trans. Aerospace and Electronic Systems*, vol. 36, no. 1, pp. 432–500, Jan. 2000.

[17] Chen, B. and Tugnait, J. K., "Multisensor tracking of a maneuvering target in clutter using IMMPDA fixed-lag smoothing," *IEEE Trans. Aerospace and Electronic Systems,* vol. 36, no. 3, pp. 983–991, Jul. 2000.

[18] Clark, D. and Vo, B.-N., "Convergence analysis of the Gaussian mixture PHD Filter," *IEEE Trans. Signal Processing,* vol. 55, no. 4, pp. 1204–1212, Apr. 2007.

[19] Cox, H. "On the estimation of state variables and parameters for noisy dynamic systems," *IEEE Transactions on Automatic Control,* vol. 9, no. 1, pp. 5-12, 1964.

[20] de Jong, P., "Smoothing and interpolation with the state-space model," *Journal of the American Statistical Association,* vol. 84, no. 408, pp. 1085-1088, 1991.

[21] Doucet, A., Godsill, S. J., and Andrieu, C., "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and Computing,* vol. 10, no. 3, pp. 197–208, 2000.

[22] Doucet, A., De Freitas, F. G., and Gordon, N. J., Eds., *Sequential Monte Carlo Methods in Practice.* New York, NY: Springer-Verlag, 2001.

[23] Drummond, O. E., "Multiple target tracking with multiple frame, probabilistic data association," *Signal and Data Processing of Small Targets,* vol. SPIE 1954, 1993.

[24] Drummond, O. E., "Multiple sensor tracking with multiple frame, probabilistic

data association," *Signal and Data Processing of Small Targets,* vol. SPIE 2561, 1995.

[25] Drummond, O. E., "Target tracking with retrodicted discrete probabilities," *Signal and Data Processing of Small Targets,* vol. SPIE 3163, 1997.

[26] Erdinc, O., Willett, P., and Bar-Shalom, Y., "Probability hypothesis density filter for multitarget multisensor tracking," in *Proc. Int. Conf. Information Fusion,* pp. 146–153, Jul. 2005.

[27] Erdinc, O., Willett, P., and Bar-Shalom, Y., "A physical-space approach for the probability hypothesis density and cardinalized probability hypothesis density filters," *Proc. Signal and Data Processing of Small Targets,* vol. 6236, Apr. 2006.

[28] Fong, W., Godsill, S.J., Doucet, A., and West, M., "Monte Carlo smoothing with application to audio signal enhancement," *IEEE Trans. on Signal Processing,* vol. 50, no. 2, pp. 438–449, Feb. 2002.

[29] Fortmann, T. E., Bar-Shalom, Y., and Scheffe, M., "Sonar tracking of multiple targets using joint probabilistic data association," *IEEE Journal of Oceanic Engineering,* vol. OE-8, pp. 173–184, Jul. 1983.

[30] Gelb, A., *Applied Optimal Estimation.* Cambridge, MA: MTI Press, 1974.

[31] Godsill, S. J., Doucet, A., and West, M., "Monte Carlo smoothing for nonlinear time series," *J. the American Statistical Association,* vol. 99, pp. 156–168, Mar. 2004.

[32] Goodman, I., Mahler, R., and Nguyen, H., *Mathematics of Data Fusion.* Norwell, MA: Kluwer Academic Publishing Co., 1997.

[33] Gordon, N. J., Salmond, D. J., and Smith, A. F. M., "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE Proc. F,* vol. 140, no. 2, pp. 107–113, Apr 1993.

[34] Gray, A. and Moore, A., "N-Body problems in statistical learning," *Advances in Neural Information Processing Systems,* vol. 13, T. K. Leen, T. G. Dietterich, and V. Tresp, Ed., 2001.

[35] Grewal, M. S. and Andrews, A. P., *Kalman Filtering: Theory and Practice.* Englewood Cliffs, NJ: Prentice Hall, 1993.

[36] Gupta, M. M., Jin, L., and Homma, N., *Static and Dynamic Neural Networks: From Fundamentals to Advanced Theory.* New York: Wiley, 2003.

[37] Hartigan, J. A., *Clustering Algorithms*, New York, NY: John Wiley, 1975.

[38] Harvey, A. C., *Forecasting, Structural Time Series Models and the Kalman Filter.* New York: Cambridge University Press, 1989.

[39] Helmick, R. E., Blair, W. D., and Hoffman, S. A., "Fixed-interval smoothing for Markovian switching systems", *IEEE Trans. Information Theory*, vol. 41, no. 6, pp. 1845–1855, Nov. 1995.

[40] Helmick, R. E., Blair, W. D., and Hoffman, S. A., "One step fixed-lag smoothers for Markovian switching systems," *IEEE Trans. Automatic Control,* vol. 41, no. 7, pp. 1051–1056, Jul. 1996.

[41] Hoffman, J. and Mahler, R., "Multitarget miss distance via optimal assignment," *IEEE Trans. Systems, Man, and Cybernetics–Part A*, vol. 34, no. 3, pp. 327–336, May 2004.

[42] Hürzeler, M. and Künsch, H. R., "Monte Carlo approximations for general state space models," *J. Computational and Graphical Statistics,* vol. 7, no. 2, pp. 175–193, Jun. 1998.

[43] Jain, A. K., Murty, M. N., and Flynn, P. J., "Data clustering: A review," *ACM Computing Surveys,* vol. 31, no. 3, pp. 264-323, 1999.

[44] Jazwinski, A. H., *Stochastic Process and Filtering Theory,* New York: Academic, 1970.

[45] Jilkov, V. P., Li, X. R., and Lu, L., "Performance enhancement of the IMM estimation by smoothing", in *Proc. Int. Conf. Information Fusion,* vol. 1, pp. 713–720, 2002.

[46] Johansen, A., Singh, S., Doucet, A., and Vo. B-N., "Convergence of the SMC implementation of the PHD filter," University of Cambridge, Tech. Rep. CUED/F-INFENG/TR-517, 2005.

[47] Julier, S. J. and Uhlmann, J. K., "Unscented filtering and nonlinear estimation," in *Proc. IEEE* vol. 92, no. 3, pp. 401–422, 2004

[48] Kalman, R., "A new approach to linear filtering and prediction problems," *Trans. the ASMEJournal of Basic Engineering,* vol. 82, no. Series D, pp. 35-45, 1960.

[49] Kirubarajan, T. and Bar-Shalom, Y., "Kalman filter versus IMM estimator: When do we need the latter?," *IEEE Trans. on Aerospace and Electronic Systems,* vol. 39, no. 4, pp. 1452-1457, Oct. 2003.

[50] Kitagawa, G. "The two-filter formula for smoothing and an implementation of

the gaussian-sum smoother," *Annals of the Institute of Statistical Mathematics,* vol. 46, no. 4, pp. 605-623, 1994.

[51] Kitagawa, G., "Monte Carlo filter and smoother for non-Gaussian non-linear state space models," *Journal of Computational and Graphical Statistics,* vol. 5, no. 1, pp. 1-25, 1996.

[52] Klaas, M., Briers, M., de Freitas, N., Doucet, A., Maskell, S., and Lang, D., "Fast particle smoothing: if I had a million particles," *Proc. 23rd Int. Conf. Machine Learning,* Pittsburgh, PA, pp. 481–488, 2006.

[53] Koch, W., "Retrodiction for Bayesian multiple hypothesis/multiple target tracking in densely cluttered environment," *Signal and Data Processing of Small Targets,* vol. SPIE 2759, 1996.

[54] Koch, W., "Fixed-interval retrodiction approach to Bayesian IMM-MHT for maneuvering multiple targets," *IEEE Trans. Aerospace and Electronic Systems,* vol. 36, no. 1, pp. 2–14, Jan. 2000.

[55] Kohn, R. and Ansley, C. F., "Filtering and smoothing algorithms for state space models," *Computers Mathematical Applications,* vol. 18, no. 6, pp. 515-528, 1989.

[56] Kurien, T., "Issues in the design of practical multitarget tracking algorithms," in Y. Bar-Shalom, Ed., *Multitarget-Multisensor Tracking: Advanced Applications.* pp. 43-83, Boston, MA: Artech House 1990.

[57] Li, X. R., and Jilkov, V. P., "Survey of maneuvering target tracking. Part V: Multiple-model methods", *IEEE Trans. Aerospace and Electronic Systems*, vol. 41, no. 4, pp. 1255–1321, Oct. 2005.

[58] MacCormick, J. and Blake, A. "Probabilistic exclusion principle for tracking multiple objects," in *Proc. IEEE International Conference on Computer Vision,* vol 1, pp. 572–578, Kerkyra, Greece, Sep 1999.

[59] Mahalanabis, A. K., Zhou, B., and Bose, N. K., "Improved multi-target tracking in clutter by PDA smoothing," *IEEE Trans. Aerospace and Electronic Systems,* vol. 26, no. 1, pp. 113–121, Jan. 1990.

[61] Mahler, R., *An Introduction to Multisensor-Multitargat Statistics and its Application*, Lockheed Martin Technical Monograph, 2000.

[61] Mahler, R., "Multitarget moments and their applications to multitarget tracking," in *Proc. Workshop on Estimation, Tracking and Fusion,* pp. 134–166, Monterey, CA, May 2001.

[62] Mahler, R., "Random set theory for target tracking and identification," in D. Hall and J. Llinas Eds., *Data Fusion Handbook,* Boca Raton, FL: CRC Press, 2001, ch. 14.

[63] Mahler, R., "Multitarget Bayes filtering via first-order multitarget moments," *IEEE Trans. Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1152–1178, Oct. 2003.

[64] Mahler, R., "Objective functions for Bayesian control-theoretic sensor management I: Multitarget first-moment approximation," in *Proc. IEEE Aerospace Conference,* vol. 4, pp. 1905–1923, Mar. 2003.

[65] Mahler, R., "'Statistics 101' for multitarget, multisensor data fusion," *IEEE Magazine on Aerospace and Electronic Systems,* vol. 19, no. 1, pp. 53–64, Jan. 2004.

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

[67] Pattipati, K., Popp, R., and Kirubarajan, T., "Survey of assignment techniques for multitarget tracking," in Y. Bar-Shalom and W. D. Blair Eds., *Multitarget-Multisensor Tracking: Applications and Advances,* vol. III, Dedham, MA: Artech House, 2000, ch. 2.

[68] Nandakumaran, N., Sinha, A., and Kirubarajan, T., "Joint detection and tracking of unresolved targets with a joint-bin processing monopulse radar," in *Proc. Signal Processing, Sensor Fusion, and Target Recognition,* Orlando, FL, Apr. 2006.

[69] Nandakumaran, N., Sinha, A., and Kirubarajan, T., "Hybrid radar signal fusion for unresolved target detection", in *Proc. Signal and Data Processing of Small Targets,* San Diego, CA, Aug. 2007.

[70] Nandakumaran, N., Punithakumar, K., and Kirubarajan, T., "Improved multitarget tracking using probability hypothesis density smoothing", in *Proc. Signal and Data Processing of Small Targets,* San Diego, CA, Aug. 2007.

[71] Nandakumaran, N., Tharmarasa, R., Lang, T., and Kirubarajan, T., "Gaussian mixture probability hypothesis density smoothing with multistatic sonar", in *Proc. Signal Processing, Sensor Fusion, and Target Recognition,* Orlando, FL, Mar. 2008.

[72] Nandakumaran, N., Sinha, A., and Kirubarajan, T., "Joint detection and tracking of unresolved targets with Monopulse radar", *IEEE Trans. on Aerospace and Electronic Systems*, vol. 22, no. 4, pp. 1326–1341, Oct. 2008.

[73] Nandakumaran, N., Kirubarajan, T., Lang, T., McDonald, M., and Punithakumar, K., "Multitarget tracking using probability hypothesis density smoothing," Under third review for *IEEE Trans. on Aerospace and Electronic Systems,* Apr. 2008.

[74] Nandakumaran, N, Kirubarajan, T., Lang, T., and McDonald, M., "Gaussian mixture probability hypothesis density smoothing with multiple sensors", Submitted to *IEEE Trans. on Aerospace and Electronic Systems*, Jun. 2009.

[75] Nandakumaran, N. and Kirubarajan, T., "Maneuvering target tracking using probability hypothesis density smoothing", in *Proc. Signal Processing, Sensor Fusion, and Target Recognition,* Orlando, FL, Apr. 2009.

[76] Punithakumar, K., Kirubarajan, T., and Sinha, A., "A multiple-model probability hypothesis density filter for tracking maneuvering targets", in *Proc. Signal and Data Processing of Small Targets*, vol. 5428, O. E. Drummond, Ed., Apr. 2004.

[77] Punithakumar, K., Kirubarajan, T., and Sinha, A., "Multiple-model probability hypothesis density filter for tracking maneuvering targets," *IEEE Trans. on Aerospace and Electronic Systems,* vol. 44, no. 1, pp. 87–98, Jan. 2008.

[78] Ramachandra, K. V., *Kalman Filtering Techniques for Radar Tracking.* New York, 2000.

[79] Rauch, H. E., Tung, F., and Striebel, C. T., "Maximum likelihood estimates of linear dynamic systems", *AIAA Journal*, vol. 3, no. 8, pp. 1445–1450, 1965.

[80] Reid, D. B., "An algorithm for tracking multiple targets," *IEEE Trans. Automatic Control,* vol. AC-24, no. 6, pp. 843–854, Dec. 1979.

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

[82] Robert, C. P., *The Bayesian Choice: A Decision-Theoretic Motivation.* New York: Springer, 1994.

[83] Rubin, D. B., "Using the SIR algorithm to simulate posterior distributions," in M.H. Bernardo, K.M. an DeGroot, D.V. Lindley, and A.F.M. Smith, Eds., *Bayesian Statistics 3,* Oxford, UK: Oxford University Press, 1992.

[84] Runnalls, A. R., "Kullback-Leibler approach to Gaussian mixture reduction," *IEEE Trans. Aerospace and Electronic Systems,* vol. 43, no. 3, pp. 989-999, Jul. 2007.

[85] Sage, A. P. and Melsa, J. L. *Estimation Theory with Applications to Communications and Control.* New York: McGraw-Hill, 1971.

[86] Salmond, D. J., "Mixture reduction algorithms for target tracking," in *IEE Colloquium on State Estimation in Aerospace and Tracking Applications,* pp. 7/1-7/4, London, UK, Dec. 1989.

[87] Schrempf, O. C., Feiermann, O., and Hanebeck, U. D., "Optimal mixture reduction of the product of mixtures," in *Proc. the 8th International Conference on Information Fusion,* pp. 85-92 Philadelphia, PA, Jul. 2005.

[88] Sidenbladh, H., "Multi-target particle filtering for the probability hypothesis density," in *Proc. Int. Conf. Information Fusion,* pp. 800–806, 2003.

[89] Simon, D., *Optimal State Estimation.* New Jersey: Wiley, 2006.

[90] Smith, A. F. M. and Gelfand, A. E., "Bayesian statistics without tears: A sampling resampling perspective," *The American Statistician*, vol. 46, no. 2, pp. 84–88, 1992.

[91] Sorenson, H. W. and Alspach, D. L., "Recursive Bayesian estimation using Gaussian sum," *Automatica,* vol. 7, pp. 465-479, 1971.

[92] Stone, L. D., Barlow, C. A., and Corwin, T. L., *Bayesian Multiple Target Tracking.* Boston: Artech House, 1999.

[93] Streit, R. L. and Luginbuhl, T. E., "Maximum likelihood method for probabilistic multi-hypothesis tracking," *Proc. Signal and Data Processing of Small Targets,* vol. 2235, pp. 394–405, O. E. Drummond, Ed., Jul. 1994.

[94] Tobias, M. and Lanterman, A.D., "A probability hypothesis density based multitarget tracker using multiple bistatic range and velocity measurements," *Proc. 36th Southeastern Symposium on System Theory,* pp. 205–209, 2004.

[95] Vo, B.-N., Singh, S., and Doucet, A., "Sequential Monte Carlo implementation of the PHD filter for multi-target tracking," *Proc. Int. Conf. Information Fusion,* pp. 792–799, 2003.

[96] Vo, B.-N., Singh, S., and Doucet, A., "Sequential Monte Carlo methods for multitarget filtering with random finite sets," *IEEE Trans. Aerospace and Electronic Systems,* vol. 41, no. 4, pp. 1224–1245, Oct. 2005.

[97] Vo, B.-N. and Ma, W.-K., "A closed-form solution to the probability hypothesis density filter," in *Proc. Int'l Conf. on Information Fusion*, Philadelphia, 2005.

[98] Vo, B.-N. and Ma, W.-K., "The Gaussian mixture probability hypothesis density filter," *IEEE Trans. Signal Processing,* vol. 54, no. 11, pp. 4091–4104, Nov. 2006.

[99] Vo, B.-N., Pasha, A., and Tuan, H. D., "A Gaussian mixture PHD filter for nonlinear jump Markov models," in *Proc. Decision and Control,* pp. 3162–3167, Dec. 2006.

[100] Wan, E. A. and Van der Merwe, R., "The unscented Kalman filter for nonlinear estimation," in *Proc. Sympsium of Adaptive Systems for Signal Processing, Communications, and Control,* Lake Louise, AB, Canada, Oct. 2000.

[101] Wan, E. A. and van der Merwe, R., "The unscented Kalman filter," In Haykin, S., Ed., *Kalman Filtering and Neural Networks,* Ch. 7, New York: Wiley, 2001.