Nonlinear Filtering Algorithms for Multitarget Tracking

NONLINEAR FILTERING ALGORITHMS FOR MULTITARGET TRACKING

by

K. PUNITHAKUMAR, B.Sc. Engineering, M.A.Sc.

A Thesis

Submitted to the School of Graduate Studies

in Partial Fulfilment of the Requirements

for the Degree

Doctor of Philosophy

McMaster University

© Copyright by K. Punithakumar, December 2007

DOCTOR OF PHILOSOPHY (2008) MCMASTER UNIVERSITY

(Electrical and Computer Engineering) Hamilton, Ontario, Canada

TITLE: **Nonlinear Filtering Algorithms for Multitarget Tracking**

AUTHOR: K. Punithakumar

B.Sc. Engineering

University of Moratuwa

Moratuwa, Sri Lanka, 2001

M.A.Sc.

McMaster University

Hamilton, Canada, 2003

SUPERVISOR: Dr. T. Kirubarajan

NUMBER OF PAGES: xiii, 153

To my family

Abstract

Tracking multiple targets with uncertain target dynamics is a difficult problem, especially with nonlinear state and/or measurement equations. Random finite set theory provides a rigorous foundation to multitarget tracking problems. It provides a framework to represent the full multitarget posterior in contrast to other conventional approaches. However, the computational complexity of performing multitarget recursion grows exponentially with the number of targets. The Probability Hypothesis Density (PHD) filter, which only propagates the first moment of the multitarget posterior, requires much less computational complexity.

This thesis addresses some of the essential issues related to practical multitarget tracking problems such as tracking target maneuvers, stealthy targets, multitarget tracking in a distributed framework. With maneuvering targets, detecting and tracking the changes in the target motion model also becomes important and an effective solution for this problem using multiple-model based PHD filter is proposed. The proposed filter has the advantage over the other methods in that it can track a time-varying number of targets in nonlinear/non-Gaussian systems.

Recent developments in stealthy military aircraft and cruise missiles have emphasized the need to track low SNR targets. The conventional approach of thresholding

the measurements throws away potential information and thus results in poor performance in tracking dim targets. The problem becomes even more complicated when multiple dim targets are present in the surveillance region. A PHD filter based recursive track-before-detect approach is proposed in this thesis to track multiple dim targets in a computationally efficient way.

This thesis also investigates multiple target tracking using a network of sensors. Generally, sensor networks have limited energy, communication capability and computational power. The crucial consideration is what information needs to be transmitted over the network in order to perform online estimation of the current state of the monitored system, whilst attempting to minimize communication overhead.

Finally, a novel continuous approximation approach for nonlinear/non-Gaussian Bayesian tracking system based on spline interpolation is presented. The resulting filter has the advantages over the widely-known discrete particle based approximation approach in that it does not suffer from degeneracy problems and retains accurate density over the state space. The filter is general enough to be applicable to nonlinear/non-Gaussian system and the density could even be multi-modal.

Acknowledgements

Praise be to GOD for helping me in finishing this thesis and in all my life aspects. I wish to express my deepest gratitude to my supervisor Dr. T. Kirubarajan for offering me an opportunity to be a graduate student, and his guidance and inspiration through 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 Dr. Tim Davidson, Dr. Tim Field and Dr. A. Sinha for their guidance and kind suggestions through out my work.

Department staff members', especially Cheryl Gies, Helen Jachna and Hamida Najak, constant administrative help and support are also gratefully appreciated. I thank Terry Greenlay and Cosmin Coriou assisting in resolving the computer related issues.

I am thankful to my friends Kamalanesan, Nanthan, Tharman and others in the ETF Laboratory for many thought-provoking and stimulating discussions. My special heartiest thanks to Mahes and his family for all of their love and support.

All above, I would like to acknowledge the immeasurable contribution of my parents in teaching me the values of education. Finally and most importantly, I wish to thank Periyappa, Sitha, Theva and their family members. Their continuous encouragement has been extraordinary and wonderful bundle of joy.

Contents

1 Introduction **1**

1.1 Target Tracking 1

1.2 Multitarget Tracking 2

1.2.1 Probability Hypothesis Density Filter 3

1.3 Original Contributions 6

1.4 Related Publications 14

1.4.1 Journal Articles 14

1.4.2 Conference Proceedings 14

2 Bayesian Filtering **16**

2.1 Single Target Bayesian Filtering 16

2.2 Multitarget Bayesian Filtering 17

3 The Probability Hypothesis Density Filter **20**

3.1 Introduction 20

3.1.1 Prediction 22

3.1.2 Update 22

3.2 PHD Filter Derivation 23

3.2.1 Point Process Theory 23
3.2.2 Janossy Measures 24
3.2.3 Probability Generating Functionals 25
3.2.4 The Prediction Equation 28
3.2.5 The Update Equation 30
3.3 A Physical-Space Approach for the PHD 34
3.3.1 Prediction 34
3.3.2 Update 35

4 PHD Filter Implementations **39**
4.1 Sequential Monte Carlo PHD filter 39
4.1.1 Prediction 40
4.1.2 Update 41
4.1.3 Resample 42
4.2 Gaussian Mixture PHD Filter 42
4.2.1 Gaussian Mixture PHD Recursion 44

5 A Multiple Model PHD Filter **47**
5.1 Introduction 47
5.2 Problem Formulation 48
5.3 The Multiple Model Approach to Target Tracking 49
5.3.1 The Mixing Stage 50
5.3.2 The Prediction Stage 51
5.3.3 The Update Stage 52
5.4 Particle Implementation 53
5.4.1 Prediction 54

5.4.2 Update . . . 57
5.4.3 Resample . . . 57
5.5 Simulations . . . 58
5.5.1 Range and Bearing Sensor . . . 58
5.5.2 Bearings-only Sensors . . . 61

6 An SMC PHD Algorithm for Multitarget TBD **83**
6.1 Introduction . . . 83
6.2 System Setup . . . 84
6.2.1 Target Model . . . 84
6.2.2 Sensor Model . . . 85
6.3 PHD Filter Implementation . . . 87
6.3.1 Prediction . . . 88
6.3.2 Update . . . 89
6.3.3 Resample . . . 90
6.4 Simulations . . . 91

7 Distributed Implementation of SMC PHD Filter **100**
7.1 Introduction . . . 100
7.2 Distributed PHD Filter . . . 102
7.2.1 Quantization and Encoding . . . 104
7.2.2 Vectorization . . . 106
7.2.3 Algorithm . . . 106
7.3 Simulations . . . 108

8 Spline Filter for Bayesian Tracking **117**

8.1 Introduction 117

8.2 Nonlinear Bayesian Filering 118

8.3 B-splines 119

8.3.1 Positivity Property of B-splines 120

8.3.2 Knot Insertion Property of B-splines 121

8.3.3 Stable Evaluation of B-splines 121

8.3.4 Integral of B-splines 123

8.3.5 Construction of B-splines 123

8.4 Spline Filter 124

8.4.1 Prediction 124

8.4.2 Update 125

8.4.3 Find estimate 126

8.5 Simulations 126

9 Conclusions and Future Work **135**

9.1 Conclusions 135

9.2 Future Work 138

List of Tables

5.1 Average target state estimation RMSE (based on 100 Monte Carlo runs) 70

5.2 Average target state estimation RMSE (based on 100 Monte Carlo runs) 75

8.1 Comparison of algorithms (based on 100 Monte Carlo runs) 132

List of Figures

5.1 Target truth and multiple model PHD estimates 61

5.2 Target truth and constant velocity PHD estimates 62

5.3 Target truth and coordinated turn PHD estimates 63

5.4 Model switching in the MMPHD filter – Target 1 64

5.5 Model switching in the MMPHD filter – Target 2 65

5.6 Position RMSE of target 1 estimates 66

5.7 Position RMSE of target 2 estimates 67

5.8 Velocity RMSE of target 1 estimates 68

5.9 Velocity RMSE of target 2 estimates 69

5.10 Target trajectory (I - initial, F - Final, S_i - Sensor) 74

5.11 Model switching in target 1 in the MMPHD filter 75

5.12 Model switching in target 2 in the MMPHD filter 76

5.13 Target 1 – Position estimation RMSE 77

5.14 Target 2 – Position estimation RMSE 78

5.15 Target 1 – Velocity estimation RMSE 79

5.16 Target 2 – Velocity estimation RMSE 80

5.17 Target 1 – Turn rate estimation RMSE 81

5.18 Target 2 – Turn rate estimation RMSE 82

6.1 Simulated sensor data – Frame 1 94
6.2 Simulated sensor data – Frame 7 94
6.3 Simulated sensor data – Frame 15 95
6.4 Simulated sensor data – Frame 50 95
6.5 Number of targets estimate (single run) 96
6.6 Target position particles at 19 s 97
6.7 Target position particles at 39 s 98
6.8 Target position particles at 75 s 99
7.1 Scenario 111
7.2 Target position estimation RMSE - Target 1 112
7.3 Target position estimation RMSE - Target 2 113
7.4 Target position estimation RMSE - Target 3 114
7.5 Compression ratio 115
7.6 Number of bits transmitted 116
8.1 True values of x_k as a function of k 128
8.2 Measurements of z_k of the states x_k shown in Figure 8.1 129
8.3 Spline filter estimate of posterior mean 130
8.4 SIR particle filter estimate of posterior mean 131
8.5 One cycle of recursive estimation (at $k = 10$) 133
8.6 One cycle of recursive estimation (at $k = 10$) 134

Chapter 1

Introduction

1.1 Target Tracking

Target tracking [13] is the process of estimating the states of objects of interest or targets based on sensor observations. It is an essential part of systems for surveillance or navigation. In a tracking system, sensors such as radar, sonar and imaging sensors report observations to the system, while the tracking algorithm processes these measurements and provides an estimate of the target state such as position and velocity. The observations are usually taken at regular intervals and the target states that evolve in time are estimated. In general, the tracking algorithm uses one or more mathematical models to describe the target state evolution.

Recursive Bayes filtering is the general probabilistic approach for estimating the target state over time using the measurements. In recursive Bayes filtering, the true state is assumed to follow an unobserved Markov process, i.e., the target state at a given time is conditionally independent of all the earlier states given the immediate previous state. Similarly, the measurement at a particular time instant is dependent

only upon the state of that time and conditionally independent of all the other states given the state at the time of measurement. With these assumptions, the posterior target probability is recursively propagated by predicting the target state distribution using state model and updated when measurements become available.

Kalman filter [57] is the most widely used recursive Bayesian filter when the system is characterized as linear dynamic system perturbed by additive Gaussian noise. Kalman filter propagates only the first and second moment, i.e., the mean and the covariance, of the single target state posterior. However, Kalman filter becomes ineffective when the system is nonlinear and perturbed by non-Gaussian noise. Sequential Monte Carlo (SMC) approaches [4,46] have been proved to be effective in such situations. They are alternative to other nonlinear filtering algorithms with the advantage that with enough samples they approach optimal Bayes estimation.

1.2 Multitarget Tracking

Tracking a single target is much less complicated than multitarget tracking since the state dimension does not evolve in time. The multitarget tracking problem [14, 16] generalizes the tracking problem into a situation where the number of targets may not be known and varies with time. In addition, the measurements might have originated from targets or clutter and some of them may be due to false alarm. Some targets may not produce measurements at a particular instant due to missed detections. In such scenarios, the state and measurement are two collections of individual targets and measurements. Since the dimensions of the multitarget state and measurement also evolve in time the problem becomes more complicated. The conventional method to solve multitarget problems is to assign a single-target filter such as Kalman filter to

each target and use a data association technique to assign the measurement associated with that target.

A large number of strategies are available in the literature to solve data association problem. The widely used classical approaches are Nearest Neighbour Standard Filter (NNSF) [12], Joint Probability Data Association Filter (JPDAF) [12,44] and Multiple Hypotheses Tracker (MHT) [84]. The NNSF predicts measurement for each target state and associates the target with the closest measurement. However, this method throws away many feasible hypotheses. In contrast, MHT keeps track of all the possible association hypotheses over time. This is an NP-hard problem, since the number of associations grows exponentially over time. Thus the method requires relaxation to reduce the computational complexity. JPDAF is a more appealing approach in that at each time step infeasible hypotheses are pruned away. A filtering estimate is then calculated for each of the remaining hypotheses and combined in proportion to the corresponding posterior hypotheses probabilities. However, to maintain tractability, the final estimate in this method is collapsed to a single Gaussian, thus throwing away pertinent information. Due to its combinatorial nature, data association takes most of the computational load in multitarget tracking problems. Alternative methods that avoid explicit association between measurements and targets include Symmetric Measurement Equations (SME) [58] and Random Finite Sets (RFS) [45,47,69].

1.2.1 Probability Hypothesis Density Filter

Multitarget filtering is a class of state estimation problem in which the entity of interest is a finite set. However, the set is random in the number of elements as well as the values of individual elements. The modelling of multitarget tracking problems

using random finite set leads to a theoretical foundation for optimal Bayes multitarget filter. The algorithm itself incorporates track initiation and termination while in many other conventional algorithms they have to be performed separately.

The application of RFS to multitarget tracking problems started recently in the literature. The earliest published work using point process theoretical formalism of multitarget tracking can be found in [77]. A point process based filter to track unknown but fixed number of targets was proposed in [100]. Finite Set Statistics (FISST) [45, 47, 65–67] provides a systematic approach for multisensor-multitarget tracking problems by prividing a unified framework for data fusion using random set theory and point process theory. FISST approach extends the classical single target single sensor Bayes theory to multisensor multitarget statistics by providing a conceptual Bayes view of the targets as a single meta-target and observations set as a single meta-observation.

The multitarget Bayes filter propagates the multitarget posterior recursively in time. However, this approach is not tractable due to very high demand on computation power and the optimal Bayes filter needs to be approximated for real-time applications. A tractable alternative approach is the Probability Hypothesis Density (PHD) filter [65, 69, 71], which propagates only the first statistical moment of the full multitarget posterior. For example, [42] describes a spectral compression-based PHD filter.

The PHD filter assumes that the predicted multitarget density is Poisson. The recursive PHD filter completely characterizes the statistics of the dynamic Poisson point process under the assumption that the predicted multitarget density is Poisson distributed. Discussion about dynamic Poisson point processes in parametric estimation context can be found in [36, 59, 95]. The mathematical derivations of the PHD filter

equations based on probability generating functionals and functional derivatives are given in [70]. The PHD is a positive function on all over the domain and integrable. However, it should be noted that the PHD is different from ordinary probability density. In contrast to probability density, it does not necessarily integrate to unity. The integral of the PHD in any region of the state space is the expected number of targets in that region. The advantage of the PHD filter is that it can track a time varying number of targets, estimating both the number of targets as well as the individual target state values. Thus, it avoids the need for data association since the identities of individual targets are not required. In [83], it has been shown that the PHD filter outperforms the MHT filter for linear Gaussian target model, based on a comparison using the Wasserstein distance [51].

Even though the PHD filter requires much less computational power than the multitarget Bayes filter, it still involves a multiple integral that cannot be solve analytically even for a simple target tracking problem. The PHD filter has been shown to be a computationally tractable method for unified group target detection, tracking and classification [64] and for clutter tracking [68]. The SMC [6] and Gaussian Mixture (GM) [9, 10] are proposed in the literature to implement the PHD filter.

The PHD filter has gained much attention by many researchers in recent years. A method to obtain track-valued estimates of individual targets for SMC PHD filter and a track labelling approach have been presented in [82] and [62], respectively. A PHD smoothing algorithm to improve the capability of the PHD based tracking system for delayed estimates is proposed in [80].

1.3 Original Contributions

This thesis addresses some essential issues related to the problem of tracking time-varying multiple number of targets in nonlinear/non-Gaussian systems. The considered application areas are tracking maneuvering targets, stealthy targets and target tracking in a distributed sensor network. We show the application of PHD filter is effective in solving in the above problems. Further, this thesis also addresses the implementation of nonlinear filtering using a novel continuous approach using splines.

Chapter 5: A Multiple Model PHD Filter

This chapter presents the theoretical development and the implementation details of the multiple model PHD filter to track multiple maneuvering targets. Tracking a maneuvering target using multiple models has been shown to be highly effective in many applications. In single-model filters, if the model used by the filter does not match the actual system dynamics, the filter will tend to diverge since the actual errors fall outside the range predicted by the filter using its error covariance [13]. Maneuvering targets might switch between different modes of operation, and tracking using a single-model filter might fail since the filter may be accurate for only one mode of operation. In multiple-model approaches, several filters, each matched to a different target motion mode, operate in parallel, and then the overall state estimate is given by a weighted sum of the estimates from each filter. In many target tracking problems with linear, Gaussian systems, the Interacting Multiple Model (IMM) estimator [13, 24, 25], in which a bank of different hypothetical target motion models is used, has been proven to have better performance than the (single-model) Kalman filter. For example, in the target tracking benchmark problem [18], in which

performances of different algorithms for tracking highly maneuvering targets were compared, the IMM estimator outperformed the Kalman and α–β filters and yielded one of the best solutions. Extensions of multiple models to particle filters [28,76] have also resulted in better results for nonlinear, non-Gaussian systems than single-model particle filters.

Tracking multiple maneuvering targets is much more difficult than tracking a single maneuvering target since there is a challenge in correctly associating the measurements with the targets. In the literature, extensions of the IMM estimator to work with standard tracking algorithms have resulted in the IMMJPDA (IMM estimator with the Joint Probabilistic Data Association algorithm) [11,26,32], IMM/MHT (IMM estimator with the Multiple Hypothesis Tracking algorithm) [37,61] and the Joint IMMPDA particle filter [27]. A drawback of these algorithms is that all of them involve the model-data association problem and the problem is further complicated in a cluttered environment.

This thesis proposes a multiple-model PHD (MMPHD) filter to address the problem of tracking multiple maneuvering targets using the SMC approach. SMC methods [4,39,40,46] are very powerful tools for Bayes filter implementation and have been shown to outperformed many other conventional nonlinear filters. They have the advantages of computational tractability [78] and provable convergence properties [7,55], and are applicable under the most general circumstances since no assumptions need to be made on the form of the underlying probability density. The SMC approaches have even been applied as tractable solutions to multitarget tracking problems in literature [5, 41, 53, 74] in non-FISST contexts. However, the SMC approaches to multitaget problems are still expensive when the number of targets is large.

Since the PHD is not a probability density and the PHD recursion is not a standard

Bayesian, the direct application of the SMC approach to the PHD filter will fail. The modified version of the standard SMC approach to PHD filter is presented in [7]. An SMC implementation of the PHD filter [6] has been shown to be efficient in a cluttered environment, and a similar implementation [94] handled missed detections well. Particle system implementations of the PHD filter can also be found in [72, 101, 102].

The SMC approximation of the MMPHD filter is applicable to track multiple maneuvering targets with nonlinear, non-Gaussian dynamics. The implementation of the proposed filter is detailed, and applications such as tracking multiple targets using a single range and bearing sensor as well as multiple bearings-only sensors are considered. The results show the effectiveness of the proposed filter over single-model PHD filters.

Chapter 6: An SMC PHD algorithm for multitarget Track-Before-Detect

Recent developments of stealthy military aircrafts and cruise missiles emphasize the essence of tracking low Signal-to-Noise Ratio (SNR) targets. Detecting and tracking low SNR targets is a very difficult problem in heavy cluttered environments. The conventional approach to target tracking is based on the filtered measurements that are extracted from thresholding the output signal of the processing unit of the sensor [17]. The thresholding of the sensor data reduce the data to be processed and thus simplify the tracking process. However, it throws away potentially useful information and results in poor performance in tracking dim targets. Track-Before-Detect (TBD) techniques, which allow simultaneous detection and tracking using unthresholded data,

show better performance over classical target tracking methods. The conventional TBD algorithms such as Hough transform [30], dynamic programming [3, 15], maximum likelihood estimation [98] require batch processing of several scans of measurement data. However, these methods require enormous computational power since they operate on multiple scans of measurement data in batch manner. The development of a recursive Bayesian TBD algorithm reduces the computational burden over conventional TBD methods. Particle filter based implementations of recursive TBD are given in [86, 90]. The particle filter based recursive TBD algorithms [20, 89, 90] offer flexibility in the target and noise models for single target tracking. Moreover, the effects of unknown and fluctuating target intensity, point spread functions and extended objects can be accommodated. A detailed explanation of an implementation of the filter is given in [85] with derivations of Cramér-Rao bounds on system performance. An efficient particle filter implementation for a more realistic simulation scenario with Rayleigh noise to reflect signal processing of an operational sensor such as radar could be found in [88].

Extension of particle based TBD algorithm for multitarget tracking is given in [21]. In this approach a modelling setup is used to accommodate the varying number of targets where a multiple model SMC based TBD approach is used to solve the problem conditioned on the model, i.e., number of targets. The algorithm can deal with a limited number of targets and it was assumed that the maximum possible number of targets is known.

We propose the PHD filter [69, 70] for recursive TBD algorithm and an SMC approach to implement the algorithm [6]. The resulting algorithm does not require a modelling setup to accommodate varying number of targets and will be computationally efficient when the number of the targets is high. The algorithm requires suitably

chosen models for target birth, spawning and disappearance. The number of target estimate in the surveillance region is given by integral of the PHD or the total weight of the samples. The proposed filter has an advantage over the other methods in that it does not have a restriction on the maximum number of possible targets. A simulation based on similar target and sensor models as in [85] for a multitarget setup is presented with results to show the effectiveness of the proposed filter.

Chapter 7: Distributed implementation of SMC PHD Filter

This chapter considers the problem of tracking multiple targets from multiple sensing platforms in a decentralized network. In wireless sensor networks, a large number of inexpensive and small sensors are distributed over the monitored region. Sensors have wireless connectivity and report observations to nodes. These nodes have computation and communication capabilities. Limited network communication is one of the challenges that must be overcome to implement practical sensor networks and one of the primary considerations is to maintain target estimation accuracy whilst minimizing network communications.

In this chapter, we consider the situation where the monitored system contains multiple targets of interest and the nature of the monitored system involves nonlinear target dynamics, nonlinear observations, and non-Gaussian process and observation noises. Further, the number of targets may change over time as the targets appear or disappear in the scene. Our goal is to perform sequential estimation of the current system based on the observations received at multiple sensors of the network. An observation might either be target-originated or false alarm. The decentralized Kalman

filter [79], which admits an attractive, relatively low-dimensional parametric information exchange between nodes, becomes inapplicable for nonlinear and non-Gaussian scenarios. The typical algorithms that can be used in nonlinear and non-Gaussian systems are the extended Kalman filter [23], and Gaussian sum approximations. However, the SMC methods, such as particle filters, perform better than these tracking algorithms at the expense of higher computational burden.

A particle filter based distributed algorithm for nonlinear and non-Gaussian scenarios is presented in [34] with a single target tracking example. It presents two approaches to run the particle filters in a distributed fashion. The first approach uses a parametric approximation to the likelihood factors that are formed using the particles of the filter. Then, the model parameters are exchanged between nodes instead of particles. The second approach uses an adaptive data-encoding scheme. In this approach, the measurements are quantized and encoded before they are transmitted over the network. An improvement to this approach using a better encoding scheme and measurement vectorization has been presented in [54]. More particle filter based distributed implementations are given in [87, 92].

This thesis introduces a distributed implementation of a SMC PHD filter to track multiple targets. The proposed algorithm uses an adaptive data-encoding approach, which involves training of predictive uniform quantizers based on identical SMC PHD filters maintained at each node. The compression is substantial because the filter provides a very good indication where an observation is likely to be. However, the encoding scheme is computationally expensive and the amount of data that can be processed is restricted by the computational power of the nodes. Further, each node waits for several time steps and transmits the vectorized, encoded observation data to all the other nodes in the network. The vectorization of measurements also provides

a substantial communication savings by reducing the number of bits transmitted on overhead bits, such as packet headers.

Chapter 8: Spline Filter for Bayesian Tracking

The implementation problem for nonlinear estimator of the state in a discrete time dynamic system employs a recursive filter arising from Bayes rule. A good representation of the probability densities and likelihood functions is a essential issue in the success of the filtering algorithm, especially in the nonlinear case. A variety of numerical methods have been developed to address the problem which include extended Kalman filter (EKF) [13], unscented Kalman filter (UKF) [56] and various SMC filters [4,46]. The SMC approaches have the advantage over EKF or UKF in that they are applicable under most general circumstances since no assumptions need to be made on the form of the underlying probability density. They are based on point mass representation of probability densities which can be applied to any state-space model. In this chapter, we present a spline based approach which gives a continuous representation of the probability densities over the target state space.

In the literature, methods for constructing splines to represent a probability density are widely analyzed in various application areas [38, 103]. Splines were also investigated in the literature to solve nonlinear estimation problems. An approach using monosplines to address the problem of nonlinear estimation using quadratures is presented in [99]. This approach assumes that the process and measurement noises are zero mean Gaussian. In [29], another method is presented to obtain the nonlinear estimate of phase demodulation with two dimensional phase process model as an alternative to the Fourier filter.

The proposed spline based filtering algorithm can also be applied to the most general circumstances and the target density could even be multi-modal. Further, splines can be used to provide more natural representation of the probability density by maintaining positivity and smoothness. Spline based approach allows easy manipulation of the density functions since they are polynomials of finite order. For example, finding the target probability in any region in the state space is straightforward with spline representation since it involves evaluation of the integral of polynomial over that region.

This chapter presents details of the prediction and update steps of the spline filter for nonlinear/non-Gaussian Bayesian filtering based on B-spline interpolation for one dimensional case. This work can be extended to multi-dimensional case using tensor product approaches. The whole region of the state space within the surveillance region could be represented by the spline interpolation. However, the proposed algorithm interpolates the probability density function only over the region in which the target density is significant to reduce the computational power needed. The target density outside the interval is assumed to be zero. This interval changes over time in the recursion and it is similar to the region that consists of all the particles in a particle filter. A point mass approach is used in the prediction step to construct the interpolation of the predicted target density. The updated density is calculated by multiplying the interpolations of the predicted target density and likelihood function and then it is normalized.

1.4 Related Publications

1.4.1 Journal Articles

1. K. Punithakumar and T. Kirubarajan, "Spline filter for nonlinear/non-Gaussian Bayesian tracking", *To be submitted to IEEE Transactions on Aerospace and Electronic Systems.*

2. K. Punithakumar, T. Kirubarajan and A. Sinha, "A distributed implementation of a sequential Monte Carlo probability hypothesis density filter for sensor networks", *Submitted to IEEE Transactions on Aerospace and Electronic Systems, 2007*

3. K. Punithakumar, T. Kirubarajan and A. Sinha, "A sequential Monte Carlo probability hypothesis density algorithm for multitarget track-before-detect", *Submitted to IEEE Transactions on Aerospace and Electronic Systems, 2007*

4. K. Punithakumar, A. Sinha and T. Kirubarajan, "A multiple-model probability hypothesis density filter for tracking maneuvering targets", *IEEE Transactions on Aerospace and Electronic Systems, in press.*

1.4.2 Conference Proceedings

1. K. Punithakumar and T. Kirubarjan, "Spline filter for nonlinear/non-Gaussian Bayesian tracking", *Proceedings of SPIE's Signal and Data Processing of Small Targets*, Vol. 6699, San Diego, CA, September 2007.

2. K. Punithakumar, T. Kirubarajan and A. Sinha, "A distributed implementation of a sequential Monte Carlo probability hypothesis density filter for sensor

networks", *Proceedings of SPIE's Signal Processing, Sensor Fusion, and Target Recognition XV*, Vol. 6235, 62350L May 2006.

3. K. Punithakumar, T. Kirubarajan and A. Sinha, "A sequential Monte Carlo probability hypothesis density algorithm for multitarget track-before-detect", *Proceedings of SPIE's Signal and Data Processing of Small Targets*, Vol. 5913, 59131S, September 2005.

4. K. Punithakumar, T. Kirubarajan and A. Sinha, "A multiple-model probability hypothesis density filter for tracking maneuvering targets", *Proceedings of SPIE's Signal and Data Processing of Small Targets 2004*, Vol. 5428, pp. 113–121, August 2004.

Chapter 2

Bayesian Filtering

2.1 Single Target Bayesian Filtering

Single-target tracking is the simpler version of the target tracking problem in which the tracking filter estimates the state of a single target at each point in time based on a discrete set of measurements. Let us take the state of the target at time k is denoted by $\mathbf{x_k}$. In Bayes filtering we assume that the true target state is conditionally independent of all earlier states given the immediate previous state. That is,

$$p(\mathbf{x}_k|\mathbf{x}_0, \ldots, \mathbf{x}_{k-1}) = p(\mathbf{x}_k|\mathbf{x}_{k-1}) \tag{2.1}$$

Similarly, if we assume that the measurement at time k is dependent only upon the state $\mathbf{x}_k$ and is conditionally independent of all the other states given the current state, then

$$p(\mathbf{z}_k|\mathbf{x}_0, \ldots, \mathbf{x}_k) = p(\mathbf{z}_k|\mathbf{x}_k) \tag{2.2}$$

Under these assumptions, using Bayes rule, the probability distribution over all

states can be written as

$$p(\mathbf{x}_0, \ldots, \mathbf{x}_k, \mathbf{z}_0, \ldots, \mathbf{z}_k) = p(\mathbf{x}_0) \prod_{i=1}^{k} p(\mathbf{z}_i|\mathbf{x}_i) p(\mathbf{x}_i|\mathbf{x}_{i-1}) \tag{2.3}$$

However, in filtering we are interested in estimating the probability distribution of the current state $\mathbf{x}_k$ conditioned on the measurements up to the current time step. This leads to two-step recursive filtering: namely the prediction and update. The probability distribution associated with the predicted state is given by

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

where $\mathbf{z}_{1:i} = \{\mathbf{z}_1, \ldots, \mathbf{z}_i\}$ denotes the set of measurements up to time i.

The probability distribution associated with the updated state is given by

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

The denominator

$$p(\mathbf{z}_k|\mathbf{z}_{1:k-1}) = \int p(\mathbf{z}_k|\mathbf{x}_k) p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) d\mathbf{x}_k \tag{2.6}$$

is a normalization term.

2.2 Multitarget Bayesian Filtering

We can generalize the single target Bayesian filtering to multitarget system by defining multitarget state and multitarget measurement. The approach is to use the

random finite set (RFS) to represent the Bayesian model for recursively estimating and updating the multitarget density based on measurements received at each time step. Let us assume that there are $M(k)$ targets $\mathbf{x}_{k,1}, \ldots, \mathbf{x}_{k,M(k)}$ in the state space E_s at time k. Then we can represent the multitarget state by a finite set $\mathbf{X}_k = \mathbf{x}_{k,1}, \ldots, \mathbf{x}_{k,M(k)} \subseteq E_s$. Similarly, we represent the $N(k)$ number of observations in the observation space E_o by $\mathbf{Z}_k = \mathbf{z}_{k,1}, \ldots, \mathbf{z}_{k,N(k)}$. Some of these measurements may be due to clutter. The dimensions of the state and measurement vectors $\mathbf{X}_k$ and $\mathbf{Z}_k$ are unknown and time varying . These randomness in the sets can be characterized by modelling the multitarget state and multitarget measurement as random finite sets (RFS) Ξ_k and Σ_k, respectively.

In a multitarget system, the number of targets may vary due to the following scenarios.

- There may be targets disappearing from the scene.
- There may be new targets spawning from the existing targets.
- There may be new target appearing spontaneously.

Given realization $\mathbf{X}_{k-1}$ of Ξ_{k-1} at time $k-1$, the multitarget state at time k can be modelled by the RFS

$$\Xi_k = S_k(\mathbf{X}_{k-1}) \bigcup B_k(\mathbf{X}_{k-1}) \bigcup \Gamma_k \tag{2.7}$$

where $S_k(\mathbf{X}_{k-1})$, $B_k(\mathbf{X}_{k-1})$ and Γ_k denote the RFS of survived targets, spawned targets and new born targets, respectively.

Similarly, given a realization of $\mathbf{X}_k$ of Ξ_k, the multitarget measurement can be

modelled by the RFS

$$\Sigma_k = \Theta_k(\mathbf{X}_k) \bigcup C_k(\mathbf{X}_k) \tag{2.8}$$

where $\Theta_k(\mathbf{X})_\mathbf{k}$ and $C_k(\mathbf{X}_k)$ denote the RFS of measurements generated by $\mathbf{X}_k$ and due to clutter, respectively.

Let $p_{k|k}(X_k|Z_{0:k})$ denote the multitarget posterior density. Then the prediction and update steps of the optimal multitarget Bayes filter recursion is given by

$$p_{k|k-1}(X_k|Z_{0:k-1}) = \int f_{k|k-1}(X_k|X) p_{k-1|k-1}(X|Z_{0:k-1}) \mu_s(dX) \tag{2.9}$$

$$p_{k|k}(X_k|Z_{0:k}) = \frac{g_k(Z_k|X_k) p_{k|k-1}(X_k|Z_{0:k-1})}{\int g_k(Z_k|X) p_{k|k-1}(X|Z_{0:k-1}) \mu_s(dX)} \tag{2.10}$$

where $f_{k|k-1}(X_k|X)$ is the dynamic model is governed by the transition density, $g_k(Z_k|X_k)$ is the multitarget likelihood and μ_s takes the place of Lebesque measure [7].

The computational complexity of the joint multitarget likelihood grows exponentially with the number of targets and thus the multitarget Bayes filter is not of practical interest [48]. The PHD filter, which uses only the first order statistical moment of the full target posterior, is derived to provide a sub-optimal strategy for the recursive multitarget tracking problems [69].

Chapter 3

The Probability Hypothesis Density Filter

3.1 Introduction

In tracking multiple targets, if the number of targets is unknown and varying with time, it is not possible to compare states with different dimensions using ordinary Bayesian statistics of fixed dimensional spaces. However, the problem can be addressed by using FISST to incorporate comparisons of state spaces of different dimensions. FISST facilitates the construction of multitarget densities from multiple-target transition functions by computing set derivatives of belief-mass functions [66], which makes it possible to combine states of different dimensions. The main practical difficulty with this approach is that the dimension of the full state space becomes large when many targets are present, which increases the computational load exponentially with the number of targets. 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 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 [94].

In general, a PHD-based multitarget tracker will experience more difficulty in resolving closely-spaced targets than the tracker based on the full target posterior. However, if the probability density functions of individual targets is highly concentrated around their means compared to the target separation, so that the individual target pdfs do not overlap significantly, it will become possible to resolve targets using the PHD filter as well. A theoretical explanation about the capability of the PHD filter to resolve closely-spaced targets in Gaussian context is given in [69].

By definition, the PHD $D_{k|k}(\mathbf{x}_k|Z_{1:k})$, with single target state vector $\mathbf{x}_k$, and given all the measurements up to time step k, is the density whose integral on any region S of the state space is the expected number of targets $N_{k|k}$ contained in S. That is,

$$N_{k|k} = \int_S D_{k|k}(\mathbf{x}_k|Z_{1:k})d\mathbf{x}_k \tag{3.1}$$

Since this property uniquely characterizes the PHD and the first-order statistical moment of the full target posterior distribution possesses this property, the first-order statistical moment of the full target posterior is indeed the PHD. The first moment of the full target posterior, or the PHD, given all the measurement $Z_{1:k}$ up to time step k, is given by the set integral [69]

$$D_{k|k}(\mathbf{x}_k|Z_{1:k}) = \int f_{k|k}(\{\mathbf{x}_k\} \cup Y|Z_{1:k})\delta Y \tag{3.2}$$

The reader is refered to [69] for detailed mathematical explanations about the PHD filter. The approximate expected target states are given by the local maxima of the PHD. The following section gives the prediction and update steps of one cycle of the PHD filter.

3.1.1 Prediction

In a general scenario of interest, there are target disappearances, target spawning, and entry of new targets. We denote the probability that a target with state $\mathbf{x}_{k-1}$ at time step $(k-1)$ will survive at time step k by $e_{k|k-1}(\mathbf{x}_{k-1})$, the PHD of spawned targets at time step k from a target with state $\mathbf{x}_{k-1}$ by $b_{k|k-1}(\mathbf{x}_k|\mathbf{x}_{k-1})$, and the PHD of new-born spontaneous targets at time step k by $\gamma_k(\mathbf{x}_k)$. Then, the predicted PHD is given by

$$\begin{aligned} D_{k|k-1}(\mathbf{x}_k|Z_{1:k-1}) &= \gamma_k(\mathbf{x}_k) \\ &\quad + \int \left[e_{k|k-1}(\mathbf{x}_{k-1}) f_{k|k-1}(\mathbf{x}_k|\mathbf{x}_{k-1}) + b_{k|k-1}(\mathbf{x}_k|\mathbf{x}_{k-1})\right] \\ &\quad \times D_{k-1|k-1}(\mathbf{x}_{k-1}|Z_{1:k-1}) d\mathbf{x}_{k-1} \end{aligned} \tag{3.3}$$

where $f_{k|k-1}(\mathbf{x}_k|\mathbf{x}_{k-1})$ denotes the single-target Markov transition density. The prediction equation (3.3) is lossless since there are no approximations.

3.1.2 Update

The predicted PHD can be corrected with the availability of measurements Z_k at time step k to get the updated PHD. We assume that the number of false alarms is Poisson-distributed with the average rate of λ_k and that the probability density of

the spatial distribution of false alarms is $c_k(\mathbf{z}_k)$. Let the detection probability of a target with state $\mathbf{x}_k$ at time step k be $p_D(\mathbf{x}_k)$. Then, the updated PHD at time step k is given by

$$D_{k|k}(\mathbf{x}_k|Z_{1:k}) \cong \left[\sum_{\mathbf{z}_k \in Z_k} \frac{p_D(\mathbf{x}_k) f_{k|k}(\mathbf{z}_k|\mathbf{x}_k)}{\lambda_k c_k(\mathbf{z}_k) + \psi_k(\mathbf{z}_k|Z_{1:k-1})} + (1 - p_D(\mathbf{x}_k))\right] D_{k|k-1}(\mathbf{x}_k|Z_{1:k-1}) \tag{3.4}$$

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

$$\psi_k(\mathbf{z}_k|Z_{1:k-1}) = \int p_D(\mathbf{x}_k) f_{k|k}(\mathbf{z}_k|\mathbf{x}_k) D_{k|k-1}(\mathbf{x}_k|Z_{1:k-1}) d\mathbf{x}_k \tag{3.5}$$

and $f_{k|k}(\mathbf{z}_k|\mathbf{x}_k)$ denotes the single-sensor/single-target likelihood. The update equation (3.4) is not lossless since approximations are made on predicted multitarget posterior to obtain the closed-form solution (3.4). The reader is referred to [69] for further explanations.

3.2 PHD Filter Derivation

This section provides the derivation of PHD filter based on point process theory and standard probability theory based on [8]. A Poisson assumption is made in the PHD filter in order to derive a closed form solution. The following subsections provides some mathematical background that used in the derivation of the PHD filter.

3.2.1 Point Process Theory

In multitarget tracking problems, the number of targets as well as their target state values need to be estimated. The set of target states can be modelled by a Point

Process, where the state of population is defined as an unordered set of points $X = x_1, \ldots, x_N$. The points are located in a complete separable state space χ where the number of targets as well as the target state values are random.

The following assumptions are made in order to formulate the multiple target tracking model as a point process. A distribution p_n, $n \in N$ is given which determines the total number of targets and satisfies $\sum_{n \in N} p_n = 1$. For each $n \geq 1$, a probability distribution d_n is given which determines the joint distribution of the n targets. Since the realization of the point process in target tracking involved are unordered sets, d_n need to be symmetric. If the distributions d_n are not symmetric, without loss of generality, they can be symmetrized [36].

$$d_n^{sym}(A_1 \times \ldots \times A_n) = \sum_{perm} d_n(A_{i1} \times \ldots \times A_{in}) \tag{3.6}$$

where $(A_1, \ldots, A_n)$ is any partition of the state space, $\sum_{perm}$ is taken over all partitions $(i_1, \ldots, i_n)$ of integers $(1, \ldots, n)$.

3.2.2 Janossy Measures

The process of estimating an unknown number of objects with no particular meaningful ordering is mathematically best characterized as a simple finite point process or, equivalently, a random finite set, and a priori and a posteriori statistics can be described as a set of Janossy measures. Janossy measures are symmetric by definition.

$$J_n(A_1 \times \ldots \times A_n) = n! p_n d_n^{sym}(A_1 \times \ldots \times A_n) \tag{3.7}$$

Let $j_n(x_1, \ldots, x_n)$ denote the density of J_n, then

$$\sum_{n=0}^{\infty} \frac{1}{n!} \int j_n(x_1, \ldots, x_n) dx_1 \ldots dx_n = 1 \tag{3.8}$$

These densities directly relate to the multitarget posterior density of random set Θ by

$$f_{k|k}(x_1, \ldots, x_n | Z_k) = j(x_1, \ldots, x_n) \tag{3.9}$$

where $x_1, \ldots, x_n$ denotes the random set of targets at time k and Z_k denotes the random set of observations.

3.2.3 Probability Generating Functionals

Let ξ be any bounded complex-valued Borel measurable function defined on complete separable metric space χ.

$$\int p_X(dX) \prod_X [\xi] = \int p_X(dx_1 \ldots dx_N) \prod_{i=1}^{N} \xi(x_i) \tag{3.10}$$

The probability generating functionals (PGFL), G_X, of a point process X is well defined, and is given by

$$G_X[\xi] = E\left(\prod_{i=1}^{N} \xi(x_i) \right) = \int p_X(dX) \prod_X [\xi] \tag{3.11}$$

where p_X is the density of probability measure P_X defined on point process X.

$$G_X[\xi] = \sum_{n=0}^{\infty} \frac{1}{n!} J_X^{(n)} |\xi, \ldots, \xi| \tag{3.12}$$

where the functional $J_X^{(n)}$ is defined by

$$J_X^{(n)}|\xi,\ldots,\xi| = \int_{\chi_1\cdots\chi_n} J_n(dx_1\times\ldots\times dx_n)\prod_{i=1}^{n}\xi(x_i) \tag{3.13}$$

Let $\mathcal{M}$ be the linear vector space of all bounded measurable complex-valued functions defined on $\mathcal{X}$ and let ξ and η be fixed elements of $\mathcal{M}$. Let $||\eta|| < 1$. If r is the largest real number s.t. $\eta+\lambda\xi \in S_g = \{\phi : ||\phi|| \leq r\}$ for $|\lambda| < r$, then,

$$G[\eta+\lambda\xi] = \sum_{k=0}^{\infty}\lambda^k\sum_{n=k}^{\infty}\int_{\chi^{(n)}}\prod_{i=1}^{k}\xi(x_i)\prod_{i=k}^{n}\eta(x_n)p_X^{(n)}(dx_1\times\ldots\times dx_n) \tag{3.14}$$

The n^{th}-order functional derivative of G is defined as

$$\delta^n_{\xi_1,\ldots,\xi_n}G[\eta] = \left[\frac{\partial^n}{\partial\lambda_1,\ldots,\lambda_n}G\left[\eta+\sum_{i=1}^{n}\lambda_i\xi_i\right]\right]|_{\lambda_1=\ldots=\lambda_n=0} \tag{3.15}$$

where $\sup_X|\eta(x)| < 1$. We can determine the n^{th}-order Janossy measure of G_X by evaluating $\delta^n_{\xi_1},\ldots,\delta^n_{\xi_n}G[\eta]$ at $\eta = 0$.

$$(d^{(n)}G_X)_0[\xi_1,\ldots,\xi_n] \triangleq \lim_{\eta\to 0}\delta^n_{\xi_1,\ldots,\xi_n}G[\xi] = J_X^{(n)}[\xi_1,\ldots,\xi_n] \tag{3.16}$$

The intensity measure V_X can be obtained by

$$V_X\cdot\xi \triangleq \int\xi(x)V_X(dx) \tag{3.17}$$

V_X is a non-negative and countably additive measure. If this measure admits a density, then it defines the Probability Hypothesis Density.

We can define the counting measure $N(\cdot|x)$, when $x = \{x_1,\ldots,x_n\}$ be a collection

of points in X, to be

$$N(A|x) = \sum_{i=1}^{n} I_A(x_i), \qquad A \subset X \tag{3.18}$$

where

$$I_A(x_i) = \begin{cases} 1 & \text{if } x_i \in A \\ 0 & \text{otherwise} \end{cases} \tag{3.19}$$

Equivalently, a counting measure of a point process X can be expressed as,

$$\prod_{i=1}^{N} \xi(x_i) = \exp \int \log \xi N(dy|x) \tag{3.20}$$

Let N_1 and N_2 be the counting measure representations of two independent point processes. Then, a third point process, which is a superposition of these two, can be defined to be,

$$N(A) = N_1(A) + N_2(A), \qquad A \subset X \tag{3.21}$$

It follows that

$$G_N[\xi] = G_{N_1}[\xi] G_{N_2}[\xi] \tag{3.22}$$

A joint probability generating functional (JPGFL), $G_{X,Y}$, of point processes X and Y can be defined by

$$G_{X,Y}[g,h] \triangleq \int \int p_{X,Y}(dx, dy) \prod_X [g] \prod_Y [h] \tag{3.23}$$

The $G_{X,Y}$ has the following properties,

$$(d^n G_{X,Y}[1,\cdot])_0[\eta_1, \ldots, \eta_n] = J_Y^{(n)}[\eta_1, \ldots, \eta_n] \tag{3.24}$$

$$(d^n G_{X,Y}[g,\cdot])_{h=0}[\eta_1, \ldots, \eta_n] = p_X(J_{Y|X}^{(n)})[\eta_1, \ldots, \eta_n | x] \prod_{(\cdot)} [g] \tag{3.25}$$

where $(\xi_1, \ldots, \xi_n)$ are complex-valued functions which are defined on the complex separable metric spaces on which elements of point processes X and Y are located. The property (3.25) is valid provided that differentiation and expectation can be interchanged. If the Janossy measure $J_{Y|X}$ admits a density $j_{Y|X}$, then it can be replaced in (3.25).

The conditional PGFL of X given $Y = y$ is defined as,

$$G_{X|Y}[\eta|y] \triangleq P_{X|Y}(\cdot|y) \cdot \prod_{(\cdot)}[\eta] \tag{3.26}$$

$$= \frac{(d^n G_{X,Y}[g, \cdot])_0[\delta_{y_1}, \ldots, \delta_{y_n}]}{(d^n G_{X,Y}[1, \cdot])_0[\delta_{y_1}, \ldots, \delta_{y_n}]} \tag{3.27}$$

3.2.4 The Prediction Equation

In PHD filter prediction, the following scenarios are considered. The target with state $x \in X_{k-1}$ survives with probability $e_{k|k-1}(x)$, evolves according to the dynamics described by the transition function $f_{k|k-1}$. The RFSs of survived targets, spawned targets and newly appeared targets are denoted by $S_{k|k-1}(x)$, $B_{k|k-1}(x)$ and Γ_k, respectively.

Theorem 3.2.1 *Suppose that the RFS of target at k is given by*

$$X_k = \left(\bigcup_{x \in X_{k-1}} S_{k|k-1}(x) \right) \bigcup \left(\bigcup_{x \in X_{k-1}} B_{k|k-1}(x) \right) \bigcup \Gamma_k \tag{3.28}$$

If the intensity measures of the multi-target probability distributions admit densities, then the prediction equation for the PHD is given by,

$$D_{k|k-1}(x|z_{1:k-1}) = \gamma_k(x) + \int \phi_{k|k-1}(x, x_{k-1}) D_{k-1|k-1}(x_{k-1}|z_{1:k-1}) dx_{k-1} \tag{3.29}$$

where $D_{k-1|k-1}$ and $D_{k|k-1}$ are the posterior and predicted densities at $k-1$ and k, respectively. γ_k is the PHD of newly appeared targets at time k. The transition kernel $\phi_{k|k-1}$ is given by

$$\phi_{k|k-1}(x, \xi) = e_{k|k-1}(\xi) f_{k|k-1}(x|\xi) + \beta_{k|k-1}(x|\xi) \tag{3.30}$$

where $\beta_{k|k-1}$ is the PHD of the RFS $B_{k|k-1}$.

Proof Let $V_{k|k-1}(\cdot|z_{1:k-1})$ and $V_k(\cdot|z_{1:k})$ denote the multitarget predicted and posterior densities, respectively. Furthermore, let these densities admit the PHDs $D_{k|k-1}$ and D_k, respectively. Let $V_{S_{k|k-1}}$, $V_{B_{k|k-1}}$ and V_{Γ_k} denote the intensity measures of the RFSs $S_{k|k-1}$, $B_{k|k-1}$ and Γ_k, respectively. The RFSs $S_{k|k-1}$, $B_{k|k-1}$ and Γ_k are independent. Therefore, by superposition (3.28) has the PGFLs $G_{S_{k|k-1}}$, $G_{B_{k|k-1}}$ and G_{Γ_k} and the intensity measure $V_{S_{k|k-1}} + V_{B_{k|k-1}} + V_{\Gamma_k}$. Using the fact that $S_{k|k-1}$ and $B_{k|k-1}$ are formed from Markov shifts and Preposition 8.2IV in [36]

$$V_{k|k-1}(A) = \langle V_{k-1}, V_{S_{k|k-1}} + V_{B_{k|k-1}} \rangle + V_{\Gamma_k} \qquad \forall A \in \mathcal{B}(\mathcal{Y}) \tag{3.31}$$

which gives the desired predicted PHD, by taking the densities.

3.2.5 The Update Equation

Let Z_k denote the RFS of the set of measurements received at time k, which may include returns from false alarm and have missed detections. Let the target originated measurements, with detection probability P_D, is distributed according to the conditional probability,

$$L_k(x, B) = P(z_k|x_k = x) \tag{3.32}$$

where L_k is assumed admits the likelihood density function l_k determined from the Radon-Nikodym derivative,

$$l_k(x, \cdot) = \frac{dL_k(x, \cdot)}{d\lambda} \tag{3.33}$$

on observation space. The RFS representing the set of measurements Z_k is given by the union of target originated measurements and the measurement set of false alarms or clutter. The point process Z_k can be described by the conditional probability measure,

$$P(z_{1:k} \in \mathcal{V}|X_k = x) \tag{3.34}$$

Theorem 3.2.2 *Suppose RFS of the set of measurements at time k is represented by*

$$Z_k = C_k \bigcup \left(\bigcup_{x \in X_k} \Theta_k(x) \right) \tag{3.35}$$

Where C_k and X_k are Poisson point processes. Furthermore, assume that the predicted target density is Poisson and that the intensity measures admit densities. Then the PHD update equation is given by

$$D_{k|k}(x|z_{1:k}) = \left[(1 - p_D) + \sum_{z \in z_{1:k}} \frac{\psi_{k,z}(x)}{\kappa_k(z) + \langle D_{k|k-1}, \Psi_{k,z} \rangle} \right] D_{k|k-1}(x|z_{1:k-1}) \tag{3.36}$$

where

$$\kappa_k(z) = \lambda_k c_k(z) \tag{3.37}$$

is the clutter model with Poisson parameter denoting the expected number of false alarms λ_k *and spacial density* c_k*, and*

$$\psi_{k,z} = p_D(x)g(z|x) \tag{3.38}$$

where p_D *is the detection probability and* $g(\cdot)$ *is the single target likelihood function.*

Proof The point processes C_k and $\Theta_k(x)$ are independent. Therefore, by superposition, the conditional PGFL $G_{z_{1:k}|X_k}[h|x]$ is given by,

$$G_{z_{1:k}|X_k}[h|x] = G_{\kappa_k}[h] \prod_{x \in X_k} G_{\theta_k(x)}[h] \tag{3.39}$$

Since x is distributed according to L_k with detection probability p_D, the PGFL $G_{\theta_k(x)}[h]$ is given by

$$G_{\theta_k(x)}[h] = (1 - p_D) + p_D H[h](x) \tag{3.40}$$

where

$$H[h](x) \triangleq \int h(z)g(z|x)dz \tag{3.41}$$

PGFL $G_{z_{1:k}|X_k}[h|x]$ can be given as

$$G_{z_{1:k}|X_k}[h|x] = G_{\kappa_k}[h] \prod_{x \in X_k} ((1 - p_D) + p_D H[h](x)) \tag{3.42}$$

Hence, using (3.25), the JPGFL $G_{X_k,z_k}[\xi,h]$ is given by,

$$G_{X_k,z_k}[\xi,h] = P_{k|k-1}\cdot(G_{z_{1:k}|X_k}[h|\cdot]\prod_{(\cdot)}[\xi]) = G_{\kappa_k}[h]G_{X_k}[\xi(1-p_D+p_DH[h])] \quad (3.43)$$

Since X_k and C_k are Poisson point processes, $G_{X_k,z_k}[\xi,h]$ is given by,

$$G_{z_{1:k}|X_k}[h|x] = \exp(V_{\kappa_k}\cdot(h-1)+V_{X_k}\cdot(\xi(1-p_D+p_DH[h]))) \quad (3.44)$$

Using (3.27), from Bayes' rule,

$$G_{X_k|z_{1:k}}[y] = \frac{(d^nG_{X_k,z_k}[z_{1:k},\cdot])_0[\delta_{z_1},\cdots,\delta_{z_n}]}{(d^nG_{X_k,z_k}[1,\cdot])_0[\delta_{z_1},\cdots,\delta_{z_n}]} \quad (3.45)$$

Then the conditional measure $V_{X_k|z_k}[\eta|y]$ is given by

$$V_{X_k|z_k}[\eta|y] = V_{X_k}\cdot\eta = \frac{(d(d^nG_{X_k,z_k[y,\cdot]})_0[\delta_{z_1},\ldots,\delta_{z_n}])_{\xi=1}[\eta]}{(d^nG_{X_k,z_k}[1,\cdot])_0[\delta_{z_1},\ldots,\delta_{z_n}]} \quad (3.46)$$

We define the notation,

$$F_{y,h}[h] \triangleq (d^nG_{X_k,z_k}[y,\cdot])_h[\xi,\ldots,\xi] \quad (3.47)$$

Then, the conditional intensity measure can be written as,

$$V_{X_k|z_k}[\eta|y] = \frac{(dF_{y,0})_1[\eta]}{F_{y,0}[1]} \quad (3.48)$$

Evaluating $F_{y,h}[\xi]$, by taking the functional derivatives, gives

$$F_{y,h}[\xi] = G_{X_k,z_k}[\xi, h] \prod_{z \in z_{1:k}} (V_{X_k} \cdot \xi\psi_{k,z} + v_{\kappa_k}(y)) \tag{3.49}$$

where $\psi_{k,z}(x) = p_D g(z|x)$. Taking the derivative at $h = 0$ and $\xi = 1$, using the chain rule, we get

$$\begin{aligned}(dF_{y,0})_1[\eta] &= V_{X_k} \cdot \eta(1 - p_D)F_{y,0}[1] + G_{X_k,z_{1:k}}[1, 0] \\ &\quad \times \prod_{z \in z_{1:k}} (V_{X_k} \cdot \xi\psi_{k,z} + v_{\kappa_k}(z)) \sum_{z \in z_{1:k}} \frac{V_{X_k} \cdot \eta\psi_{k,z}}{V_{X_k}\psi_{k,z} + v_{\kappa_k}(z)}\end{aligned} \tag{3.50}$$

Using (3.49), we get

$$(dF_{y,0})_1[\eta] = V_{X_k}\eta(1 - p_D)F_{y,0}[1] + F_{y,0}[1] \sum_{z \in z_{1:k}} \frac{V_{X_k} \cdot \eta\psi_{k,z}}{V_{X_k}\psi_{k,z} + v_{\kappa_k}(z)} \tag{3.51}$$

and taking out a factor of $F_{y,0}[1]$,

$$(dF_{y,0})_1[\eta] = F_{y,0}[1]V_{X_k} \cdot \eta \left((1 - p_D) + \sum_{z \in z_{1:k}} \frac{V_{X_k} \cdot \eta\psi_{k,z}}{V_{X_k}\psi_{k,z} + v_{\kappa_k}(z)} \right) \tag{3.52}$$

Using (3.46), the conditional measure $V_{X_k|z_{1:k}}[\eta|z]$ is given by,

$$V_{X_k|z_{1:k}}[\eta|z] = V_{X_k} \cdot \eta \left((1 - p_D) + \sum_{z \in z_{1:k}} \frac{V_{X_k} \cdot \eta\psi_{k,z}}{V_{X_k}\psi_{k,z} + v_{\kappa_k}(z)} \right) \tag{3.53}$$

Since $V_{X_k} = V_{k|k-1}(\cdot|z_{1:k-1})$ and $V_{X_k|Z_k} = V_k$, we have,

$$V_k[\eta|z] = V_{k|k-1} \left((1 - p_D) + \sum_{z \in z_{1:k}} \frac{\psi_{k,z}}{V_{k|k-1} \cdot \psi_{k,z} + v_{\kappa_k}(z)} \right) \tag{3.54}$$

which by taking the densities gives the PHD update equation.

3.3 A Physical-Space Approach for the PHD

This section describes the physical-space approach based on [43] to derive the PHD filter. This method avoids using finite point process, set derivative or higher mathematical concepts which were originally used to derive the PHD filter [69]. In this approach the surveillance region of the target state space is assumed to be consist of infinitesimally small bins, each potentially contains at most one target. The approach devise a prediction and update bin-occupancy probabilities of the recursive filter and then considered the limiting case in which the bin-occupancy probabilities tend to zero.

3.3.1 Prediction

The event that bin i contains a target is given by

$$U_k(i) = \text{“bin } i \text{ contains a target”} = \left(\bigcup_j \text{“target from } x_j \text{ went to } x_i\text{”} \right) \bigcup \text{“birth”} \tag{3.55}$$

The RHS of the above equation are mutually-exclusive because of the assumption that the bins are infinitesimally small. Thus the bin occupancy probabilities are given by

$$P(U_k(i) = 1|Z_{1:k-1}) = P(\text{“birth”}) + \sigma_j(1 - d(j))p(x_i|x_j)p(U_{k-1}(j) = 1|Z_{1:k-1}) \tag{3.56}$$

where $d(j)$ denotes the probability of target disappearance from bin j and $p(x_i|x_j)$ denotes the probability of target transition from x_j to x_i.

In the limiting case of bin width $\longrightarrow 0$, we have $p(x_i|x_j) \longrightarrow f(x_i|x_j)$, then (3.56) yields the continuous PHD prediction equation,

$$p_{k|k-1} = b(x) + \int f(x|y)(1 - d(y))p_{k-1|k-1}(y)dy \tag{3.57}$$

3.3.2 Update

Using the Bayesian update,

$$p_{k|k}(U_i) = \frac{f(Z_k|U_k(i) = 1, Z_{1:k-1})}{f(Z_k|Z_{1:k-1})} p_{k|k-1}(U_i) \tag{3.58}$$

where $p_{k|k}(U_i) = PrU_k(i) = 1|Z_{1:k-1}$.

Some targets may go undetected at time k and it is indicated by a "visibility" variable $V_k(i)$. It is a Bernoulli random variable with probability $P_d(i)$, i.e., it is unity if the target in bin i is get detected and zero otherwise. The received measurements at time step k is then given by

$$|Z_k| = \sum_j U_k(j)V_k(j) + \text{number of false alarms} \tag{3.59}$$

where the number of false alarms are Poisson distributed with parameter $\lambda|V|$, $|V|$ being the volume of the surveillance region. The first term in (3.59) is Poisson approximated by the parameter $\sum_j p_{k|k-1}(U_j)P_d(j)$, as the bin size decreases to zero. Thus the overall measurements is Poisson with parameter

$$\mu = \sum_j p_{k|k-1}(U_j)P_d(j) + \lambda|V| \tag{3.60}$$

The first case in which there is no measurements at time k, $|Z_k| = m = 0$

$$f(Z_k|U_k(i) = 1, Z_{1:k-1}) = (1 - P_d(i)) \exp(-\mu) \tag{3.61}$$

$$f(Z_k|U_k(i) = 0, Z_{1:k-1}) = \exp(-\mu) \tag{3.62}$$

and

$$\begin{aligned} f(Z_k|Z_{1:k-1}) &= f(Z_k|U_k(i) = 1, Z_{1:k-1}) p_{k|k-1}(U_i) \\ &\quad + f(Z_k|U_k(i) = 0, Z_{1:k-1})(1 - p_{k|k-1}(U_i)) \end{aligned} \tag{3.63}$$

From (3.58),(3.61),(3.62) and (3.63) we have,

$$p_{k|k}(U_i) = \frac{(1 - P_d(i)) \exp(-\mu) p_{k|k-1}(U_i)}{(1 - P_d(i)) \exp(-\mu) p_{k|k-1}(U_i) + \exp(-\mu)(1 - p_{k|k-1}(U_i))} \tag{3.64}$$

We get the PHD update equation by applying the limiting case $p_{k|k-1}(U_i) \to 0$.

$$p_{k|k}(U_i) = (1 - P_d(i)) p_{k|k-1}(U_i) \tag{3.65}$$

In case there are $|Z_k| = m > 0$ observations, we have

$$\begin{aligned} f(Z_k|U_k(i) = 1, Z_{1:k-1}) &= f(Z_k|U_k(i) = 1, V_k(i) = 1, Z_{1:k-1}) P_d(i) \\ &\quad + f(Z_k|U_k(i) = 1, V_k(i) = 1, Z_{1:k-1})(1 - P_d(i)) \end{aligned} \tag{3.66}$$

Now we consider the hypothesis that z_s originated from the target in bin i and all the other measurements $Z_k(s^c)$ originated from either from other targets or clutter,

where $s = 1, 2, \ldots, m$ and $Z_k(s^c) = \{z_1, \ldots, z_{s-1}, z_{s+1}, \ldots, z_m\}$. Then we have,

$$\begin{aligned} f(Z_k|U_k(i)=1, V_k(i)=1, Z_{1:k-1}) &= \sum_s \frac{1}{m} f(z_s|x_i) f(Z_k(s^c)|Z_{1:k-1}) \\ &\quad \times \frac{\mu^{m-1}\exp(-\mu)}{(m-1)!} \qquad (3.67) \\ f(Z_k|U_k(i)=1, V_k(i)=0, Z_{1:k-1}) &= f(Z_k|U_k(i)=0, Z_{1:k-1}) \qquad (3.68) \\ &= f(Z_k|Z_{1:k-1}) \frac{\mu^m}{m!} \exp(-\mu) \qquad (3.69) \end{aligned}$$

Then the update equation becomes,

$$p_{k|k}(U_i) = \frac{\Omega_1 p_{k|k-1}(U_i)}{\Omega_1 p_{k|k-1}(U_i) + \Omega_2 (1 - p_{k|k-1}(U_i))} \qquad (3.70)$$

where Ω_1 and Ω_2 are given by

$$\begin{aligned} \Omega_1 &= \sum \frac{\mu^{m-1}}{m!} \exp(-\mu) f(z_s|x_i) f(Z_k(s^c)|Z_{1:n-1}) P_d(i) \\ &\quad + \frac{\mu^m}{m!} \exp(-\mu) f(Z_k|Z_{1:k-1})(1 - P_d(i)) \qquad (3.71) \\ \Omega_2 &= \frac{\mu^m}{m!} \exp(-\mu) f(Z_k|Z_{1:k-1}) \qquad (3.72) \end{aligned}$$

Ω_2 dominates the denominator in the limiting case $p_{k|k-1}(U_i) \to 0$,

$$\begin{aligned} p_{k|k}(U_i) &= p_{k|k-1}(U_i) \left((1 - P_d(i)) + \sum_s \frac{P_d(i) f(z_s|x_i) f(Z_k(s^c)|Z_{1:k-1})}{f(Z_k|Z_{1:k-1})} \right) \qquad (3.73) \\ p_{k|k}(U_i) &= p_{k|k-1}(U_i) \left((1 - P_d(i)) + \sum_s \frac{P_d(i)}{\mu} \frac{f(z_s|x_i)}{f(z_s|Z_{1:k-1})} \right) \qquad (3.74) \end{aligned}$$

We have

$$\begin{aligned} f(z_s|Z_{1:k-1}) &= \frac{1}{|V|}P(z_s \text{ is false alarm } |Z_{1:k-1}|) \\ &\quad + \sum_i f(z_s|x_i)P(z_s \text{ from } x_i|Z_{1:k-1}) \qquad (3.75) \\ &= \frac{1}{|V|}\frac{\lambda|V|}{\lambda|V| + \sum_j p_{k|k-1}(U_j)P_d(j)} \\ &\quad + \sum_j f(z_s|x_i)\frac{p_{k|k-1}(U_i)P_d(i)}{\lambda|V| + \sum_j p_{k|k-1}(U_j)P_d(j)} \qquad (3.76) \\ &= \frac{1}{\mu}\left(\lambda + \sum_i f(z_s|z_i)p_{k|k-1}(U_i)P_d(i)\right) \qquad (3.77) \end{aligned}$$

and

$$p_{k|k}(U_i) = p_{k|k-1}(U_i)\left((1 - P_d(i)) + \sum_s \frac{P_d(i)f(z_s|x_i)}{\lambda + \sum_j f(z_s|x_j)p_{k|k-1}(U_j)}\right) \qquad (3.78)$$

In the limiting case $p_{k|k-1} \to 0$, We get the PHD update equation,

$$D_{k|k}(x) = D_{k|k-1}(x)\left[(1 - P_d(x)) + \sum_s \frac{P_d(x)f(z_s|x)}{\lambda + \int f(z_s|y)D_{k|k-1}(y)dy}\right] \qquad (3.79)$$

Chapter 4

PHD Filter Implementations

4.1 Sequential Monte Carlo PHD filter

This section describes the SMC approach to the PHD filter. This approach provides a mechanism to represent the posterior probability hypothesis density by a set of random samples or particles, which consists of state information with associated weights, to approximate the PHD. The advantage of this method is that the number of particles can be adaptively allocated in a way that a constant ratio between the number of particles and the expected number of targets is maintained. This has a significant effect on the computational complexity of the algorithm that the complexity does not increase exponentially but linear with the increasing number of targets.

The SMC implementation considered here is structurally similar to the Sampling Importance Resampling (SIR) type of particle filter [4]. Let the posterior PHD $D_{k-1|k-1}(\mathbf{x}_{k-1}|Z_{1:k-1})$ be represented by a set of particles $\left\{w_{k-1}^{(s)}, \mathbf{x}_{k-1}^{(s)}\right\}_{s=1}^{L_{k-1}}$. That

is,

$$D_{k-1|k-1}(\mathbf{x}_{k-1}|Z_{1:k-1}) = \sum_{s=1}^{L_{k-1}} w_{k-1}^{(s)} \delta(\mathbf{x}_{k-1} - \mathbf{x}_{k-1}^{(s)}) \tag{4.1}$$

where $\delta(\cdot)$ is the Dirac Delta function.

In contrast to particle filters, the total weight $\sum_{s=1}^{L_{k-1}} w_{k-1}^{(s)}$ is not equal to one; instead, it gives the expected number of targets n_{k-1}^{X} at time step $(k-1)$, which follows from the property that the integral of the PHD over the state space gives the expected number of targets.

4.1.1 Prediction

We now apply importance sampling to generate state samples that approximate the predicted PHD $D_{k|k-1}(\mathbf{x}_k|Z_{1:k-1})$. We generate $\{\mathbf{x}_{k|k-1}^{(s)}\}_{s=1}^{L_{k-1}}$ state samples from the proposal density

$q_k(\cdot|\mathbf{x}_{k-1}, Z_k)$ and i.i.d. state samples $\{\mathbf{x}_{k|k-1}^{(s)}\}_{s=L_{k-1}+1}^{L_{k-1}+J_k}$ corresponding to new spontaneously born targets from another proposal density $p_k(\cdot|Z_k)$. That is,

$$\mathbf{x}_{k|k-1}^{(s)} \sim \begin{cases} q_k(\cdot|\mathbf{x}_{k-1}, Z_k) & s = 1, \ldots, L_{k-1} \\ \\ p_k(\cdot|Z_k) & s = L_{k-1}+1, \ldots, L_{k-1}+J_k \end{cases} \tag{4.2}$$

Then, the weighted approximation of the predicted PHD is given by

$$D_{k|k-1}(\mathbf{x}_k|Z_{1:k-1}) = \sum_{s=1}^{L_{k-1}+J_k} w_{k|k-1}^{(s)} \delta(\mathbf{x}_k - \mathbf{x}_{k|k-1}^{(s)}) \tag{4.3}$$

where

$$w_{k|k-1}^{(s)} = \begin{cases} \frac{e_{k|k-1}(\mathbf{x}_{k|k-1}^{(s)}) f_{k|k-1}(\mathbf{x}_{k|k-1}^{(s)}|\mathbf{x}_{k-1}^{(s)}) + b_{k|k-1}(\mathbf{x}_{k|k-1}^{(s)}|\mathbf{x}_{k-1}^{(s)})}{q_k(\mathbf{x}_{k|k-1}^{(s)}|\mathbf{x}_{k-1}^{(s)}, Z_k)} w_{k-1}^{(s)} & s = 1, \ldots, L_{k-1} \\ \frac{\gamma_k(\mathbf{x}_{k|k-1}^{(s)})}{p_k(\mathbf{x}_{k|k-1}^{(s)}|Z_k)} \frac{1}{J_k} & s = L_{k-1}+1, \ldots, L_{k-1}+J_k \end{cases} \tag{4.4}$$

The functions that characterize the Markov target transition density $f_{k|k-1}(\cdot)$, target spawning $b_{k|k-1}(\cdot)$ and entry of new targets $\gamma_k(\cdot)$ in (5.14) are conditioned on the target motion model.

4.1.2 Update

With the available set of measurements Z_k at time step k, the updated particle weights can be calculated by

$$w_k^{*(s)} = \left[(1 - p_D(\mathbf{x}_{k|k-1}^{(s)})) + \sum_{i=1}^{N_k^Z} \frac{p_D(\mathbf{x}_{k|k-1}^{(s)}) f_{k|k}(\mathbf{z}_k^i|\mathbf{x}_{k|k-1}^{(s)})}{\lambda_k c_k(\mathbf{z}_k^i) + \Psi_k(\mathbf{z}_k^i)} \right] w_{k|k-1}^{(s)} \tag{4.5}$$

where

$$\Psi_k(z_k^i) = \sum_{s=1}^{L_{k-1}+J_k} p_D(\mathbf{x}_{k|k-1}^{(s)}) f_{k|k}(\mathbf{z}_k^i|\mathbf{x}_{k|k-1}^{(s)},) w_{k|k-1}^{(s)} \tag{4.6}$$

and $f_{k|k}(\cdot)$ is the single-target/single-sensor measurement likelihood function.

4.1.3 Resample

To perform resampling, since the weights are not normalized to unity in PHD filters, the expected number of targets is calculated by summing up the total weights, i.e.,

$$\hat{n}_k^X = \sum_{s=1}^{L_{k-1}+J_k} w_k^{*(s)} \tag{4.7}$$

Then the updated particle set $\left\{w_k^{*(s)}/n_k^X, \mathbf{x}_{k|k-1}^{(s)}\right\}_{s=1}^{L_{k-1}+J_k}$ is resampled to get $\left\{w_k^{(s)}/n_k^X, \mathbf{x}_k^{(s)}\right\}_{s=1}^{L_k}$ such that the total weight after resampling remains n_k^X. Now, the discrete approximation of the updated posterior PHD at time step k is given by

$$D_{k|k}(\mathbf{x}_k|Z_{1:k}) = \sum_{s=1}^{L_k} w_k^{(s)} \delta(\mathbf{x}_k - \mathbf{x}_k^{(s)}) \tag{4.8}$$

4.2 Gaussian Mixture PHD Filter

This section describes the linear Gaussian multitarget model and the PHD filter based on Gaussian mixture [9, 10]. This approach gives an analytic solution to the PHD recursion for linear Gaussian target dynamics and Gaussian birth model and it is analogous to Kalman filter in a single target Bayes filter. When the initial prior density is Gaussian mixture, this approach shows that the posterior density at any subsequent time step is also Gaussian. This filter propagates the Gaussian mixture posterior intensity in time as the measurements arrive in the same way as the Gaussian sum filter [2, 96]. The main difference between the Gaussian sum filter and this approach is that former propagates a probability density using the Bayes recursion and latter propagates the PHD using PHD recursion. One of the advantage

of the Gaussian mixture model is that it allows state estimates to be extracted from the posterior intensity in a much more reliable form than the clustering method in SMC implementation.

In linear Gaussian multitarget model, we assume that each target follows linear Gaussian dynamics, i.e.,

$$f_{k|k-1}(x|\xi) = \mathcal{N}(x; F_{k-1}\xi, Q_{k-1}) \tag{4.9}$$

$$g_k = \mathcal{N}(z; H_k x, R_k) \tag{4.10}$$

where $\mathcal{N}(;m,P)$ denotes a Gaussian density with mean m and covariance P. F_{k-1} is the state transition matrix, Q_{k-1} is the process noise covariance, H_k is the observation matrix and R_k is the measurement noise covariance.

Gaussian mixture form RFS of the birth intensity is given by,

$$\gamma_k(x) = \sum_{i=1}^{J^{(\gamma)}_k} w_k^{(\gamma,i)} \mathcal{N}(x; m_k^{(\gamma,i)}, P_k^{(\gamma,i)}) \tag{4.11}$$

where $J^{(\gamma)}_k, w_k^{(\gamma,i)}, m_k^{(\gamma,i)}, P_k^{(\gamma,i)}, i = 1, \ldots, J_{\gamma,k}$, are known model parameters that describes the target birth model.

Similarly, Gaussian mixture form RFS of the spawned intensity is given by,

$$\beta_{k|k-1}(x|\gamma) = \sum_{i=1}^{J^{(\beta)}_k} w_k^{(\beta,j)} \mathcal{N}(x; \xi + m_k^{\beta,j}, P_k^{\beta,j}) \tag{4.12}$$

where $J^{(\beta)}_k, w_k^{(\beta,j)}, m_k^{(\beta,j)}, P_k^{(\beta,j)}, j = 1, \ldots, J_{\beta,k}$ are the known parameters that describes the target spawning.

4.2.1 Gaussian Mixture PHD Recursion

The posterior intensity at time $k-1$ is a Gaussian mixture of the form

$$v_{k-1} = \sum_{i=1}^{J_{k-1}} w_{k-1}^{(i)} \mathcal{N}(x; m_{k-1}^{(i)}, P_{k-1}^{(i)}) \tag{4.13}$$

Then the predicted intensity is given by

$$v_{k|k-1}(x) = v_{S,k|k-1}(x) + v_{\beta,k|k-1} + \gamma_k(x) \tag{4.14}$$

where

$$\begin{aligned}
v_{S,k|k-1}(x) &= p_{S,k} \sum_{j=1}^{J_{k-1}} w_{k-1}^{(j)} \mathcal{N}(x; m_{S,k|k-1}^{(j)}, P_{S,k|k-1}^{(j)}) && (4.15)\\
m_{S,k|k-1}^{(j)} &= F_{k-1} m_{k-1}^{(j)} && (4.16)\\
P_{S,k|k-1}^{(j)} &= Q_{k-1} + F_{k-1} P_{k-1}^{(j)} F_{k-1}^T && (4.17)\\
v_{\beta,k|k-1}(x) &= \sum_{j=1}^{J_{k-1}} \sum_{l=1}^{J_{\beta,k}} w_{k-1}^{(j)} w_{\beta,k}^{(l)} \mathcal{N}(x; m_{\beta,k|k-1}^{(j,l)}, P_{\beta,k|k-1}^{(j,l)}) && (4.18)\\
m_{\beta,k|k-1}^{(j,l)} &= F_{\beta,k-1}^{(l)} m_{k-1}^{(j)} + d_{\beta,k-1}^{(l)} && (4.19)\\
P_{\beta,k|k-1}^{(j,l)} &= Q_{\beta,k-1}^{(l)} + F_{\beta,k-1}^{(l)} P_{\beta,k-1}^{(j)} (F_{\beta,k-1}^{(l)})^T && (4.20)\\
& && (4.21)
\end{aligned}$$

The predicted intensity in Gaussian mixture form is then given by

$$v_{k|k-1}(x) = \sum_{i=1}^{J_{k|k-1}} w_{k|k-1}^{(i)} \mathcal{N}(x; m_{k|k-1}^{(i)}, P_{k|k-1}^{(i)}) \tag{4.22}$$

The posterior PHD intensity at time k is given by the Gaussian mixture,

$$v_k(x) = (1 - p_{D,k})v_{k|k-1}(x) + \sum_{z \in Z_k} v_{D,k}(x; z) \tag{4.23}$$

where

$$\begin{aligned}
v_{D,k}(x; z) &= \sum_{j=1}^{J_{k|k-1}} w_k^{(j)}(z)\mathcal{N}(x; m_{k|k}^{(j)}(z), P_{k|k}^{(j)}) && (4.24)\\
w_k^{(j)}(z) &= \frac{p_{D,k} w_{k|k-1}^{(j)} q_k^{(j)}(z)}{\kappa_k(z) + p_{D,k} \sum_{l=1}^{J_{k|k-1}} w_{k|k-1}^{(l)} q_k^{(l)}(z)} && (4.25)\\
m_{k|k}^{(j)}(z) &= m_{k|k-1}^{(j)} + K_k^{(j)}(z - H_k m_{k|k-1}^{(j)}) && (4.26)\\
P_{k|k}^{(j)} &= [I - K_k^{(j)} H_k] P_{k|k-1}^{(j)} && (4.27)\\
K_k^{(j)} &= P_{k|k-1}^{(j)} H_k^T (H_k P_{k|k-1}^{(j)} H_k^T + R_k)^{-1} && (4.28)
\end{aligned}$$

As the time progress, Gaussian mixture PHD filter suffers from computation problems associated with the increasing number of Gaussian components. The number of Gaussian terms propagated to the next step can be reduced by a pruning procedure. A good approximation of the Gaussian mixture intensity

$$v_k(x) = \sum_{i=1}^{J_k} w_k^{(i)} \mathcal{N}(x; m_k^{(i)}, P_k^{(i)}) \tag{4.29}$$

can be obtained by truncating the Gaussian components that have weaker weights $w_k^{(i)}$. This can be done by removing the Gaussian components that have weights below a threshold value. Further, some of the Gaussian components that are close to each other could be approximated by a single Gaussian component by merging them into one.

The extraction of individual multitarget state estimates is straight forward in Gaussian mixture filter, since the means of the constituent Gaussian components are indeed the local maxima of the posterior density, provided that they are reasonably well separated. After the pruning procedure, the closely space Gaussian components would have been merged. In general, selecting $\hat{N}_k$ highest peaks of the posterior density may result in state estimates that corresponds to Gaussian with weak weights since the height of each peak depends on both the weight and covariance. Thus, usually in Gaussian mixture filter, means of the Gaussian that have weights greater than a threshold 0.5 will be selected.

Chapter 5

A Multiple Model PHD Filter

5.1 Introduction

Tracking multiple targets with uncertain target dynamics is a difficult problem, especially with nonlinear state and/or measurement equations. With multiple targets, representing the full posterior distribution over target states is not practical. The problem becomes even more complicated when the number of targets varies, in which case the dimensionality of the state space itself becomes a discrete random variable. With maneuvering targets, detecting and tracking the changes in the target motion model also become important.The target dynamic model uncertainty can be resolved by assuming multiple models for possible motion modes and then combining the mode-dependent estimates in a manner similar to the one used in the Interacting Multiple Model (IMM) estimator. This thesis propose a multiple-model implementation of the PHD filter, which approximates the PHD by a set of weighted random samples propagated over time using Sequential Monte Carlo methods. The resulting filter can handle nonlinear, non-Gaussian dynamics with uncertain model parameters

in multisensor-multitarget tracking scenarios.

5.2 Problem Formulation

This section first describes the mathematical formulation of the problem of tracking multiple maneuvering targets and the solution to it using the MMPHD algorithm. This is followed by the implementation details of the algorithm using an SMC approach.

The general parameterized target dynamics of the j th target is given by

$$\mathbf{x}_k^j = a_{r,k}(\mathbf{x}_{k-1}^j, \mathbf{w}_{r,k-1}, r_k^j) \qquad j = 1, \ldots, N_k^X, \tag{5.1}$$

where $\mathbf{x}_k^j$ is the target state vector at time step k, N_k^X is the number of targets at time step k, $a_{r,k}$, in general, is a nonlinear function, $\mathbf{w}_{r,k-1}$ is the mode-dependent process noise vector of known statistics, and $r_k^j \in \{1, \ldots, N_r\}$ is the model index parameter governed by an underlying Markov process with the model transition probability

$$f_{k|k-1}(r_k = q \,|\, r_{k-1} = p) = h_{pq}. \tag{5.2}$$

We denote the PHD of the mode-dependent full target state distribution at time step k by $D_{k|k}(\mathbf{x}_k, r_k|Z_{1:k})$. Once again we will consider target survival, target spawning and appearance of completely new targets given by $e_{k|k-1}(\cdot)$, $b_{k|k-1}(\cdot)$ and $\gamma_k(\cdot)$, respectively.

The measurements originate from either targets or clutter. We denote the total number of measurements at time step k by N_k^Z. The target-originated measurement

due to the j th target on i th sensor is given by

$$\mathbf{z}_k^{i,j} = g_{r,k}(\mathbf{x}_k^j, \mathbf{v}_{r,k}, r_k^j) \tag{5.3}$$

where $g_{r,k}$, in general, is a nonlinear function and $\mathbf{v}_{r,k}$ is a mode-dependent measurement noise vector of known statistics. Some targets may not be detected at time step k. We denote the probability that the target with state $\mathbf{x}_k$ is detected at time step k by $p_D(\mathbf{x}_k)$.

5.3 The Multiple Model Approach to Target Tracking

The multiple model approach to tracking maneuvering targets by detecting maneuvers and identifying the appropriate model has been shown to be highly effective. In this approach, a finite number of filters operate in parallel, and the target motion is assumed to follow one of the models in the mode set of the tracker. Considering the compromise between complexity and performance, the interacting multiple model (IMM) approach has been shown to be the most effective of the known multiple model approaches, including the generalized pseudo-Bayesian (GPB) [1,31] algorithms . A GPB algorithm of order n (GPB_n) requires N_r^n filters in its bank, where N_r is the number of models. The IMM estimator performs nearly as well as GPB_2, but requires only N_r filters operating in parallel. Thus, it has significantly less computational complexity, which is almost same as that of GPB_1. Further, the IMM estimator does not require maneuver detection decisions as in the case of Variable State Dimension (VSD) filter [13] algorithms, and undergoes a soft switching between models based

on the updated mode probabilities.

The multiple-model approach used in the proposed MMPHD algorithm is structurally similar to that of the IMM estimator in the mixing and combination stages. The MMPHD filter also requires, implicitly, only N_r PHD filters to operate in parallel when there are N_r models describing target dynamics. Further, the MMPHD filter does not require a maneuver detection decision and undergoes a soft switching between the models. The IMM estimator differs from the multiple model approach presented in this chapter of the thesis in that the former uses only first and second order statistics of the target densities in the the mixing and combination stages. The techniques used in the IMM estimator cannot be applied to combine the mode-dependent PHD filter outputs since the densities are not necessarily Gaussian. Further, the densities might be multi-modal when they represent multiple targets; thus, it is not reasonable to approximate them by only first and second order statistics. Thus, the multiple-model approach presented in this chapter uses the branched true densities, i.e., the complete densities conditioned on each model, in the mixing and update stages. This approach can handle multi-modal target densities at the expense of increased computational load compared to the IMM estimator. One cycle of the recursive MMPHD algorithm can be described in three stages as follows.

5.3.1 The Mixing Stage

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

prior density $D_{k-1|k-1}(\mathbf{x}_{k-1}, r_{k-1} = p|Z_{1:k-1})$, i.e.,

$$\tilde{D}_{k|k-1}(\mathbf{x}_{k-1}, r_k = q|Z_{1:k-1}) = \sum_{p=1}^{N_r} D_{k-1|k-1}(\mathbf{x}_{k-1}, r_{k-1} = p|Z_{1:k-1}) h_{pq} \qquad q = 1, \ldots, N_r \tag{5.4}$$

Target spawning, birth and disappearance are not considered at the mixing stage and they are considered in the prediction stage. The densities $\tilde{D}_{k|k-1}(\cdot)$ and $D_{k-1|k-1}(\cdot)$ in (5.4) are similar to probability densities except that they do not integrate to unity. The mixing described in (5.4) is then similar to total probability theorem.

5.3.2 The Prediction Stage

Once the initial density for the PHD filter that is matched to target model q is calculated, the mode-dependent predicted density is calculated as

$$\begin{aligned} D_{k|k-1}(\mathbf{x}_k, r_k = q|Z_{1:k-1}) \;=\; & \gamma_k(\mathbf{x}_k, r_k = q) \\ & + \int \left[e_{k|k-1}(\mathbf{x}_{k-1}) f_{k|k-1}(\mathbf{x}_k|\mathbf{x}_{k-1}, r_k = q) + b_{k|k-1}(\mathbf{x}_k|\mathbf{x}_{k-1}, r_k = q) \right] \\ & \quad \times \tilde{D}_{k|k-1}(\mathbf{x}_{k-1}, r_k = q|Z_{1:k-1}) d\mathbf{x}_{k-1} \end{aligned} \tag{5.5}$$

The prediction of the mode-dependent PHD cannot be just applied to that of the single-model PHD, which is matched to the target model, because the PHD of spontaneous target birth $\gamma_k(\cdot)$ and target spawning $b_{k|k-1}(\cdot)$ are also mode dependent. The integral of the mode-dependent PHD $D_{k|k-1}(\mathbf{x}_k, r_k = q|Z_{1:k-1})$ over a region gives the expected (predicted) number of targets in that region assuming all the targets are travelling with the target dynamics described by model q.

5.3.3 The Update Stage

With the availability of measurements at time step k, the mode-dependent updated density is calculated as (for $q = 1, \ldots, N_r$)

$$D_{k|k}(\mathbf{x}_k, r_k = q|Z_{1:k}) \cong \left[\sum_{\mathbf{z}_k \in Z_k} \frac{p_D(\mathbf{x}_k) f_{k|k}(\mathbf{z}_k|\mathbf{x}_k, r_k = q)}{\lambda_k c_k(\mathbf{z}_k) + \psi_k(\mathbf{z}_k|Z_{k-1})} + (1 - p_D(\mathbf{x}_k))\right] \times D_{k|k-1}(\mathbf{x}_k, r_k = q|Z_{1:k-1}) \quad (5.6)$$

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

$$\psi_k(\mathbf{z}_k|Z_{1:k-1}) = \int p_D(\mathbf{x}_k) f_{k|k}(\mathbf{z}_k|\mathbf{x}_k, r_k = q) D_{k|k-1}(\mathbf{x}_k, r_k = q|Z_{1:k-1}) d\mathbf{x}_k \quad (5.7)$$

Although there is no explicit mode probability update in the MMPHD filter, the update stage implicitly updates the mode probabilities as well. Therefore, the updated mode probability for a particular model can be calculated by integrating the mode-dependent updated PHD divided by the total expected number of targets. The total number of targets can be found by summing up all the integrals of the updated mode-dependent PHDs. The updated mode probability for a particular target, assuming that the target densities do not overlap, can be found by integrating the mode-dependent PHD over the region in which that target is represented. However, it is not necessary to calculate the numerical value of the updated mode probability in the recursive MMPHD filter since the mode-dependent density as a whole is used in the mixing stage to feed the filter.

Using multiple-model algorithms for benign nonmaneuvering targets might degrade the performance of the tracker and increases the computational load. However, with higher target maneuverability, a multiple model approach is needed. The

decision about whether a multiple model is preferred or not is done based on the maneuvering index [13], which quantifies the maneuverability of the target in terms of the process noise, sensor measurement noise and sensor revisit interval. A study that compares the IMM estimator with the Kalman filter based on the maneuvering index is given in [60].

5.4 Particle Implementation

This section describes the SMC approach to the MMPHD filter. This approach provides a mechanism to represent the posterior multiple model probability hypothesis density by a set of random samples or particles, which consists of state and model information with associated weights. The key idea in this implementation is to include the model index parameters in the sample set with the associated weights to represent the mode-dependent posterior density. Although it does not appear explicitly, there are several PHD filters, each matched to different target dynamics, running in parallel in the MMPHD filter. The model index parameter in a sample directs the MMPHD filter to choose the PHD filter that matches the associated target model. Further, the number of samples used in each PHD filter is not necessarily the same for different target models. Since the mode probabilities are updated through the update step of the filter, the PHD filter that matches the target motion will contain the higher number of samples. This makes the filter computationally efficient compared to a filter that uses an equal number of samples to represent the PHD matched to each model.

The particle filter implementation of the proposed algorithm can be considered as a special case of the algorithm given in [6] with some modifications. The target

state is augmented with the model index parameter and the prediction and update operators of the algorithm given in [6] are modified in order to include target motion model uncertainty.

The SMC implementation considered here is structurally similar to the Sampling Importance Resampling (SIR) type of particle filter [4]. Let the posterior MMPHD $D_{k-1|k-1}(\mathbf{x}_{k-1}, r_{k-1}|Z_{1:k-1})$ be represented by a set of particles $\left\{w_{k-1}^{(s)}, \mathbf{x}_{k-1}^{(s)}, r_{k-1}^{(s)}\right\}_{s=1}^{L_{k-1}}$. That is,

$$D_{k-1|k-1}(\mathbf{x}_{k-1}, r_{k-1}|Z_{1:k-1}) = \sum_{s=1}^{L_{k-1}} w_{k-1}^{(s)} \delta(\mathbf{x}_{k-1} - \mathbf{x}_{k-1}^{(s)}, r_{k-1} - r_{k-1}^{(s)}) \tag{5.8}$$

where $\delta(\cdot)$ is the Dirac Delta function.

In contrast to particle filters, the total weight $\sum_{s=1}^{L_{k-1}} w_{k-1}^{(s)}$ is not equal to one; instead, it gives the expected number of targets n_{k-1}^{X} at time step $(k-1)$, which follows from the property that the integral of the PHD over the state space gives the expected number of targets. Further, the posterior model probabilities corresponding to a particular target are approximately equal to the proportion of the total sample weights in the index set $\left\{r_{k-1}^{(s)}\right\}_{s=1}^{L_{k-1}}$ for each model corresponding to that target.

5.4.1 Prediction

The MMPHD filter prediction step involves predicting the models in addition to state prediction. The model prediction for existing targets is performed based on the model transition probabilities $f_{k|k-1}(r_k|r_{k-1})$. Model samples $\{r_{k|k-1}^{(s)}\}_{s=1}^{L_{k-1}}$ from the model-predicted MMPHD $\tilde{D}_{k|k-1}(\mathbf{x}_{k-1}, r_k|Z_{1:k-1})$ are generated by importance sampling from a proposal density $\pi_k(\cdot|r_{k-1})$. We generate i.i.d. model samples $\{r_{k|k-1}^{(s)}\}_{s=L_{k-1}+1}^{L_{k-1}+J_k}$ corresponding to new spontaneously born targets by sampling from

another proposal density $\beta_k(\cdot)$. Then,

$$r_{k|k-1}^{(s)} \sim \begin{cases} \pi_k(\cdot|r_{k-1}) & s = 1, \ldots, L_{k-1} \\ \\ \beta_k(\cdot) & s = L_{k-1}+1, \ldots, L_{k-1}+J_k \end{cases} \tag{5.9}$$

where $\pi_k(\cdot|r_{k-1})$ and $\beta_k(\cdot)$ are probability mass functions. Then, a discrete weighted approximation to the model predicted MMPHD $\tilde{D}_{k|k-1}(\mathbf{x}_{k-1}, r_k|Z_{1:k-1})$ is given by

$$\tilde{D}_{k|k-1}(\mathbf{x}_{k-1}, r_k|Z_{1:k-1}) = \sum_{s=1}^{L_{k-1}+J_k} \varpi_{k|k-1}^{(s)} \delta(\mathbf{x}_{k-1} - \mathbf{x}_{k-1}^{(s)}, r_k - r_{k|k-1}^{(s)}) \tag{5.10}$$

where

$$\varpi_{k|k-1}^{(s)} = \begin{cases} \frac{f_{k|k-1}(r_{k|k-1}^{(s)}|r_{k-1}^{(s)})}{\pi_k(r_{k|k-1}^{(s)}|r_{k-1}^{(s)})} w_{k-1}^{(s)} & s = 1, \ldots, L_{k-1} \\ \\ \frac{\theta_k(r_{k|k-1}^{(s)})}{\beta_k(r_{k|k-1}^{(s)})} \frac{1}{J_k} & s = L_{k-1}+1, \ldots, L_{k-1}+J_k \end{cases} \tag{5.11}$$

In (5.11), the probability mass function $\theta_k(\cdot)$ denotes the model distribution of spontaneous target births at time step k. The number of new particles J_k can be a function of time step k to accommodate varying number of targets.

We now apply importance sampling to generate state samples that approximate the predicted MMPHD $D_{k|k-1}(\mathbf{x}_k, r_k|Z_{1:k-1})$. We generate $\{\mathbf{x}_{k|k-1}^{(s)}\}_{s=1}^{L_{k-1}}$ state samples from the proposal density

$q_k(\cdot|\mathbf{x}_{k-1}, r_{k|k-1}, Z_k)$ and i.i.d. state samples $\{\mathbf{x}_{k|k-1}^{(s)}\}_{s=L_{k-1}+1}^{L_{k-1}+J_k}$ corresponding to new

spontaneously born targets from another proposal density $p_k(\cdot|r_{k|k-1}, Z_{k-1})$. That is,

$$\mathbf{x}_{k|k-1}^{(s)} \sim \begin{cases} q_k(\cdot|\mathbf{x}_{k-1}, r_{k|k-1}, Z_k) & s = 1, \ldots, L_{k-1} \\ \\ p_k(\cdot|r_{k|k-1}, Z_k) & s = L_{k-1}+1, \ldots, L_{k-1}+J_k \end{cases} \tag{5.12}$$

Then, the weighted approximation of the predicted MMPHD is given by

$$D_{k|k-1}(\mathbf{x}_k, r_k|Z_{1:k-1}) = \sum_{s=1}^{L_{k-1}+J_k} w_{k|k-1}^{(s)} \delta(\mathbf{x}_k - \mathbf{x}_{k|k-1}^{(s)}, r_k - r_{k|k-1}^{(s)}) \tag{5.13}$$

where

$$w_{k|k-1}^{(s)} = \begin{cases} \frac{e_{k|k-1}(\mathbf{x}_{k|k-1}^{(s)}) f_{k|k-1}(\mathbf{x}_{k|k-1}^{(s)}|\mathbf{x}_{k-1|k-1}^{(s)}, r_{k|k-1}^{(s)}) + b_{k|k-1}(\mathbf{x}_{k|k-1}^{(s)}|\mathbf{x}_{k-1|k-1}^{(s)}, r_{k|k-1}^{(s)})}{q_k(\mathbf{x}_{k|k-1}^{(s)}|\mathbf{x}_{k|k-1}^{(s)}, r_{k|k-1}^{(s)}, Z_k)} \varpi_{k|k-1}^{(s)} \\ \qquad\qquad s = 1, \ldots, L_{k-1} \\ \\ \frac{\gamma_k(\mathbf{x}_{k|k-1}^{(s)}|r_{k|k-1}^{(s)})}{p_k(\mathbf{x}_{k|k-1}^{(s)}|r_{k|k-1}^{(s)}, Z_k)} \varpi_{k|k-1}^{(s)} \qquad s = L_{k-1}+1, \ldots, L_{k-1}+J_k \end{cases} \tag{5.14}$$

The functions that characterize the Markov target transition density $f_{k|k-1}(\cdot)$, target spawning $b_{k|k-1}(\cdot)$ and entry of new targets $\gamma_k(\cdot)$ in (5.14) are conditioned on the particular motion model. Thus, although it is not explicitly explained by the equations, there are N_r PHD filters running in parallel in this implementation.

5.4.2 Update

With the available set of measurements Z_k at time step k, the updated particle weights can be calculated by

$$w_k^{*(s)} = \left[(1 - p_D(\mathbf{x}_{k|k-1}^{(s)})) + \sum_{i=1}^{N_k^Z} \frac{p_D(\mathbf{x}_{k|k-1}^{(s)}) f_{k|k}(\mathbf{z}_k^i | \mathbf{x}_{k|k-1}^{(s)}, r_{k|k-1}^{(s)})}{\lambda_k c_k(\mathbf{z}_k^i) + \Psi_k(\mathbf{z}_k^i)} \right] w_{k|k-1}^{(s)} \tag{5.15}$$

where

$$\Psi_k(z_k^i) = \sum_{s=1}^{L_{k-1}+J_k} p_D(\mathbf{x}_{k|k-1}^{(s)}) f_{k|k}(\mathbf{z}_k^i | \mathbf{x}_{k|k-1}^{(s)}, r_k^{(s)}) w_{k|k-1}^{(s)} \tag{5.16}$$

The single-target/single-sensor measurement likelihood function $f_{k|k}(\cdot)$ in (5.15) and (5.16) are written as dependent on the model, considering a general case in which the measurement model can also be mode dependent.

5.4.3 Resample

To perform resampling, since the weights are not normalized to unity in PHD filters, the expected number of targets is calculated by summing up the total weights, i.e.,

$$\hat{n}_k^X = \sum_{s=1}^{L_{k-1}+J_k} w_k^{*(s)} \tag{5.17}$$

Then the updated particle set $\left\{ w_k^{*(s)}/n_k^X, \mathbf{x}_{k|k-1}^{(s)}, r_{k|k-1}^{(s)} \right\}_{s=1}^{L_{k-1}+J_k}$ is resampled to get $\left\{ w_k^{(s)}/n_k^X, \mathbf{x}_k^{(s)}, r_k^{(s)} \right\}_{s=1}^{L_k}$ such that the total weight after resampling remains n_k^X. Now, the discrete approximation of the updated posterior MMPHD at time step k is given by

$$D_{k|k}(\mathbf{x}_k, r_k | Z_{1:k}) = \sum_{s=1}^{L_k} w_k^{(s)} \delta(\mathbf{x}_k - \mathbf{x}_k^{(s)}, r_k - r_k^{(s)}) \tag{5.18}$$

The mode-dependent posterior PHDs can be easily identified by grouping the particles based on model index parameters.

5.5 Simulations

5.5.1 Range and Bearing Sensor

In this section, a 2-D tracking example is presented to compare the MMPHD filter's performance with those of single model PHD filters. In this example, two maneuvering targets, namely, target 1 and target 2, are traveling with initial target positions $(0, 0)\,\mathrm{m}$ and $(-200, 0)\,\mathrm{m}$ respectively. The initial velocities of targets 1 and 2 are $(10, 20)\,\mathrm{ms}^{-1}$ and $(20, 10)\,\mathrm{ms}^{-1}$, respectively. The targets are assumed to switch their dynamics between two different motion models, namely, constant velocity and coordinated turn model. The target dynamics that characterizes the constant velocity model ($r_k = 1$) is given by

$$\mathbf{x}_k^i = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{x}_{k-1}^i + \begin{bmatrix} \frac{1}{2}T^2 & 0 \\ T & 0 \\ 0 & \frac{1}{2}T^2 \\ 0 & T \end{bmatrix} \mathbf{w}_k^i \tag{5.19}$$

where $\mathbf{x}_k^i = [x_k^i, \dot{x}_k^i, y_k^i, \dot{y}_k^i]$ is the state of the i th target, which consists target position (x_k^i, y_k^i) and target velocity $(\dot{x}_k^i, \dot{y}_k^i)$ at time step k, $T = 1$ is the time interval between time steps $k-1$ and k, and $\mathbf{w}_k^i$ is i.i.d sequence of standard Gaussian variables of dimension 2. A similar target dynamics that characterizes the coordinated turn model

$(r_k = 2)$ is given by

$$\mathbf{x}_k^i = \begin{bmatrix} 1 & \frac{\sin \Omega T}{\Omega} & 0 & -\frac{1-\cos \Omega T}{\Omega} \\ 0 & \cos \Omega T & 0 & -\sin \Omega T \\ 0 & \frac{1-\cos \Omega T}{\Omega} & 1 & \frac{\sin \Omega T}{\Omega} \\ 0 & \sin \Omega T & 0 & \cos \Omega T \end{bmatrix} \mathbf{x}_{k-1}^i + \begin{bmatrix} \frac{1}{2}T^2 & 0 \\ T & 0 \\ 0 & \frac{1}{2}T^2 \\ 0 & T \end{bmatrix} \mathbf{w}_k^i \tag{5.20}$$

where $\Omega = 0.1$ is the turn rate of the model.[1]

The observations are taken from a single sensor located at the origin of the referenced coordinated system and the measurements are available at discrete time intervals of 1 second.

The target generated measurement from j th target consists of target range and bearing

$$z_k = \begin{bmatrix} \tan^{-1}\left(\frac{y_k^j - y_S}{x_k^j - x_S}\right) \\ \sqrt{(y_k^j - y_S)^2 + (x_k^j - x_S)^2} \end{bmatrix} + v_k^j \tag{5.21}$$

where v_k^j is additive zero-mean Gaussian white noise with covariance diag[$9\,\mathrm{m}^2, 1 \times 10^{-3}\,\mathrm{rad}^2$].

The average false alarm rate $\lambda_k = 1 \times 10^{-4}$, the volume of observation $= 1000\,\mathrm{m} \times 2\pi\,\mathrm{rad}$, the false alarms are distributed uniformly within the observation region and the probability of target detection is taken as $p_D(\mathbf{x}_k) = 1$, regardless of the target state $\mathbf{x}_k$.

[1]The simulations are being updated to estimate the time-varying turn rate online.

The Markovian model transition probability matrix in MMPHD filter is taken as

$$[h_{pq}] = \begin{bmatrix} 0.95 & 0.05 \\ 0.05 & 0.95 \end{bmatrix} \tag{5.22}$$

The initial model probabilities are taken equal for both models as 0.5. In all the tracking algorithms, 800 sample points are used to represent one target.

5.5.1.1 Simulation Results

Figures 5.1, 5.2 and 5.3 show the true and estimated target trajectories for MMPHD, constant velocity PHD and coordinated turn PHD filters, respectively. The results are based on a single run. It can be seen that MMPHD filter gives better target position estimation results over the single model PHD filters. Further, it can be noted that there is a track loss in coordinated turn model due to the model mismatch at a particular instant. Track loss was observed in almost all runs.

Figures 5.4 and 5.5 illustrate the model switching property of the MMPHD filter for targets 1 and 2, respectively. The number of particles in each model is plotted against the time. The particles considered here are of equal weight and the proportion of the particles in each model of a particular target gives the approximate posterior model probabilities of that target.

Figure 5.6, 5.7, 5.8 and 5.9 show the root mean squared errors (RMSE) of position and velocity estimation for target 1 and 2, respectively. It can be seen that, in both position and velocity estimation, for about initial 40 time steps the performances of the single model PHD filter are somewhat better than the MMPHD filter. However,

the performance of both single model PHD filters degrade due to their poor adaptation to target maneuvers. Table 5.1 gives the average values of the position and velocity RMSE values. It can be seen that the MMPHD filter results in significant improvements in both target position and velocity estimates on both targets.

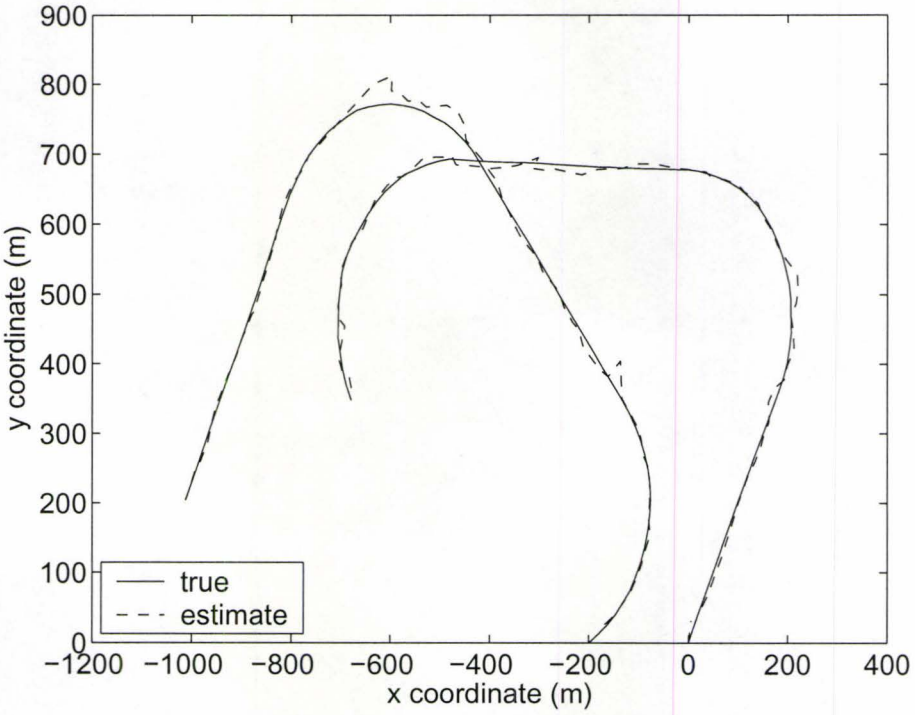


Figure 5.1: Target truth and multiple model PHD estimates

5.5.2 Bearings-only Sensors

The problem of bearings-only tracking arises in many practical applications such as submarine tracking or aircraft surveillance using a radar in passive mode [50,81]. This section presents a two-dimensional tracking example to compare the MMPHD filter's performance with that of the single-model PHD filters, namely, a constant-velocity

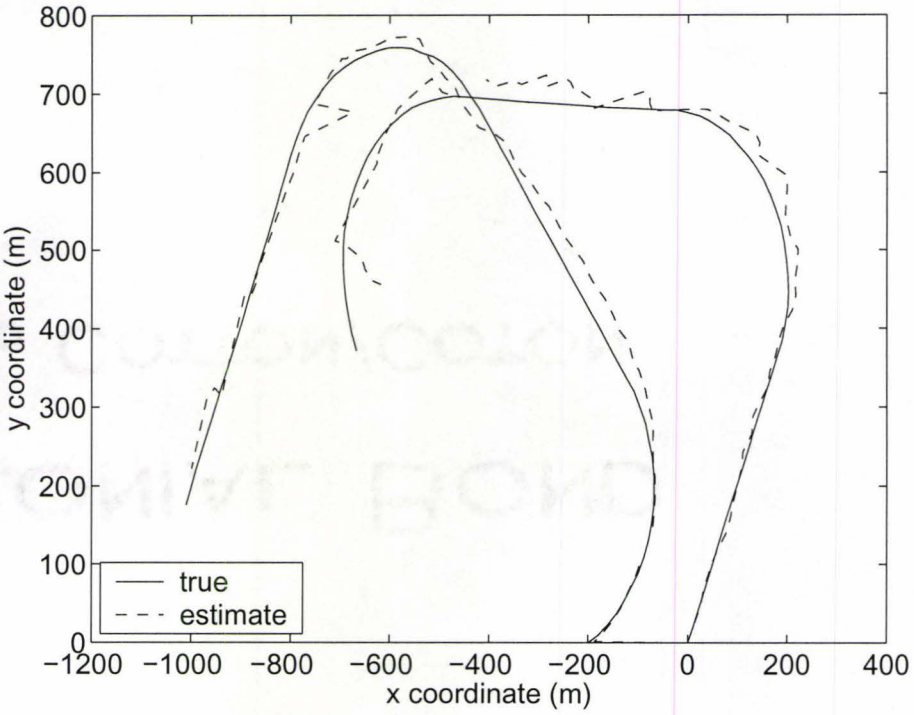


Figure 5.2: Target truth and constant velocity PHD estimates

model PHD filter and a coordinated-turn model PHD filter. The test scenario is constructed as follows.

1. The observations are taken from four fixed bearing-only sensors located at $(0, 0)$ m, $(0, 1 \times 10^4)$ m, $(1 \times 10^4, 0)$ m, and $(1 \times 10^4, 1 \times 10^4)$ m. The measurements are available at discrete-time sampling interval $T = 60$ seconds. The target-generated measurements corresponding to target j on sensor i are given by

$$z_k^{i,j} = \tan^{-1}\left(\frac{y_k^j - y_S^i}{x_k^j - x_S^i}\right) + v_k^{i,j}, \tag{5.23}$$

where $v_k^{i,j}$ is an i.i.d. sequence of zero-mean Gaussian variables with standard deviation 0.01 rad. (x_k^j, y_k^j) and (x_S^i, y_S^i) denote the locations of target j and

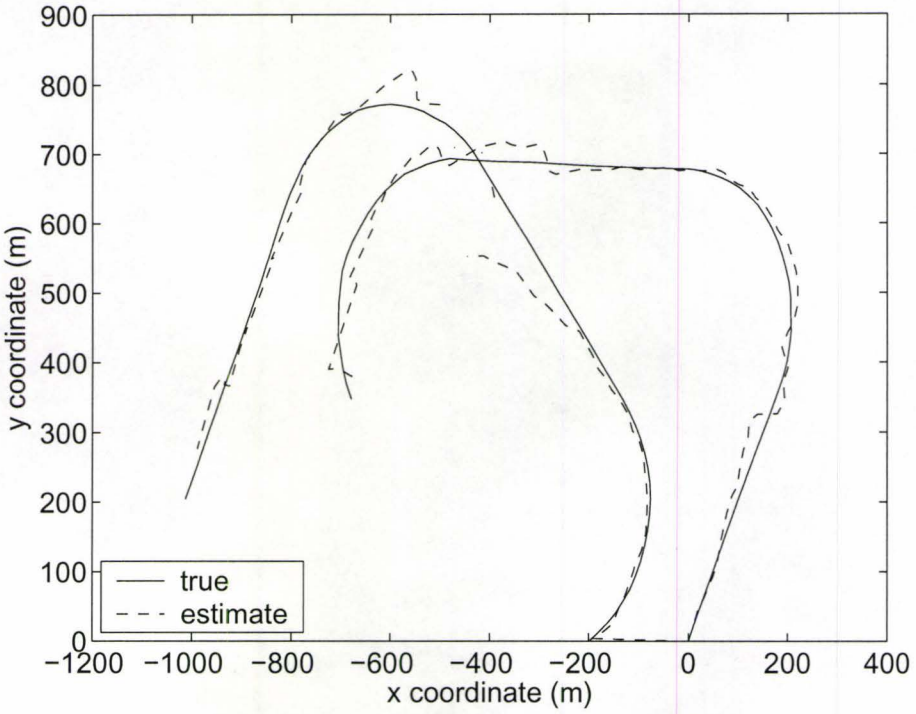


Figure 5.3: Target truth and coordinated turn PHD estimates

sensor i at time step k, respectively. The average false alarm rate $\lambda_k = 4 \times 10^{-3}$ rad^{-1}, the volume of observation of each sensor $= 2\pi$ rad, the false alarms are distributed uniformly within the sensor field of view, and the probability of target detection $p_D(\mathbf{x}_k) = 1$, regardless of the target state $\mathbf{x}_k$.

2. Two maneuvering targets, namely, target 1 and target 2, are travelling from initial target positions $(-3 \times 10^3,\ 5 \times 10^3)$ m and $(1.4 \times 10^4,\ 8 \times 10^3)$ m, respectively. This simulation does not include any spawning of new targets from existing ones. Target 1 moves eastward for 20 minutes at a nearly constant velocity with an initial velocity of 5 ms^{-1}, before executing a 9^0/minute (which amounts to an acceleration of about 0.1 ms^{-2}) coordinated turn [13] in the

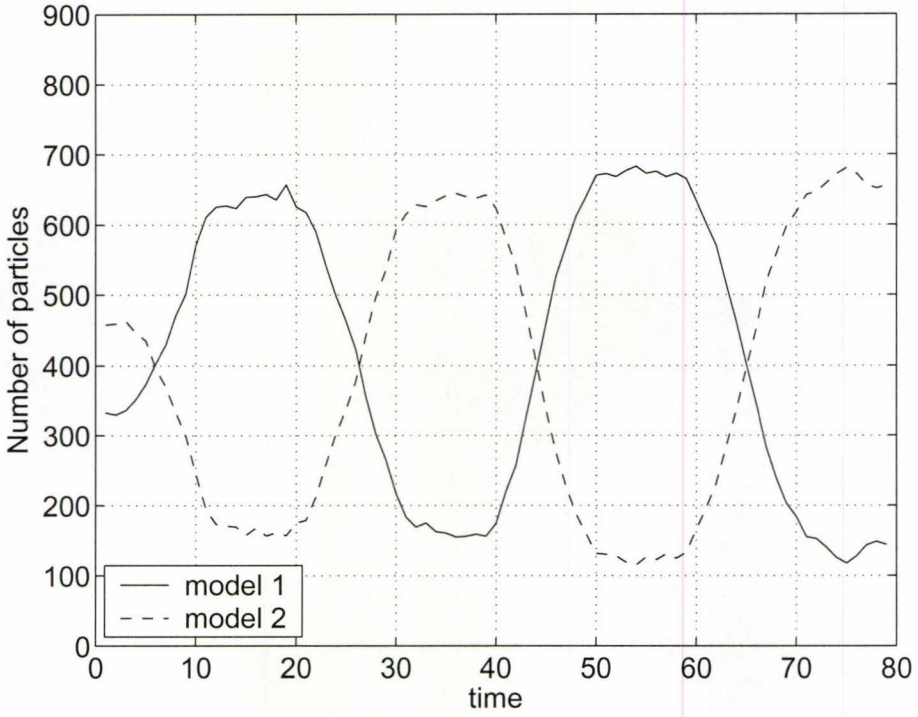


Figure 5.4: Model switching in the MMPHD filter – Target 1

anti-clockwise direction for 10 minutes. Then it moves northward for another 20 minutes, followed by a clockwise 9^0/minute coordinated turn for 10 minutes. Target 2 moves southward for 10 minutes at a nearly constant velocity with an initial velocity of 5 ms^{-1}, before executing a coordinated turn of 9^0/minute in the clockwise direction for 10 minutes. Then it moves westward at nearly a constant velocity for 30 minutes followed by an anti-clockwise coordinated turn of 9^0/minutes for 10 minutes. The target trajectories are shown in figure 5.10. This scenario leads to a maneuvering index[2] of more than 1.

3. The single target Markov transition equation that characterizes the constant

[2]The measurement contribution part of the Posterior Cramér-Rao Lower Bound is used to obtain the measurement standard deviation in calculating the maneuvering index.

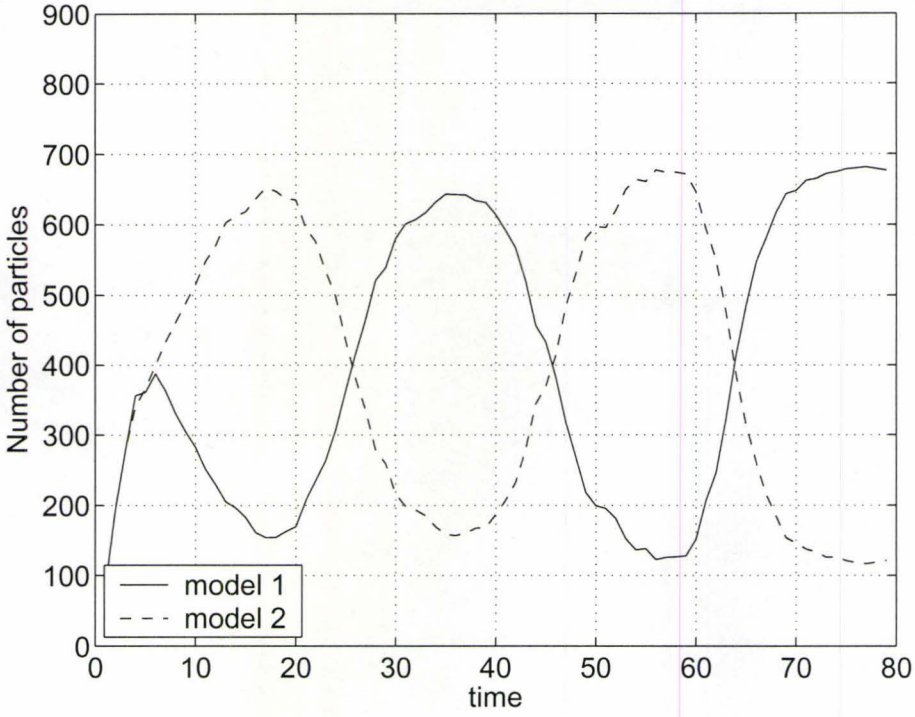


Figure 5.5: Model switching in the MMPHD filter – Target 2

velocity target dynamics ($r_k = 1$) is given by

$$\mathbf{x}_k^j = A_{1,k}^j \mathbf{x}_{k-1}^j + \mathbf{w}_{1,k}^j, \tag{5.24}$$

where $\mathbf{x}_k^j = [x_k^j, \dot{x}_k^j, y_k^j, \dot{y}_k^j]$ is the state of the j th target, which consists of target position (x_k^j, y_k^j) and target velocity $(\dot{x}_k^j, \dot{y}_k^j)$ at time step k, and $\mathbf{w}_{1,k}^j$ is an i.i.d sequence of zero-mean Gaussian vectors with covariance Σ_k^j. The matrices $A_{1,k}^j$

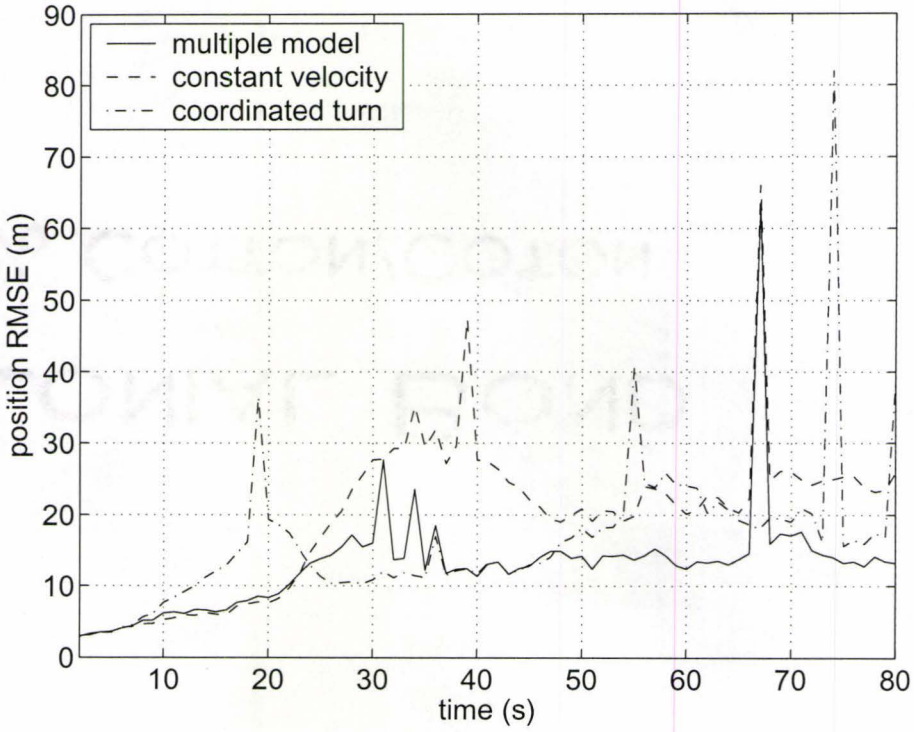


Figure 5.6: Position RMSE of target 1 estimates

and $\Sigma_{1,k}^{j}$ are given as follows

$$A_{1,k}^{j} = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.25}$$

$$\Sigma_{1,k}^{j} = \begin{bmatrix} \frac{T^3}{3} & \frac{T^2}{2} & 0 & 0 \\ \frac{T^2}{2} & T & 0 & 0 \\ 0 & 0 & \frac{T^3}{3} & \frac{T^2}{2} \\ 0 & 0 & \frac{T^2}{2} & T \end{bmatrix} l \tag{5.26}$$

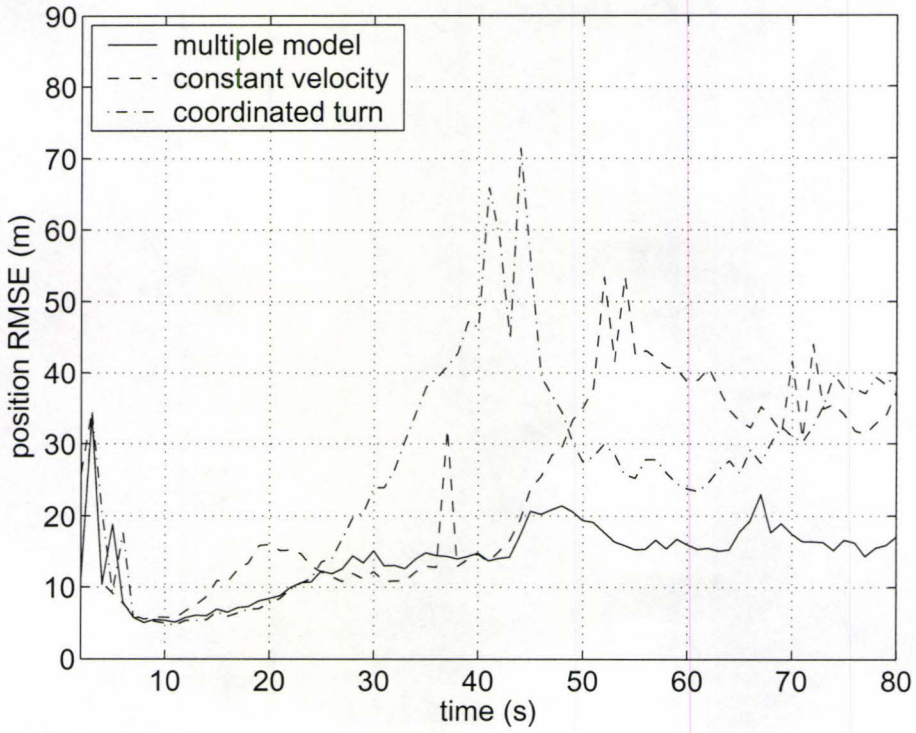


Figure 5.7: Position RMSE of target 2 estimates

where $l = 1 \times 10^{-4}$ $\mathrm{m^2 s^{-3}}$ is the level of the power spectral density of the corresponding continuous process noise.

4. The single-target Markov transition equation that characterizes the variable rate coordinated turn target dynamics ($r_k = 2$) is given by

$$\mathbf{x}_k^j = A_{2,k}^j \mathbf{x}_{k-1}^j + \mathbf{w}_{2,k}^j, \tag{5.27}$$

where $\mathbf{x}_k^j = [x_k^j, \dot{x}_k^j, y_k^j, \dot{y}_k^j, \Omega_k^j]$ is the augmented state vector of the j th target, which consists of target position (x_k^j, y_k^j), target velocity $(\dot{x}_k^j, \dot{y}_k^j)$ and turn rate Ω_k^j at time step k, and $\mathbf{w}_{2,k}^j$ is an i.i.d sequence of zero-mean Gaussian vectors

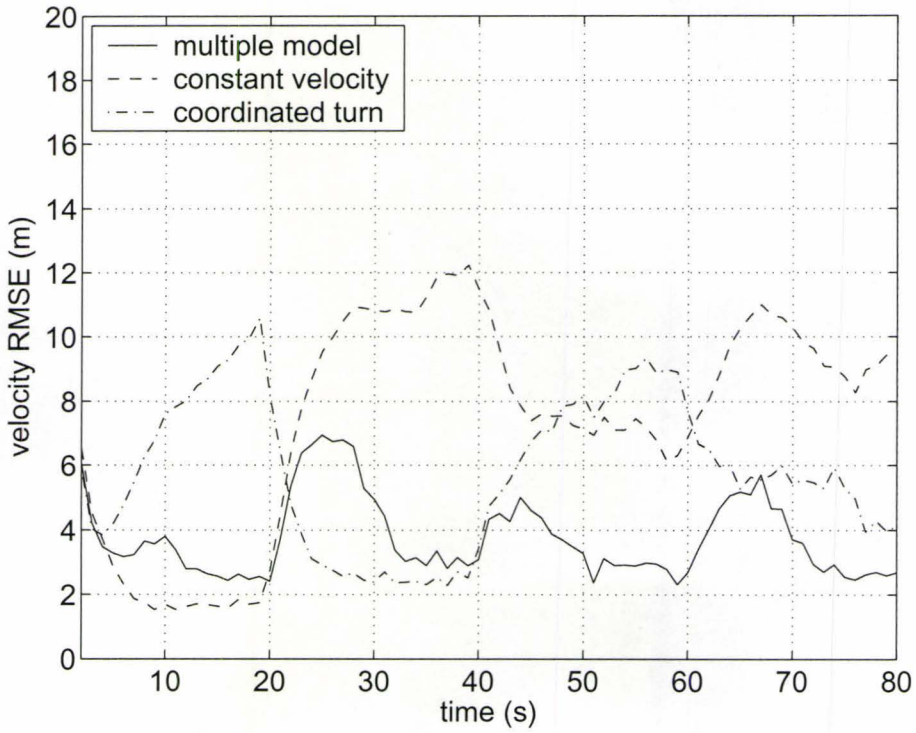


Figure 5.8: Velocity RMSE of target 1 estimates

with covariance $\Sigma_{2,k}^{j}$. The matrices $A_{2,k}^{j}$ and $\Sigma_{2,k}^{j}$ are given as follows

$$A_{2,k}^{j} = \begin{bmatrix} 1 & \frac{\sin \Omega_{k-1}^{j} T}{\Omega_{k-1}^{j}} & 0 & -\frac{1-\cos \Omega_{k-1}^{j} T}{\Omega_{k-1}^{j}} & 0 \\ 0 & \cos \Omega_{k-1}^{j} T & 0 & -\sin \Omega_{k-1}^{j} T & 0 \\ 0 & \frac{1-\cos \Omega_{k-1}^{j} T}{\Omega_{k-1}^{j}} & 1 & \frac{\sin \Omega_{k-1}^{j} T}{\Omega_{k-1}^{j}} & 0 \\ 0 & \sin \Omega_{k-1}^{j} T & 0 & \cos \Omega_{k-1}^{j} T & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.28}$$

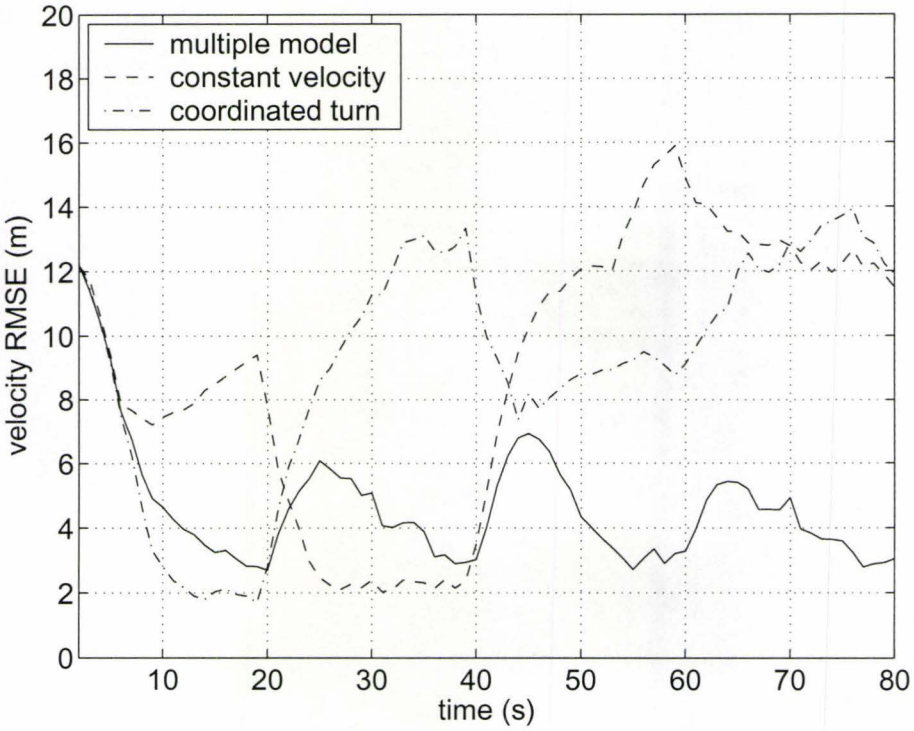


Figure 5.9: Velocity RMSE of target 2 estimates

$$\Sigma_{2,k}^{j} = \begin{bmatrix} \frac{T^3 l_1}{3} & \frac{T^2 l_1}{2} & 0 & 0 & 0 \\ \frac{T^2 l_1}{2} & T l_1 & 0 & 0 & 0 \\ 0 & 0 & \frac{T^3 l_1}{3} & \frac{T^2 l_1}{2} & 0 \\ 0 & 0 & \frac{T^2 l_1}{2} & T l_1 & 0 \\ 0 & 0 & 0 & 0 & T l_2 \end{bmatrix} \tag{5.29}$$

where the levels of the power spectral densities are $l_1 = 1 \times 10^{-4}$ $\mathrm{m^2 s^{-3}}$ and $l_2 = 1 \times 10^{-10}$ $\mathrm{rad^2 s^{-3}}$. Since the turn rate Ω_k^j is unknown, it is included in the state vector and must be estimated.

5. The Markovian model transition probability matrix in the MMPHD filter is

	Multiple model PHD	Constant velocity PHD	Coordinated turn PHD	units
Target 1 (position)	12.83	19.60	16.22	m
Target 2 (position)	14.08	23.68	26.24	m
Target 1 (velocity)	3.75	7.45	5.80	ms^{-1}
Target 2 (velocity)	4.64	9.00	9.10	ms^{-1}

Table 5.1: Average target state estimation RMSE (based on 100 Monte Carlo runs)

taken as

$$[h_{pq}] = \begin{bmatrix} 1 - \frac{T}{\tau_1} & \frac{T}{\tau_1} \\ \frac{T}{\tau_2} & 1 - \frac{T}{\tau_2} \end{bmatrix} \tag{5.30}$$

with sojourn times [13] $\tau_1 = 20$ minutes and $\tau_2 = 10$ minutes. The initial model probabilities for both models are 0.5.

6. We assume that the probability of target survival = 0.99, the probability of target spawning = 0, and the probability of spontaneous target birth = 0.01. The birth model that describes the entry of new targets is taken as follows.

   The x and y position elements of the state vector of new born particles are normally distributed centered around every two intersecting measurements with a standard deviation of 200 m. This one point initialization is done to all intersecting measurement pairs within the surveillance region, which is a square region defined by $(-5\times10^3, -5\times10^3)$ m and $(1.5\times10^4, 1.5\times10^4)$ m as the lower left and upper right corners, respectively. The velocity and turn rate elements of the state vector for the new born particles are distributed uniformly within the interval $[-5, 5]$ ms^{-1} and $[-0.02, 0.02]$ rads^{-1} respectively.

7. 1000 particles are used to represent one target and $J_k = (10 \times$ number of intersecting measurement points within the surveillance region) particles are used to

represent new spontaneously born targets in all the tracking algorithms. Before simulations begin, all PHD filters are initialized with 1000 particles normally distributed with mean $[-3 \times 10^3$ m, 5 ms^{-1}, 5×10^3 m, 0 ms^{-1}, 0 rads$^{-1}]$ and covariance diag$[4 \times 10^4$ m^2, 1 m^{2}s^{-2}, 4×10^4 m^2, 1 m^{2}s^{-2}, 4×10^{-4} rad^{2}s$^{-2}]$ representing target 1, and the model index parameters of the initial particles are taken as 1, i.e., the constant-velocity model, in the MMPHD filter.

8. The PHD filter does not provide a mechanism to get the target state estimates directly. One way of getting the estimates is by identifying the local maxima of the density. An expectation maximization based peak extraction algorithm to obtain the target state estimates can be found in [97]. For simplicity and to avoid added computational load we use the K-means clustering algorithm [49] to associate particles to targets[3] based on the position elements of the particle state vectors. The targets are associated to tracks using nearest-neighbor approach based on the mean of each target cluster.

Although we have assumed Gaussian process noise in this particular example, the MMPHD algorithm presented in Section 5.2 is able to deal with any process noise as long as the samples can be drawn from the process noise distribution. Since the PHD recursion given in this chapter is valid only for single sensor, The measurements are grouped from each sensor and used to update the filter iteratively.

5.5.2.1 Simulation Results

Figures 5.11 and 5.12 illustrate the model-switching property of the MMPHD filter by plotting number of particles in each model against time. The particles considered here

[3]We assume that the targets are sufficiently far apart that a particle represents only one target at any particular time.

are of equal weight, so the proportion of the particles in each model of a particular target gives the approximate posterior model probabilities of that target. Figures 5.11 and 5.12 also illustrate the number of targets estimated from the PHD filter. At the beginning of the scenario, all PHD filters assume that there is only one track, i.e. target 1, within the surveillance region and the particles corresponding to target 1 in the MMPHD filter are shown in Figure 5.11. However, it can be seen in Figure 5.12 that the total number of particles corresponding to target 2 increases with time at the beginning until it reaches nearly 1000, which is the number of particles assigned to represent one target. Once the track is formed, the MMPHD filter clearly displays the model switching property for target 2 and assigns more particles to the model that matches with the target motion.

Figures 5.13, 5.14, 5.15 and 5.16 show the root mean squared errors (RMSE) of position and velocity estimates for target 1 and 2, respectively. All figures indicate that the overall performance of the MMPHD filter is significantly better than that of both single-model PHD filters. Further, the single-model PHD filters show poor adaptation to target maneuvers and yield larger estimation errors. Figures 5.17 and 5.18 compare the RMSEs of turn rate estimates of the MMPHD filter and the single-model PHD filter that uses the coordinated-turn model for target 1 and 2, respectively. The single-model PHD filter that uses the constant-velocity model is not considered since there is no turn rate estimate in that filter. Table 5.2 gives the average of the estimation errors; note the values for the MMPHD filter show significant improvements over the single-model PHD filters.

The average computational times (based on 100 Monte Carlo runs) on dual 2.4 GHz Intel Xeon processors for a tracking period of 60 time steps are 205, 201 and 199

seconds with standard deviation 2.6, 1.7 and 2.5 seconds for multiple model, constant-velocity model, and coordinated-turn model PHD filter algorithms, respectively. All filters were implemented in MATLAB. Thus, if same number of sample points are used, the MMPHD filter is only marginally more expensive compared with single-model PHD filters since the only additional calculation required in the MMPHD filter is to generate model index parameters in the mixing stage.

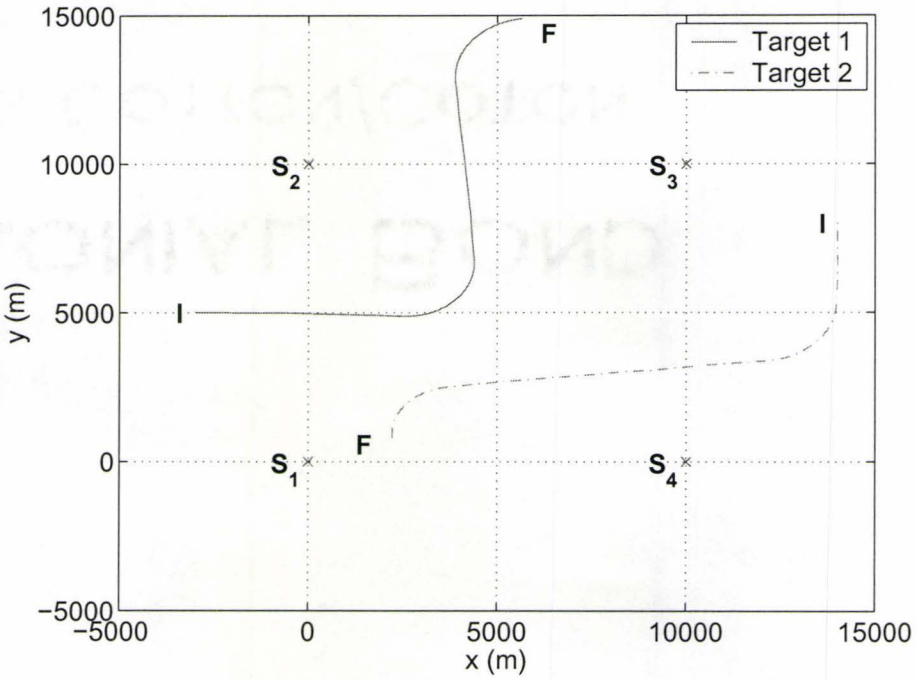


Figure 5.10: Target trajectory (I - initial, F - Final, S_i - Sensor)

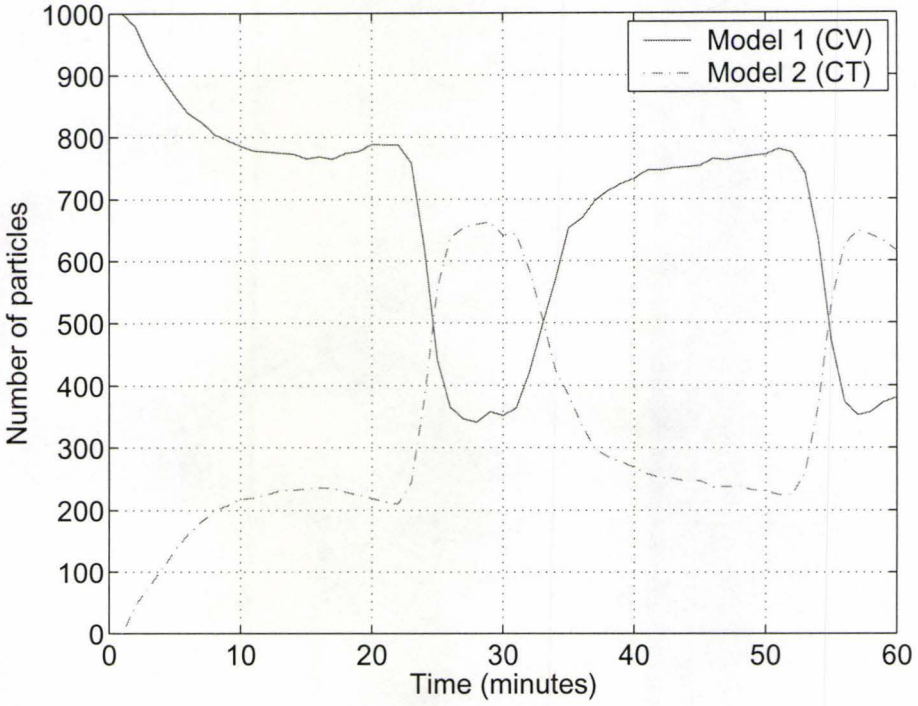


Figure 5.11: Model switching in target 1 in the MMPHD filter

Table 5.2: Average target state estimation RMSE (based on 100 Monte Carlo runs)

	Multiple model PHD	Constant velocity PHD	Coordinated turn PHD	units
Target 1 (position)	95.48	311.56	168.59	m
Target 2 (position)	210.89	401.88	237.09	m
Target 1 (velocity)	0.69	1.38	1.15	ms^{-1}
Target 2 (velocity)	1.17	1.57	1.29	ms^{-1}
Target 1 (turn rate)	0.00086	-	0.00121	$rads^{-1}$
Target 2 (turn rate)	0.00106	-	0.00110	$rads^{-1}$

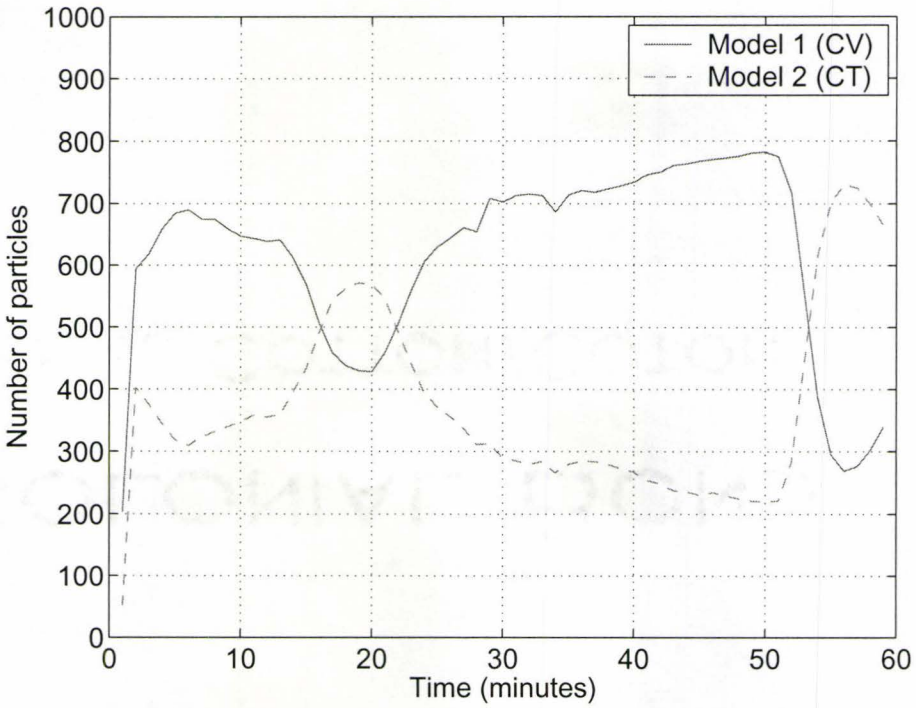


Figure 5.12: Model switching in target 2 in the MMPHD filter

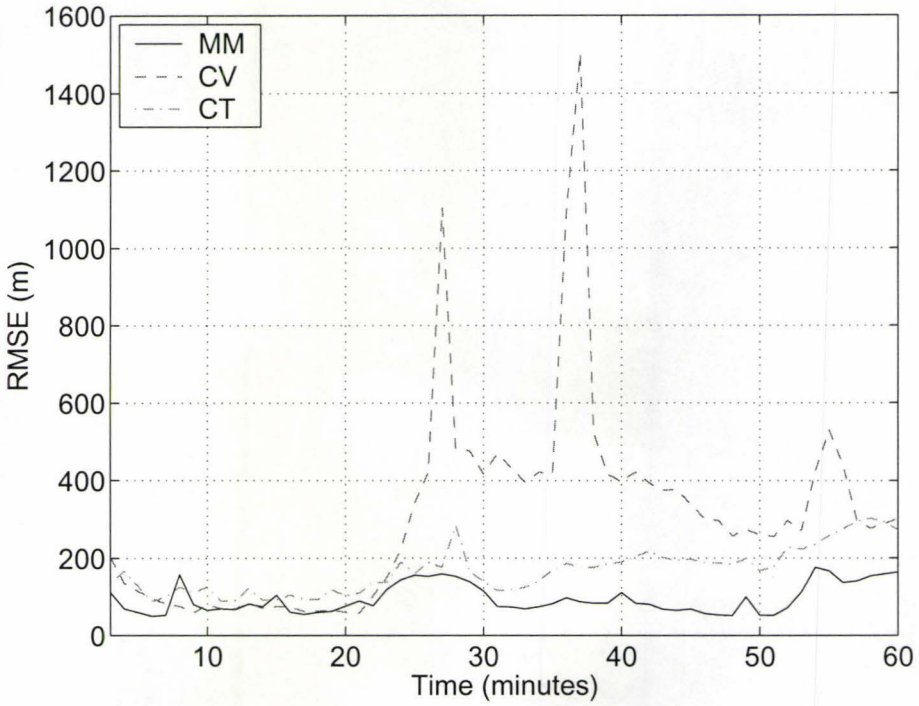


Figure 5.13: Target 1 – Position estimation RMSE

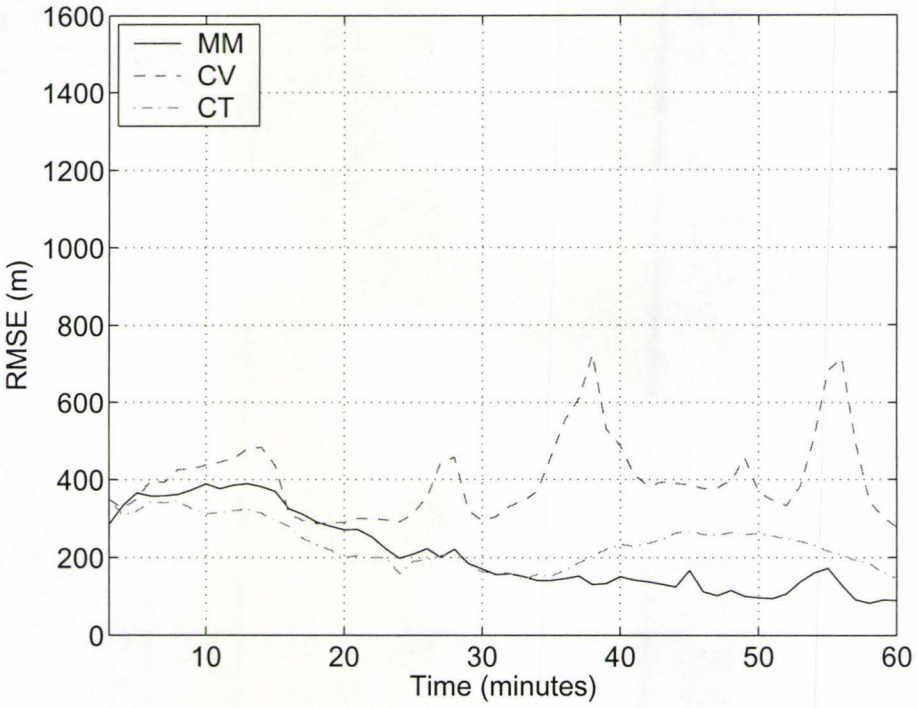


Figure 5.14: Target 2 – Position estimation RMSE

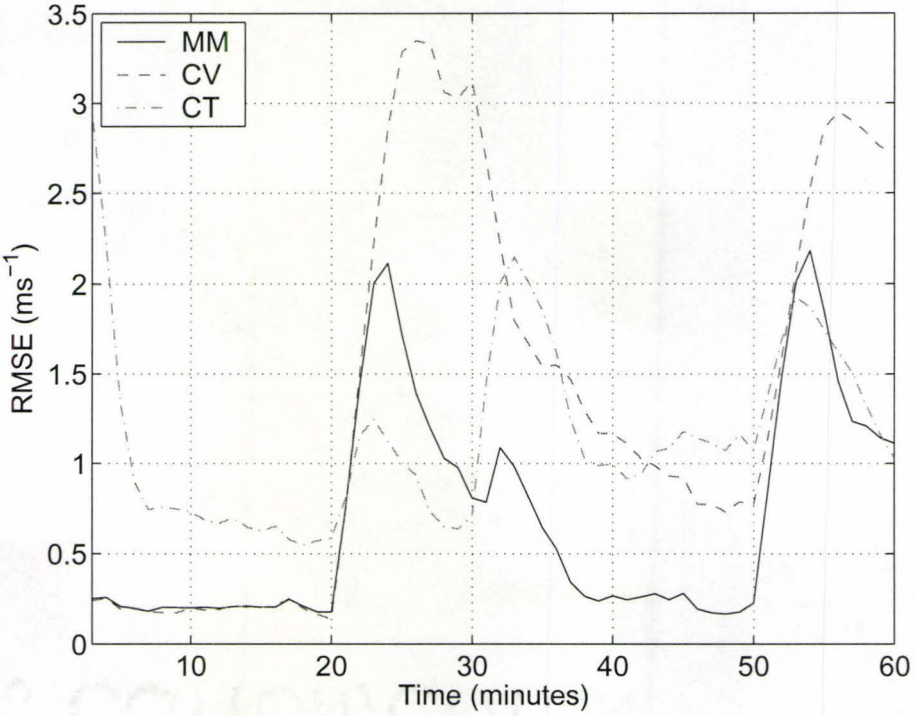


Figure 5.15: Target 1 – Velocity estimation RMSE

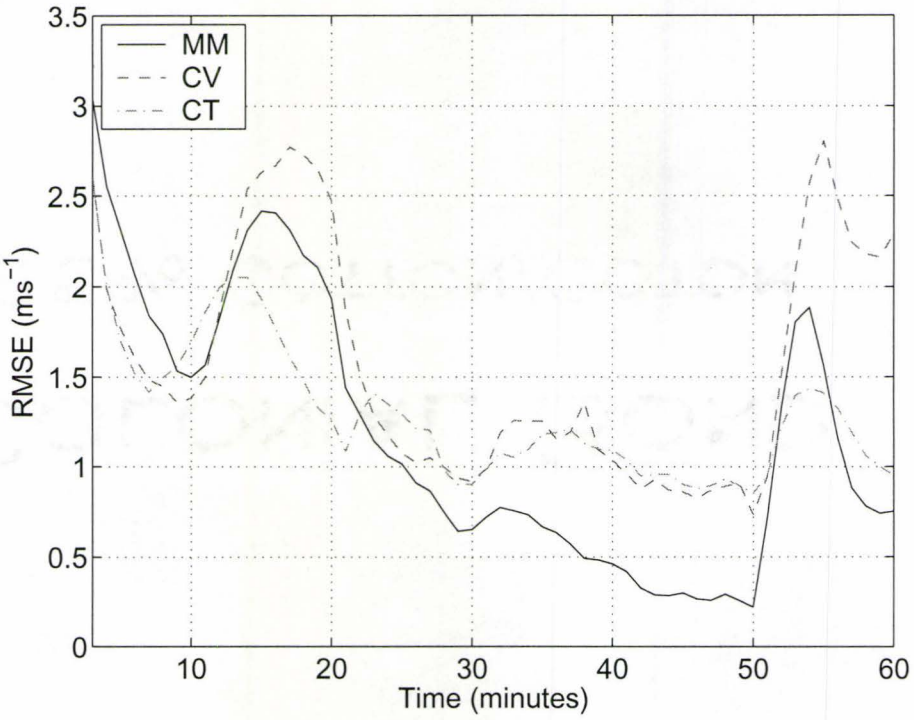


Figure 5.16: Target 2 – Velocity estimation RMSE

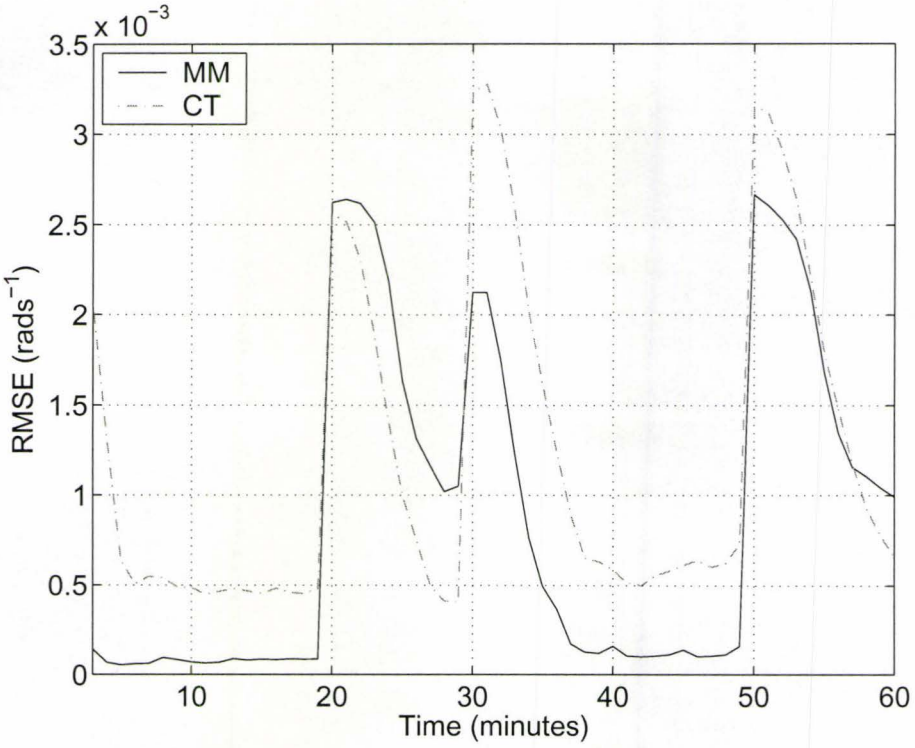


Figure 5.17: Target 1 – Turn rate estimation RMSE

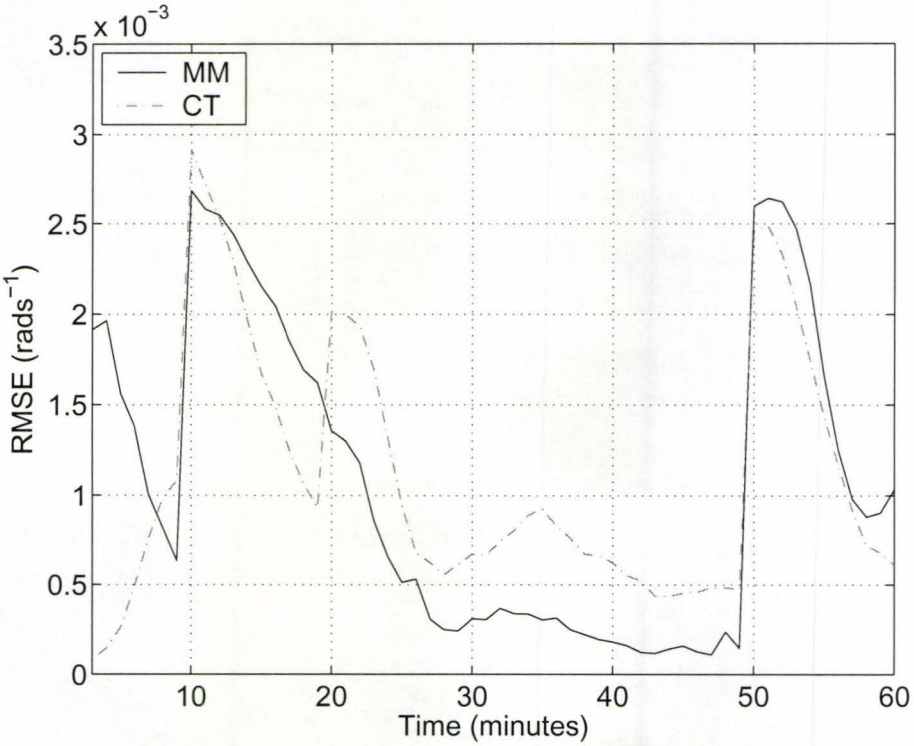


Figure 5.18: Target 2 – Turn rate estimation RMSE

Chapter 6

An SMC PHD Algorithm for Multitarget TBD

6.1 Introduction

In this chapter of the thesis, we consider the problem of tracking multiple time varying low SNR targets in a cluttered environment. The recent developments of stealth aircrafts and cruise missiles have emphasized the essence of tracking dim targets. we present a recursive track-before-detect (TBD) algorithm based on the Probability Hypothesis Density (PHD) filter to solve the above problem. TBD algorithms are better suited over standard target tracking methods for tracking dim targets in noise. In classical target tracking, where the measurements are pre-processed at each time step before passing them to the tracking filter results in information loss, which is very damaging if the target signal-to-noise ratio is low. However, in TBD approaches the tracking filter operates directly on the raw measurements at the expense of added computational burden. The development of a recursive TBD algorithm reduces the

computational burden over conventional TBD methods, namely, Hough transform, dynamic programming, etc. The TBD is a hard nonlinear non-Gaussian problem even for single target scenarios. Recent advances in Sequential Monte Carlo (SMC) based nonlinear filtering make multitarget TBD feasible. However, the current implementations use a modelling setup to accommodate the varying number of targets where a multiple model SMC based TBD approach is used to solve the problem conditioned on the model, i.e., number of targets. The PHD filter, which propagates only the first-order statistical moment (or the PHD) of the full target posterior, has been shown to be a computationally efficient solution to multitarget tracking problems with varying number of targets. We propose a PHD filter based TBD so that there is no assumption to be made on the number of targets. Simulation results are presented to show the effectiveness of the proposed filter in tracking multiple weak targets.

6.2 System Setup

6.2.1 Target Model

A general parameterized target dynamics of the t th target is given by

$$\mathbf{x}_{k+1}^t = \mathbf{f}_k(\mathbf{x}_k^t, \mathbf{v}_k) \qquad t = 1, \ldots, N_k \tag{6.1}$$

where $\mathbf{x}_k^t = [x_k^t \ \dot{x}_k^t \ y_k^t \ \dot{y}_k^t \ I_k^t]$ is the target state vector at time step k, N_k is the number of targets at time step k, $\mathbf{f}_k(\cdot)$ is a nonlinear function and $\mathbf{v}_k$ is the process noise of known statistics. Here (x_k^t, y_k^t), $(\dot{x}_k^t, \dot{y}_k^t)$ and I_k^t denote the position, velocity and the intensity of the target, respectively.

6.2.2 Sensor Model

The sensor provides two-dimensional images of the surveillance region at an interval of T and each image consists of $n \times m$ resolution cells. Each cell corresponds to a rectangular region of dimensions $\Delta_x \times \Delta_y$ and the center of each cell (i, j) is defined to be at $(i\Delta_x, j\Delta_y)$ for $i = 1, \ldots, n$ and $j = 1, \ldots, m$. The measured intensity $z_k^{(i,j)}$ at resolution cell (i, j) is given by

$$z_k^{(i,j)} = \begin{cases} \sum_{t=1}^{N_k} h_k^{(i,j)}(\mathbf{x}_k^t) + w_k^{(i,j)} & \mathcal{H}_1\text{: if there are } N_k \text{ targets present} \\ \\ w_k^{(i,j)} & \mathcal{H}_0\text{: if there are no targets present} \end{cases} \tag{6.2}$$

where $w_k^{(i,j)}$ is the measurement noise of known statistics. The noise is assumed to be independent from cell to cell and from image to image. The target contribution intensity $h_k^{(i,j)}(\mathbf{x}_k^t)$ can be either due to target extent or to the sensor point spread function. If we assume a point target, the target intensity will be distributed over the surrounding cells according to the sensor's point spread function. Thus, for a point target of state $\mathbf{x}_k^t$, the contribution to cell (i, j) is approximated as

$$h_k^{(i,j)}(\mathbf{x}_k^t) \approx \frac{\Delta_x \Delta_y I_k^t}{2\pi\Sigma^2} \exp\left\{-\frac{(i\Delta_x - x_k^t)^2 + (j\Delta_y - y_k^t)^2}{2\Sigma^2}\right\} \tag{6.3}$$

where Σ represents the amount of blurring introduced by the sensor.

The complete set of measurements at time step k is denoted by

$$\mathbf{z}_k = \left\{z_k^{(i,j)} : \; i = 1, \ldots, n, \; j = 1, \ldots, m\right\} \tag{6.4}$$

and the set of complete measurements collected up to time step k is denoted by

$$Z_{1:k} = \{\mathbf{z}_i : i = 1, \ldots, k\} \tag{6.5}$$

The likelihood function for the above sensor model is given as

$$p(\mathbf{z}_k|\mathbf{x}_k^1, \mathbf{x}_k^2, \ldots, \mathbf{x}_k^{N_k}, \mathcal{H}_h) = \begin{cases} \prod_{i=1}^{n}\prod_{j=1}^{m} p_{S+N}(z_k^{(i,j)}|\mathbf{x}_k^1, \mathbf{x}_k^2, \ldots, \mathbf{x}_k^{N_k}) & \text{under } \mathcal{H}_1 \\ \\ \prod_{i=1}^{n}\prod_{j=1}^{m} p_N(z_k^{(i,j)}) & \text{under } \mathcal{H}_0 \end{cases} \tag{6.6}$$

Here $p_N(z_k^{(i,j)})$ is the pdf of background noise in pixel (i, j), while $p_{S+N}(z_k^{(i,j)}|\mathbf{x}_k^1, \mathbf{x}_k^2, \ldots, \mathbf{x}_k^{N_k})$ is the likelihood of summation of individual target signal plus noise in pixel (i, j), given that the targets are in states $\mathbf{x}_k^1, \mathbf{x}_k^2, \ldots, \mathbf{x}_k^{N_k}$

The probability density functions can be expressed as

$$p_N(z_k^{(i,j)}|\mathcal{H}_0) = \mathcal{N}(z_k^{(i,j)}; 0, \sigma^2) \tag{6.7}$$

$$p_{S+N}(z_k^{(i,j)}|\mathcal{H}_1) = \mathcal{N}(z_k^{(i,j)}; \sum_{t=1}^{N_k} h_k^{(i,j)}(\mathbf{x}_k^t), \sigma^2) \tag{6.8}$$

Here we assume a zero-mean Gaussian background noise $w_k^{(i,j)}$ with standard deviation σ. However, the algorithm presented in this chapter is applicable to non-Gaussian measurement noise, as long as the statistics of $w_k^{(i,j)}$ are known for each cell (i, j).

Since the target will affect largely the cells in the vicinity of its location, we can

approximate $p(\mathbf{z}_k|\mathbf{x}_k^t)$ as

$$p(\mathbf{z}_k|\mathbf{x}_k^t, \mathcal{H}_1) \approx \prod_{i \in C_i(\mathbf{x}_k^t)} \prod_{j \in C_j(\mathbf{x}_k^t)} p_{S+N}(z_k^{(i,j)}|\mathbf{x}_k^t) \prod_{i \notin C_i(\mathbf{x}_k^t)} \prod_{j \notin C_j(\mathbf{x}_k^t)} p_N(z_k^{(i,j)}) \tag{6.9}$$

where $C_i(\mathbf{x}_k^t)$ and $C_j(\mathbf{x}_k^t)$ are the sets of subscripts i and j, respectively, corresponding to pixels affected by the target. Here we use the approximation in order to reduce the computational load of the sequential Monte Carlo PHD filter. $C_i(\mathbf{x}_k^n)$ is usually selected as $C_i(\mathbf{x}_k^n) = \{i_0 - q, \ldots, i_0 - 1, i_0, i_0 + 1, \ldots, i_0 + q\}$, where i_0 is the nearest integer value of the particle state vector component $x_k^n = \mathbf{x}_k^n[1]$ and q is a design parameter. $C_j(\mathbf{x}_k^n)$ is also selected by a similar procedure. Let's define the cell that has the most influence by the target.

$$D_r(\mathbf{x}_k^t) = \max_r h_k^{(r,s)}(\mathbf{x}_k^t) \tag{6.10}$$

$$D_s(\mathbf{x}_k^t) = \max_s h_k^{(r,s)}(\mathbf{x}_k^t) \tag{6.11}$$

6.3 PHD Filter Implementation

This section presents the implementation details of the proposed filter. The PHD filter has a property of estimating the number of targets and it is not necessary to assume the maximum possible number of targets within the surveillance region. However, the algorithm requires a suitably chosen models for target birth, spawning and disappearance.

We use an SMC approach to implement the algorithm which provides a mechanism to represent the posterior density by a set of random samples or particles. Each particle consists of a target state with an associated weight. The SMC implementation

considered here is structurally similar to Sampling Importance Resampling (SIR) type of particle filter [4].

Let the posterior $D_{k-1|k-1}(\mathbf{x}_{k-1}|Z_{1:k-1})$ be represented by a set of particles $\left\{w_{k-1}^{(p)}, \mathbf{x}_{k-1}^{(p)}\right\}_{p=1}^{L_{k-1}}$. That is,

$$D_{k-1|k-1}(\mathbf{x}_{k-1}, Z_{1:k-1}) = \sum_{p=1}^{L_{k-1}} w_{k-1}^{(p)} \delta(\mathbf{x}_{k-1} - \mathbf{x}_{k-1}^{(p)}) \tag{6.12}$$

6.3.1 Prediction

We apply importance sampling to generate samples that approximate the predicted density. Generate samples $\{\mathbf{x}_{k|k-1}^{(p)}\}_{p=1}^{L_{k-1}}$ from the proposal density $q_k(\cdot|\mathbf{x}_{k-1}, Z_k)$ and i.i.d. samples $\{\mathbf{x}_{k|k-1}^{(p)}\}_{p=L_{k-1}+1}^{L_{k-1}+J_k}$ corresponding to new spontaneously born targets from another proposal density $p_k(\cdot|Z_k)$.

$$\mathbf{x}_{k|k-1}^{(p)} \sim \begin{cases} q_k(\cdot|\mathbf{x}_{k-1}, Z_k) & p = 1, \ldots, L_{k-1} \\ \\ p_k(\cdot|Z_k) & p = L_{k-1}+1, \ldots, L_{k-1}+J_k \end{cases} \tag{6.13}$$

Then, the weighted approximation of the predicted density is given by

$$D_{k|k-1}(\mathbf{x}_{k|k-1}|Z_{1:k-1}) = \sum_{p=1}^{L_{k-1}+J_k} w_{k|k-1}^{(p)} \delta(\mathbf{x}_{k|k-1} - \mathbf{x}_{k|k-1}^{(p)}) \tag{6.14}$$

where

$$w_{k|k-1}^{(p)} = \begin{cases} \frac{e_{k|k-1}(\mathbf{x}_{k-1}^{(p)}) f_k(\mathbf{x}_{k|k-1}^{(p)}|\mathbf{x}_{k-1}^{(p)}) + b_{k|k-1}(\mathbf{x}_{k|k-1}^{(p)}|\mathbf{x}_{k-1}^{(p)})}{q_k(\mathbf{x}_{k|k-1}^{(p)}|\mathbf{x}_{k-1}^{(p)}, Z_k)} & p = 1, \ldots, L_{k-1} \\ \\ \frac{\gamma_k(\mathbf{x}_{k|k-1}^{(p)})}{p_k(\mathbf{x}_{k|k-1}^{(p)}|Z_k)} & p = L_{k-1}+1, \ldots, L_{k-1}+J_k \end{cases} \tag{6.15}$$

where $e_{k|k-1}(\mathbf{x}_{k-1})$ denotes the probability that a target with state $\mathbf{x}_{k-1}$ will survive at time step k and $b_{k|k-1}(\mathbf{x}_{k|k-1}^{(p)}|\mathbf{x}_{k-1}^{(p)})$ denote the PHD of spawned targets at time step k from a target with state $\mathbf{x}_{k-1}$. The PHD of the new born spontaneous targets at time step k is denoted by $\gamma_k(\mathbf{x}_{k|k-1}^{(p)})$. The samples for new born targets are drawn as follows. For target position components, the proposal density is uniform over the regions of the surveillance area for which $z_k^{(i,j)} > \gamma$, where γ is a predefined threshold value. For target velocity components, the density is uniform between $[-v_{max}, v_{max}]$ where v_{max} is the maximum target speed. For target intensity components, the density is uniform between $[I_{min}, I_{max}]$ where I_{min} and I_{max} are the minimum and maximum target intensity levels.

6.3.2 Update

With the availability of measurement image at time step k, the importance weights are computed using the likelihood ratio in cell (i, j) for a target in state $\mathbf{x}_k^n$ as follows.

$$l(z_k^{(i,j)}|\mathbf{x}_k^n) \triangleq \frac{p_{S+N}(z_k^{(i,j)}|\mathbf{x}_k^n)}{p_N(z_k^{(i,j)})} \tag{6.16}$$

$$= \exp\left\{-\frac{h_k^{(i,j)}(h_k^{(i,j)} - 2z_k^{(i,j)})}{2\sigma^2}\right\} \tag{6.17}$$

The updated particle weights can be calculated by

$$w_k^{*(p)} = \frac{\prod_{i\in C_i(\mathbf{x}_{k|k-1}^{(p)})} \prod_{j\in C_j(\mathbf{x}_{k|k-1}^{(p)})} l(\mathbf{z}_k^{(i,j)}|\mathbf{x}_{k|k-1}^{(p)})}{\lambda_k + \Psi_k(\mathbf{z}_k^{(r,s)})} \quad \text{for } r = D_r(\mathbf{x}_{k|k-1}^{(p)}),\ s = D_s(\mathbf{x}_{k|k-1}^{(p)}) \tag{6.18}$$

where λ_k is a normalization constant. $\Psi_k(z_k^{(r,s)})$ is given by

$$\Psi_k(z_k^{(r,s)}) = \sum_{p\in P^{r,s}} \prod_{i\in C_i(\mathbf{x}_{k|k-1}^{(p)})} \prod_{j\in C_j(\mathbf{x}_{k|k-1}^{(p)})} l(\mathbf{z}_k^{(i,j)}|\mathbf{x}_{k|k-1}^{(p)}) w_{k|k-1}^{(p)} \tag{6.19}$$

where the set of particles $P^{r,s}$ is given by

$$P^{r,s} = \left\{ p : p \in \{1,\ldots,L_{k-1}+J_k\};\ r = D_r(\mathbf{x}_{k|k-1}^{(p)}) \text{ and } s = D_s(\mathbf{x}_{k|k-1}^{(p)}) \right\} \tag{6.20}$$

6.3.3 Resample

To perform resampling, since the weights are not normalized to unity in PHD filters, the expected number of targets is calculated by summing up the total weights, i.e.,

$$\hat{n}_k^X = \sum_{p=1}^{L_{k-1}+J_k} w_k^{*(p)} \tag{6.21}$$

Then, the updated particle set $\left\{w_k^{*(p)}/n_k^X, \mathbf{x}_{k|k-1}^{(p)}\right\}_{p=1}^{L_{k-1}+J_k}$ is resampled to get $\left\{w_k^{(p)}/n_k^X, \mathbf{x}_k^{(p)}\right\}_{p=1}^{L_k}$ such that the total weight after resampling remains n_k^X. Now, the discrete approximation of the updated posterior density at time step k is given by

$$D_{k|k}(\mathbf{x}_k|Z_{1:k}) = \sum_{p=1}^{L_k} w_k^{(p)} \delta(\mathbf{x}_k - \mathbf{x}_k^{(p)}) \tag{6.22}$$

6.4 Simulations

In this section, a two dimensional tracking example is presented to show the efficacy of the proposed track-before-detect algorithm. The simulation generates random synthetic target trajectories with

- a nearly constant velocity model for target motion.
- a random walk model for target intensity.

Then the dynamic model equation for target dynamics is given by

$$\mathbf{x}_{k+1} = \mathbf{F}\mathbf{x}_k + \mathbf{v}_k \tag{6.23}$$

where

$$\mathbf{F} = \begin{bmatrix} 1 & T & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & T & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{6.24}$$

and $\mathbf{v}_k$ is zero-mean white Gaussian noise with covariance $\mathbf{Q}$ given by

$$\mathbf{Q} = \begin{bmatrix} \frac{q_1}{3}T^3 & \frac{q_1}{2}T^2 & 0 & 0 & 0 \\ \frac{q_1}{2}T^2 & q_1 T & 0 & 0 & 0 \\ 0 & 0 & \frac{q_1}{3}T^3 & \frac{q_1}{2}T^2 & 0 \\ 0 & 0 & \frac{q_1}{2}T^2 & q_1 T & 0 \\ 0 & 0 & 0 & 0 & q_2 T \end{bmatrix} \tag{6.25}$$

where $q_1 = 0.001$ and $q_2 = 0.01$ denote the level of process noise in target motion and intensity, respectively. The adopted target model accommodates deviations from the straight-line motion as well as fluctuations in target intensity.

A sequence of 80 frames of sensor data has been generated with a time interval $T = 1$ s. The other parameter values are $\Delta_x = \Delta_y = 1$, $n = m = 20$, $\sigma = 1$ and $\Sigma = 0.7$.

Target 1 enters the measurement region at time $T = 7$ s with initial target state [4.2 0.15 7.2 0.10 30]. Target 2 and target 3 enter the region at times $T = 20$ s and $T = 40$ s with initial target states [8.2 0.10 0 0.10 20] and [0 0.10 8.2 0.10 20], respectively. The SNR is given by

$$\text{SNR} = 10 \log \left[\frac{I \Delta_x \Delta_y / 2\pi\Sigma^2}{\sigma} \right]^2 \tag{6.26}$$

Then, the initial SNR values are 10.23 dB, 6.71 dB and 6.71 dB for target 1, target 2 and target 3, respectively. All the targets exist within the surveillance region until the last time step. This simulation does not include any spawning new targets from existing ones.

The filter parameters are taken as follows: The number of particles representing one target $N = 3000$, number of particles representing new born targets $J_k = 40000$, the probability of target survival = 0.99, the probability of target spawning = 0, the probability of spontaneous target birth = 0.01, threshold $\gamma = 2$, $v_{max} = 1$ unit/s, initial intensity range from $I_{min} = 10$ to $I_{max} = 40$, $q = 2$ and $\lambda_k = 2 \times 10^{-5}$.

Figures 6.1, 6.2, 6.3 and 6.4 display four synthesized measurement data sequence at times 1, 7, 15 and 50 seconds, respectively. The cell intensity is shown in grey linear scale with white color indicating the highest intensity. These images are presented to illustrate that by visual inspection it is impossible to detect the existence and infer the locations of the targets with lower SNR.

Figure 6.5 displays the number of target estimate based on a single run. Here the number of targets estimate is calculated by dividing the total particles by the number of particles representing one target, since the particles are of equal weights after the resampling step. The figure shows that targets with lower SNR, i.e., target 2 and target 3, take more time steps to get detected by the algorithm.

Figures 6.6, 6.7 and 6.8 show the target position samples at times 19, 39 and 75 seconds, respectively. The cross indicates current true target position. The initial samples are distributed uniformly within cells that have intensity levels above a pre-defined threshold value. Eventually, several time steps after targets have appeared, particles begin to cluster around the target.

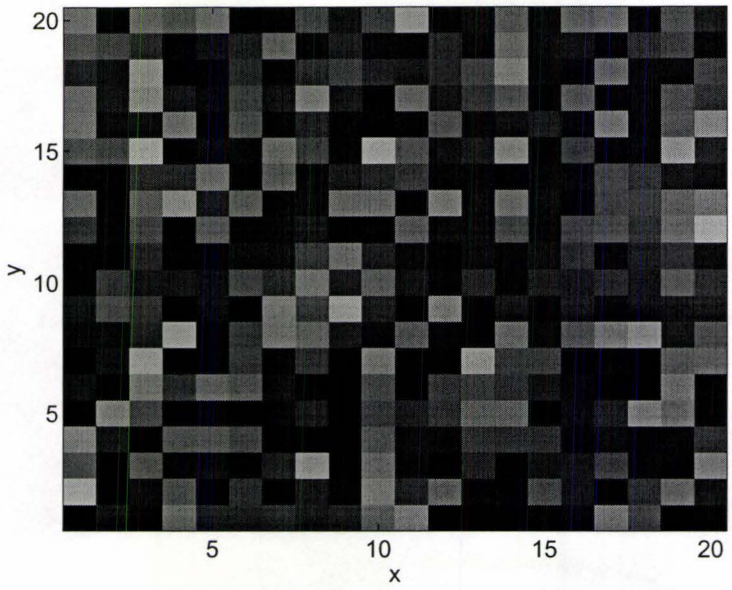


Figure 6.1: Simulated sensor data – Frame 1

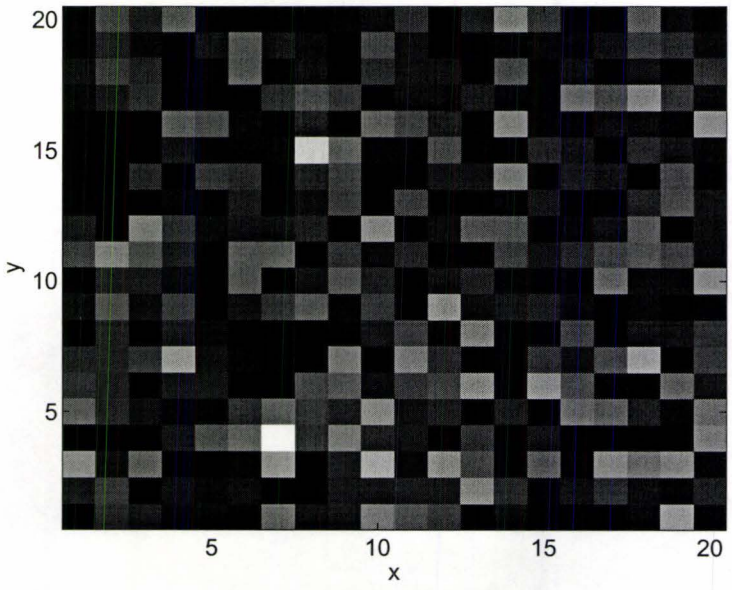


Figure 6.2: Simulated sensor data – Frame 7

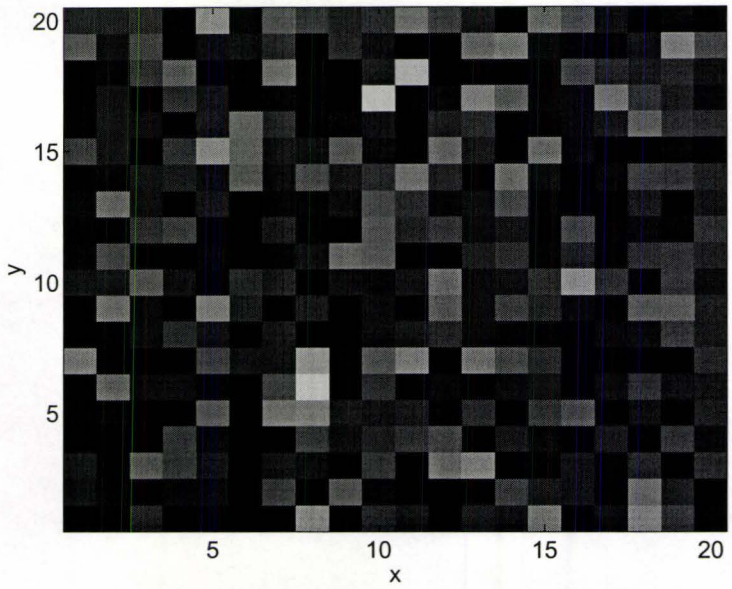


Figure 6.3: Simulated sensor data – Frame 15

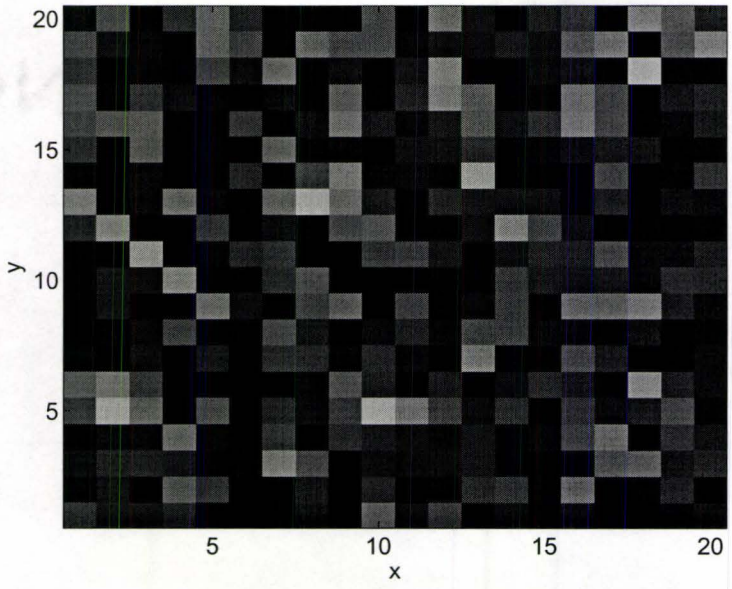


Figure 6.4: Simulated sensor data – Frame 50

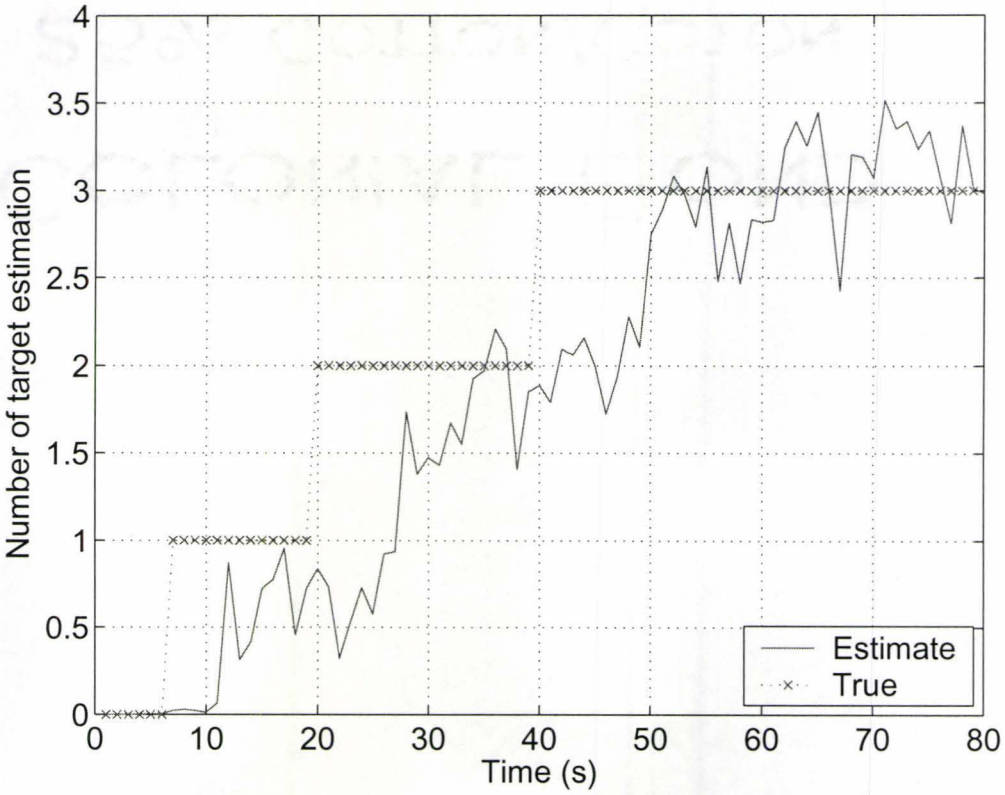


Figure 6.5: Number of targets estimate (single run)

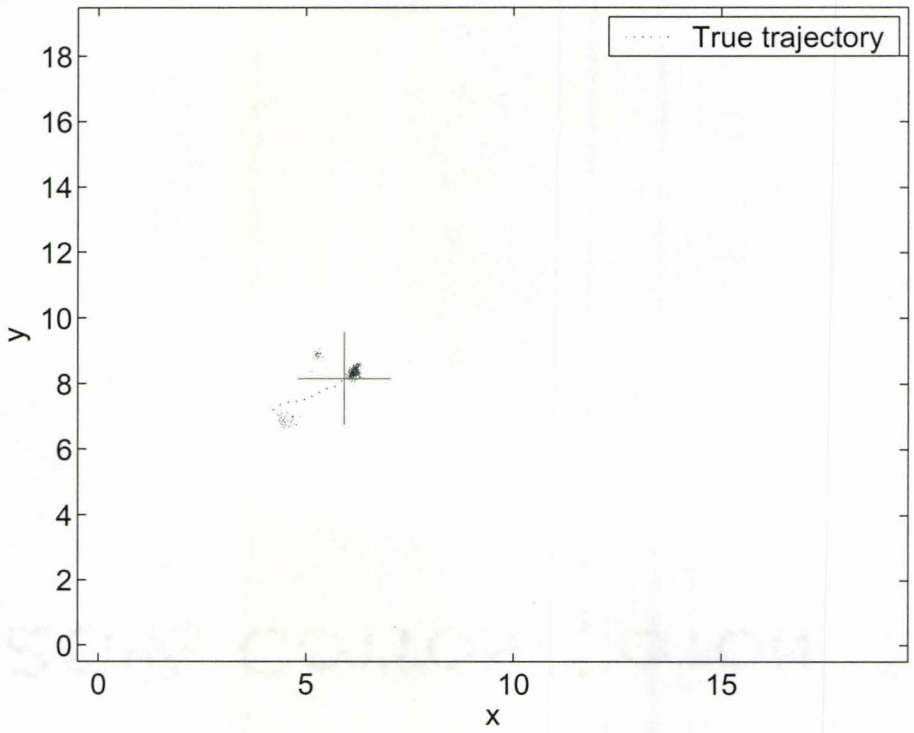


Figure 6.6: Target position particles at 19 s

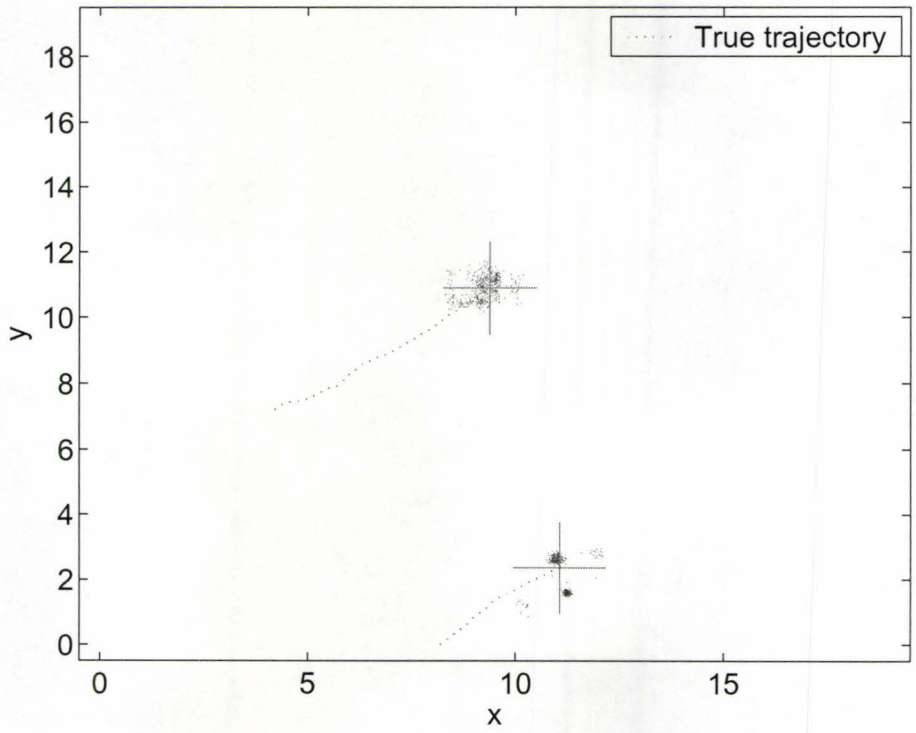


Figure 6.7: Target position particles at 39 s

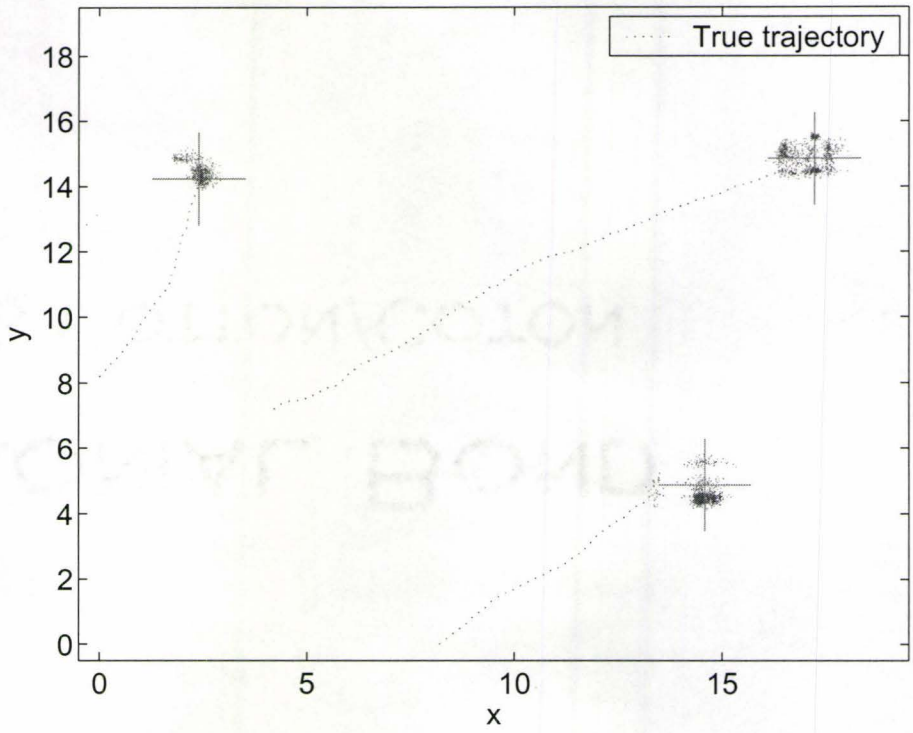


Figure 6.8: Target position particles at 75 s

Chapter 7

Distributed Implementation of SMC PHD Filter

7.1 Introduction

A sensor network consists of a collection of nodes that can be deployed in large numbers over the monitored area. The nodes are interconnected using wireless communication with the ability to detect or monitor some aspect of an environment. Each node in the network is self-powered and usually equipped with a radio transceiver, a small micro-controller and a power source. The main constraints in sensor network are that each node has limited energy, memory and computational power. Sensor networks are usually consists of automatic control with self-organizing capabilities and often there is no need for human intervention after they are deployed. Sensor networks are widely used in many application areas [33,52] that include target tracking. Sensor networks are used in military to detect and track threats such as enemy vehicles or toxin in the air.

A common practice in sensor network application has been to process collected data in a central processor. This kind of architecture is known as centralized sensor network. Centralized architectures are generally simpler to execute since the processing of data at one location can reduce the computational requirement of an algorithm. However, there are several drawbacks associated with this type of architecture. Firstly, that the network relying on one processor to perform the task of the network may introduce the single point failure problem. Secondly, in real-time applications, the central node may reside many hops away and sending data from one node to a central node may take too long. This may introduce latency, synchronization problems and imbalance workload in the network. Further, the centralized architecture utilizes a great deal of resources in communicating the data. Thus the centralized architecture keep the data processing simple while neglecting communication costs. However, in many real-time applications, the energy consumption corresponding to computational work is lower than the consumption related to communication. Thus major focus will be on avoiding high data transmission in this type of network setup. Distributed processing over the sensor network can be used to alleviate the problems inherent with the centralized architectures.

Distributed algorithms based on particle filters to track a single object have gained much attention by sensor network researchers. One of the first developments for nonlinear/non-Gaussian systems was in [34]. The paper proposes two methods to use the distributed sensor network. One of them based on likelihood factorization of particles of the filter and the other one based on adaptive data-encoding scheme.

The primary focus of this chapter is on creating distributed algorithms which minimizes energy consumption and network communication relating to sensor data fusion when multiple time varying targets are present in the monitored area. We assume

that the sensor management, which involves the optimization of sensor resources to collect data, and communication issues such as network protocols are already efficient enough.

The proposed algorithm uses a sensor network to track multiple targets based on a powerful scheme using the SMC implementation of the PHD filter, which was described earlier. The PHD filter has been showed to be more effective in tracking time varying multiple number of targets that avoids model-data association problems. Each sensing platform makes multiple, noisy measurements of an underlying, time-varying state that describes the monitored system. The monitored system involves potentially nonlinear target dynamics described by Markovian state-space model, nonlinear measurements, and non-Gaussian process and measurement noises. Each sensing platform reports measurements to a node in the network, which performs sequential estimation of the current system state using the PHD filter. The crucial consideration is what information needs to be transmitted over the network in order to perform online estimation of the current state of the monitored system, whilst attempting to minimize communication overhead. Simulation results demonstrate the efficiency of the proposed algorithm for a team of bearing only sensors.

7.2 Distributed PHD Filter

The underlying sensor network architecture consists of two different types of devices. The sensors are responsible for collecting measurements from the target state and report them to computational nodes. Nodes are responsible for running filters to track the targets. Each sensor is usually associated with a single node and therefore located in a position closer to that node. In general, using sensor management techniques, the

sensors closest to the object's predicted position are activated to take measurements. These active sensors may not belong to a single node and therefore a mechanism must be in place to utilize all the sensor data to obtain the estimates.

The most simple solution is to use a centralize fusion center which collects all the measurements and processes the data using PHD filter to obtain the multitarget state estimates. However, this method is not desirable as discussed earlier, since communication cost of relaying the measurement data obtained at a node far away from the central fusion center will be high. Sensor network that are used to track single target may employ a leader node, which could be located near the object to reduce the communication cost. However, this approach may not be suitable when multiple targets are present in the monitored area.

We propose a SMC PHD filter that is maintained in all the computational nodes. This idea may seem to be worse than employing a centralized computational node since communication costs can be dramatically increased due to the maintenance of additional PHD filters that are not previously necessary. However, a proposal can be given to maintain lower communication costs compared to centralized fusion approach.

It is often the case that a query is periodically made for the estimates of the objects in a tracking system. Thus, it is not necessary to update the PHD filter over the network at each measurement time epoch. Thus, we can employ a scheme, vectorization, where measurement data over several scans are collected and sent together. This scheme reduces the communication cost by eliminating the need to send extra packet headers.

We can also employ a scheme to ensure that the update of the PHD filter is done in a effective manner. Various methods can be used to address the update such as

using particles from other communication nodes, communicating measurements over the network. We will consider transmitting measurement data over the network since the other methods involve higher communication and computational costs.

In sensor network, it is quite possible that each node may have enough active sensors to track an object by itself with reasonable tracking accuracy. Therefore, a PHD filter can be used to obtain estimates based on the measurements collected from sensors local to that node. Since these nodes maintain PHD filters based on local measurements, they can also be used to encode the measurement data without much additional computational cost. Therefore a quantization and encoding technique can simply be employed for the measurement data using the PHD filter.

The proposed framework will be performed in two layers. The first layer collects the measurement data for a period of time that are local to each node while maintaining a local PHD filter using its associated sensors. Then, periodically, all the measurements are exchanged to all other nodes in the network to maintain the global PHD filter. We will discuss the following two important steps employed in the proposed algorithm to reduce the communication cost.

7.2.1 Quantization and Encoding

The data transmitted over the sensor network is generally quantized to certain extent. In our approach, we perform quantization by dividing the measurement space into small bins and transmitting the label of the bin where the measurement data lies. This scheme may still require a substantial number of bits to represent the actual data. One approach to reduce the number of bits transmitted is to use Lloyd-Max algorithm [34, 63, 75]. However, the Lloyd-Max algorithm is computationally demanding and

thus a simpler encoding scheme is better suited for real-time applications. A Huffman encoding [35] based scheme is proposed in [54] for sensor network to track single target.

The proposed quantization process is performed by blindly predicting the SMC PHD filter at each time step and constructing the histogram based on the expected measurements of each predicted particle state. The blind prediction of the SMC PHD filter may not give the exact location of the target position. However, it might maintain a reasonable tracking accuracy if it is used only for short period of time. It should be noted that computational cost of blindly predicting the target state particle is relatively low, since it does not include update and resampling steps, which are computationally intensive. Then a Huffman tree [35] is constructed using the histogram of the expected measurement as the dictionary. This dictionary can be used to encode the quantized measurement data. If the predicted SMC PHD filter gives a good representation of the multitarget state, then measurements should lie in most densely-populated bins and the encoded quantized measurements should contain very few bits. We need to have the same Huffman dictionary in order to decode the receive data in another node. Thus it is necessary to maintain identical copy of the predicted SMC PHD filter in all the nodes that decodes the incoming measurement data. This could be achieved by initializing all the nodes with same SMC PHD filter and this result in a very high reduction in the communication costs in the sensor network.

7.2.2 Vectorization

Since, in general, the query is made periodically and it is not necessary to have the accurate estimation at every time step, the global PHD filter is performed occasionally. We need to have the measurements to be disseminated to all the other nodes in order to update the global PHD filter. Therefore, the measurement data at each node is collected over several time steps and sent together. This procedure greatly reduces the communication costs on sending the overhead packets. During the intervening time steps, the local nodes will have up-to-date local estimates. Once the vectored measurements are shared by each node in the network, they can perform PHD filtering on the received data. At this point, the global estimate of the multitarget state can be formed in all the nodes.

7.2.3 Algorithm

In the proposed approach, each node maintains an identical copy of the SMC PHD filter. Initially, this is achieved by initializing all the filters using the same random seed. In order to encode the measurement data at time k, the samples of time step $k-v$ are blindly propagated using target dynamic model. Here $k-v$ is the last time instant when the data is transmitted over the network. A histogram of the expected measurement values is constructed for each sensor. Then, the measurements are quantized and encoded by bin labelling using Huffman encoding algorithm, where the histogram is used to train the Huffman encoder. The encoded measurements are then transmitted to all the other nodes and each node decodes the data to obtain the quantized measurements. Each node performs filtering using these quantized measurements to obtain the target state estimates. All the nodes use the same set of

measurement data to update the filter and thus identical copy of filter is maintained at the end of each network communication.

The distributed SMC PHD filter is detailed below.

1. Initialization $k = 0$

   - Initialize SMC PHD filter on each node $n = 1, \ldots, N$ using the same random seed to generate identical particle distribution on all the nodes.
   - For each node $n = 1, \ldots, N$.
     - Generate samples $\left\{\mathbf{x}_0^{(s)}\right\}_{s=0}^{L_0}$.

2. Quantization and Encoding

   - Local Estimation
     - Perform filtering using the SMC PHD filter acting only on the measurements local to the node.
   - Quantization

     For the length of the vector $v = 1, \ldots, V$
     - For each node $n = 1, \ldots, N$
       * For $s = 1, \ldots, L_{k-v}$, predict $\mathbf{x}_{k|k-v}^{(s)}$.
       * Calculated expected values of measurements $g_k^{(s)} = E(z_k|\mathbf{x}_{k|k-v}^{(s)})$.
       * Create a histogram of the $g_k^{(s)}$ values with h equal bins encompassing the range of values [min($\mathbf{g}_k$),max($\mathbf{g}_k$)].
       * Use the histogram to form Huffman tree H_k and encode quantized measurements $\tilde{\mathbf{z}}_k$.

3. Network Communication

- For each $n = 1, \ldots, N$, send the vectorized measurements $\tilde{\mathbf{z}}'_k = \{\tilde{\mathbf{z}}_{k-v}, \ldots, \tilde{\mathbf{z}}_k\}$ to all the other nodes in the network.

4. Global Estimate

- For each node $n = 1, \ldots, N$, create the Huffman tree H_k, to reconstruct the quantized data $\tilde{\mathbf{z}}'_k$.
- Using $\{\tilde{\mathbf{z}}_{k-v}, \ldots, \tilde{\mathbf{z}}_k\}$ as the set of measurements obtained from time interval $\{k-v, \ldots, k\}$, perform filtering to obtain the global state estimates.

7.3 Simulations

In this section, a two-dimensional tracking example is presented to show the effectiveness of the proposed algorithm. The simulation uses four computational nodes placed at $(-5\times10^3, 1\times10^4)$, $(1.5\times10^4, 1\times10^4)$, $(-5\times10^3, -1\times10^4)$ and $(1.5\times10^4, -1\times10^4)$. Each node receives observations from 4 bearing-only sensors at a time interval of $T = 30$ s. The target-generated measurements corresponding to target j on sensor i of node n are given by

$$z_k^{i,j,n} = \tan^{-1}\left(\frac{y_k^j - y_S^i}{x_k^j - x_S^i}\right) + v_k^{i,j,n} \tag{7.1}$$

where $v_k^{i,j,n}$ is i.i.d. sequence of zero-mean Gaussian variables with standard deviation 0.01 rad. (x_k^j, y_k^j) and (x_S^i, y_S^i) denote the locations of target j and sensor i at time step k, respectively. The average false alarm rate $\lambda_k = 4 \times 10^{-3}$ rad^{-1}, the volume of observation of each sensor $= 2\pi$ rad, the false alarms are distributed uniformly within the sensor field of view, and the probability of target detection $p_D(\mathbf{x}_k) = 1$,

regardless of the target state $\mathbf{x}_k$.

The simulation generates random synthetic target trajectories with a nearly constant velocity model [13] for target motion. The mathematical model for target dynamics is given by

$$\mathbf{x}_{k+1} = \mathbf{F}\mathbf{x}_k + \mathbf{v}_k \tag{7.2}$$

where

$$\mathbf{F} = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7.3}$$

and $\mathbf{v}_k$ is zero-mean white Gaussian noise with covariance $\mathbf{Q}$ given by

$$\mathbf{Q} = \begin{bmatrix} \frac{1}{3}T^3 & \frac{1}{2}T^2 & 0 & 0 \\ \frac{1}{2}T^2 & T & 0 & 0 \\ 0 & 0 & \frac{1}{3}T^3 & \frac{1}{2}T^2 \\ 0 & 0 & \frac{1}{2}T^2 & T \end{bmatrix} q \tag{7.4}$$

where $q_1 = 0.001$ denotes the level of process noise in target motion. Target 1 and Target 2 are present in the scene at $k = 0$ and Target 3 enters at time $k = 10$. The initial target positions are $(-3 \times 10^3, 5 \times 10^3)$, $(-1.4 \times 10^4, 8 \times 10^3)$ and $(6 \times 10^3, 0)$ m. The initial target velocities are (5, 0), (-5, -5) and (0, -5) ms^{-1}. Figure 7.1 shows the target trajectories and the architecture of sensor network.

The following filter parameters are taken: Number of particles representing one target = 1000; Number of particles used to represent new targets J_k = (10 × number of intersecting measurement points within the monitored area); The probability of

target survival = 0.99; The probability of target spawning = 0; The probability of target birth = 0.05;

Figures 7.2, 7.3 and 7.4 show the root mean squared error (RMSE) of position estimates of target 1, 2 and 3, respectively. The RMSEs of local node estimates are smaller at lower vector lengths because the local estimates are updated more frequently.

Figure 7.5 shows compression ratio of the Huffman encoding scheme plotted against vector length v. The overhead bits are not included in this case. The figure indicates that the compression is substantial especially at lower vector lengths. At lower vector lengths, the filter provides more accurate indication of where the measurements are most likely to be.

Figure 7.6 displays the average number of bits, including packet headers, transmitted by a node per time step for different values of v. As the vector length increases, the number of bits wasted on overhead bits decreases and this property is clearly displayed. However, longer vector lengths are not preferred since it delays computation of the global estimates which in turn results in larger estimation errors.

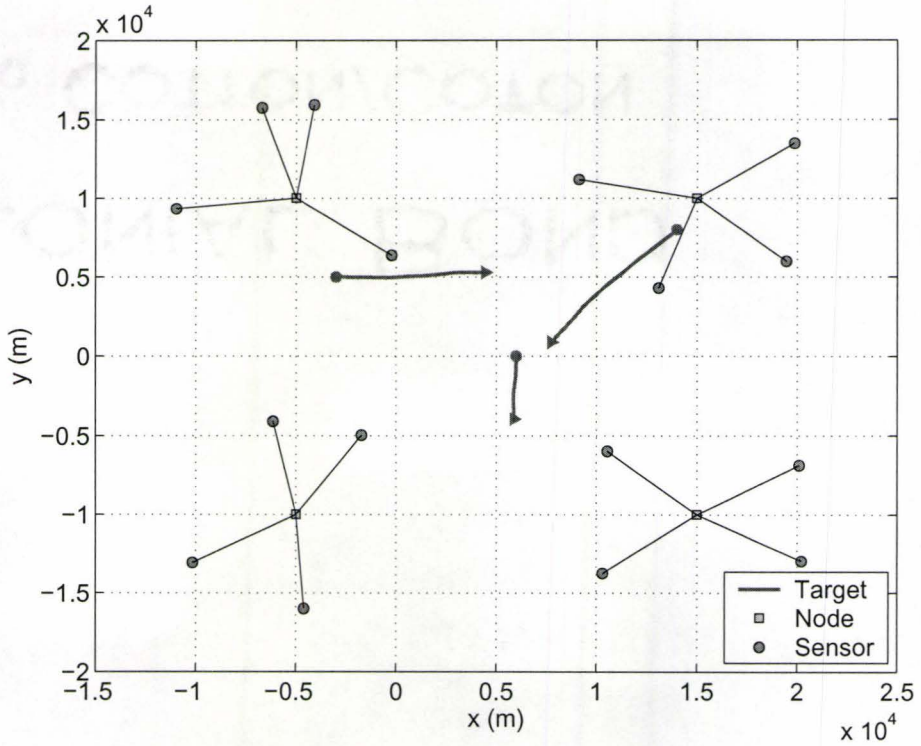


Figure 7.1: Scenario

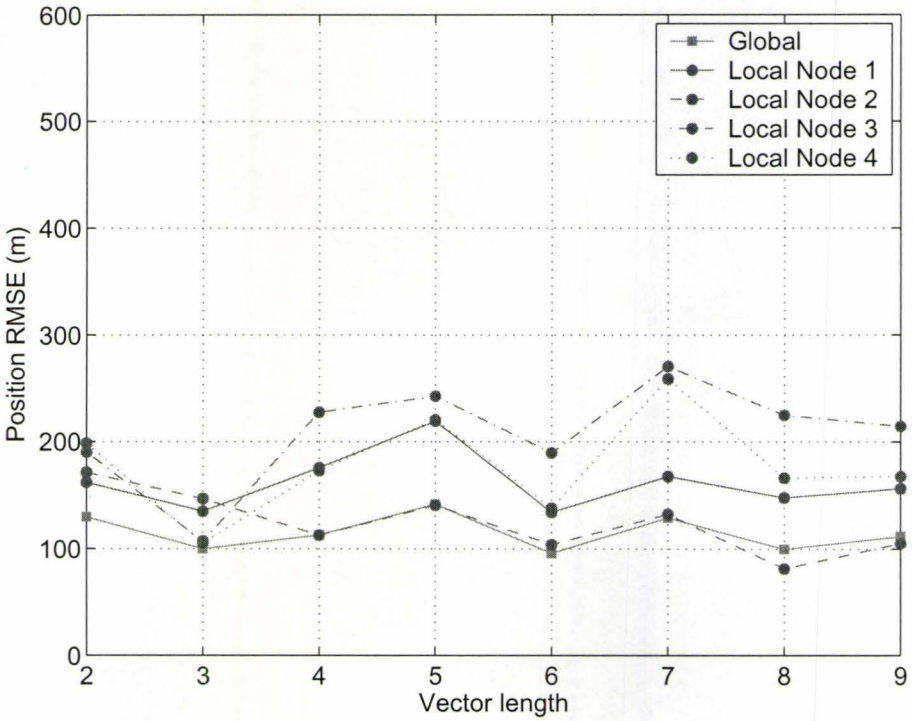


Figure 7.2: Target position estimation RMSE - Target 1

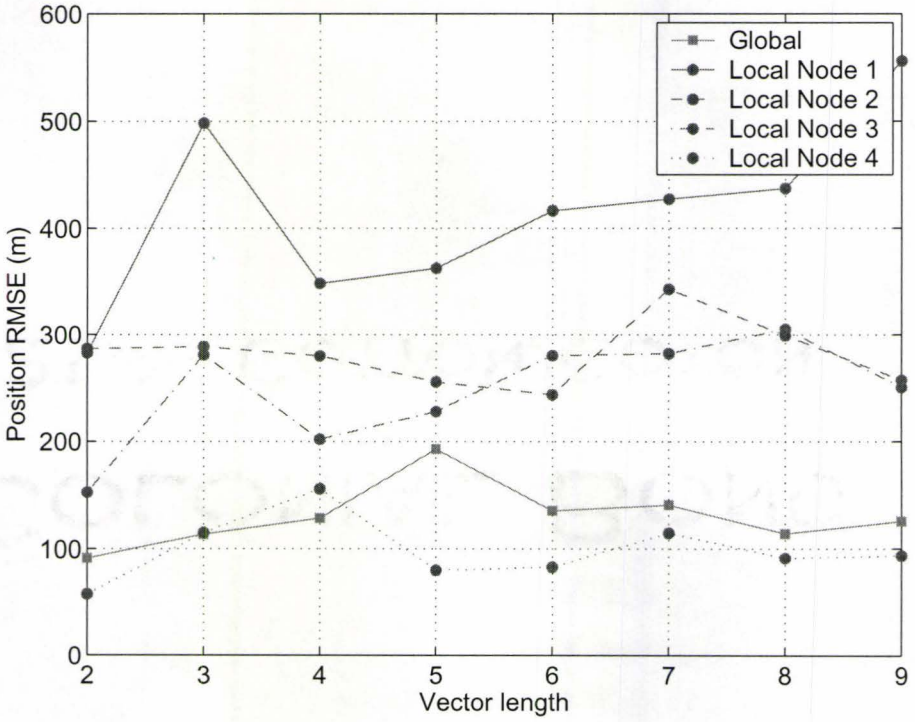


Figure 7.3: Target position estimation RMSE - Target 2

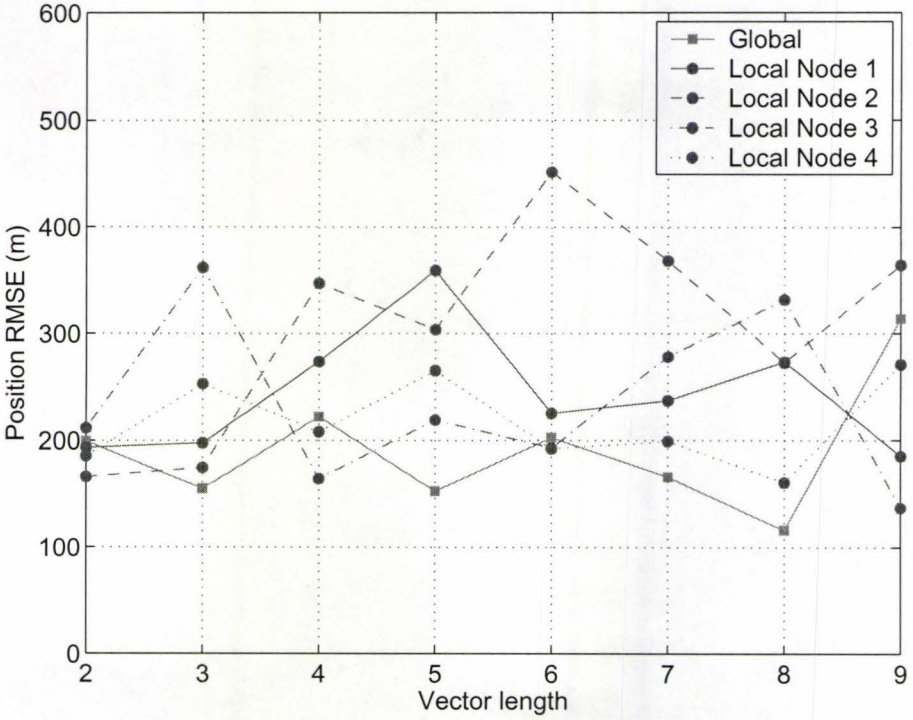


Figure 7.4: Target position estimation RMSE - Target 3

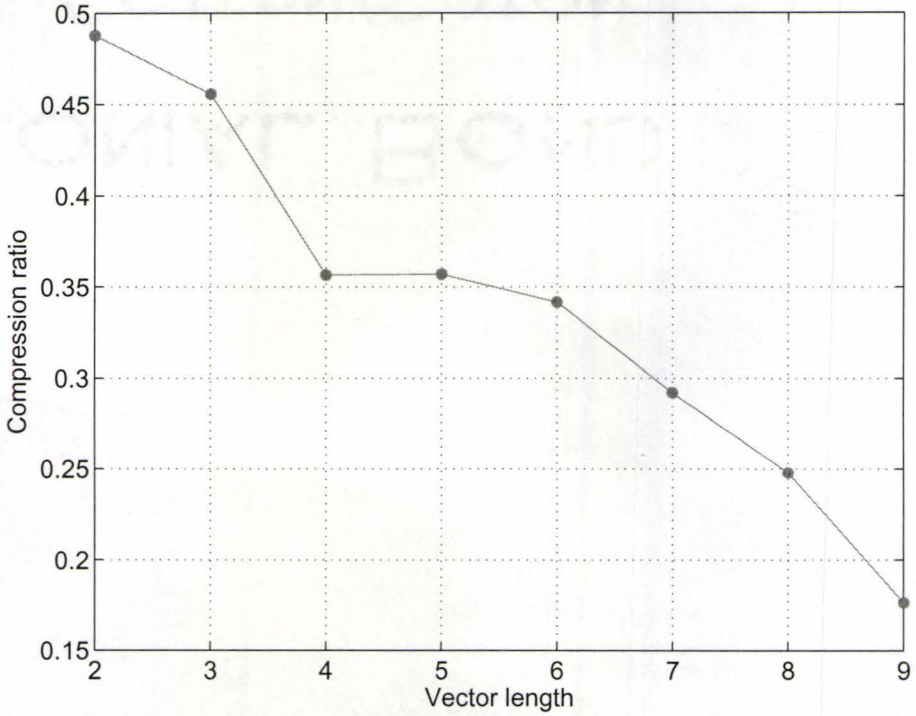


Figure 7.5: Compression ratio

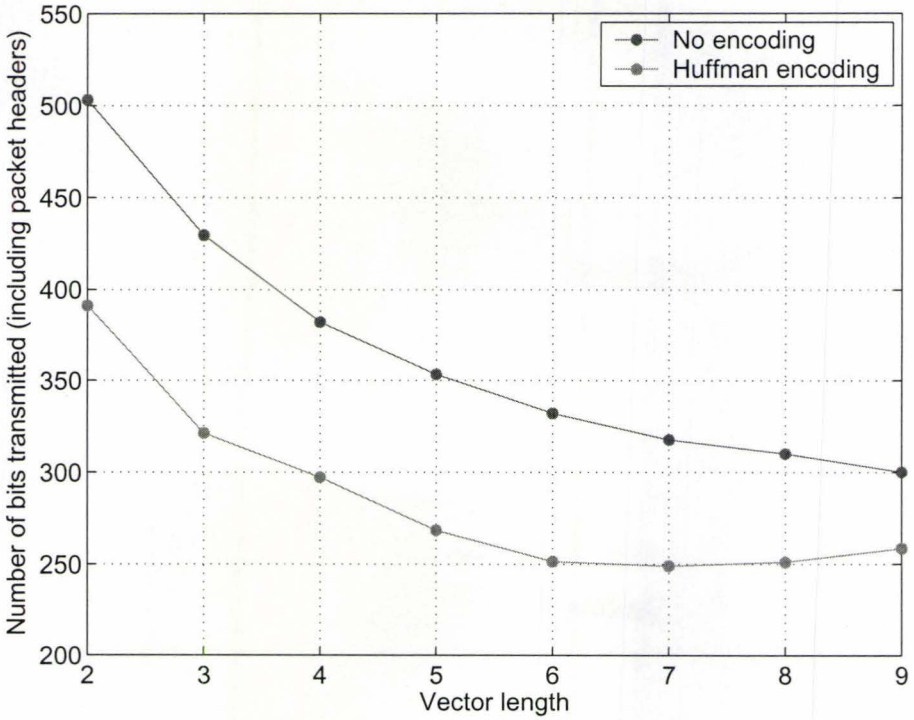


Figure 7.6: Number of bits transmitted

Chapter 8

Spline Filter for Bayesian Tracking

8.1 Introduction

This chapter presents a novel method for the realization of nonlinear/non-Gaussian Bayesian filtering based on spline interpolation. SMC approaches are widely used in nonlinear/non-Gaussian Bayesian filtering in which the densities are approximated by taking discrete set of points in the state space. In contrast to the SMC methods, the proposed approach uses spline polynomial interpolation to approximate the probability densities as well as the likelihood functions. A good representation of the probability densities is an essential issue in the success of the filtering algorithm, especially in nonlinear filtering, since the probability densities in nonlinear filtering could be multi-modal. An advantage of the proposed approach is that it retains the accurate density information and thus a target probability at any region in the state space can easily be obtained by evaluating the integral of the polynomial. Further, the probability densities are represented with polynomials of low order and any further processing on probability densities could be performed with less computation.

The proposed approach uses the B-spline interpolation in order to maintain the positivity of probability density functions and likelihood functions. Simulation results are presented to compare the performance and computational cost of the proposed algorithm with an SMC method.

8.2 Nonlinear Bayesian Filering

The general target dynamic model, which governs the evolution of the state sequence $\{x_k\}$ of a target, given by

$$x_k = f_k(x_{k-1}) + v_k \tag{8.1}$$

where f_k is possibly a nonlinear function of the state x_{k-1} and $\{v_k\}$ is an i.i.d. process noise sequence of known statistics.

The objective of target tracking is to recursively estimate x_k from measurements

$$z_k = h_k(x_k) + w_k \tag{8.2}$$

where h_k is, in general, a nonlinear function and $\{w_k\}$ is an i.i.d. measurement noise sequence of known statistics.

In filtering, we are interested in estimating the probability distribution of x_k condition up to the current time step. This leads to the pdf $p(x_k|z_{1:k})$ could be obtained probabilistically in two steps known as prediction and update. It is assumed that the initial pdf $p(x_0|z_0)$ of the state vector is available. Suppose the prior pdf $p(x_{k-1}|z_{1:k-1})$ at time $k-1$ is available. Then the pdf associated with the predicted state can be

obtained using the system model (8.1) via Chapman-Kolmogorov equation

$$p(x_k|z_{1:k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|z_{1:k-1})dx_{k-1} \tag{8.3}$$

where $z_{1:i} = \{z_1, \ldots, z_i\}$ denotes the set of measurements up to time i.

The pdf associated with the update state is given by

$$p(x_k|z_{1:k}) = \frac{p(z_k|x_k)p(x_k|z_{1:k-1})}{p(z_k|z_{1:k-1})} \tag{8.4}$$

where the denominator

$$p(z_k|z_{1:k-1}) = \int p(z_k|x_k)p(x_k|z_{1:k-1})dx_k \tag{8.5}$$

is a normalization term. The normalization term depends on the likelihood function $p(z_k|x_k)$ defined by the measurement model (8.2).

8.3 B-splines

This section gives a brief introduction to B-splines and its properties. The reader is referred to [22] for a detailed background on splines. The spline interpolation in B-form is given by [22],

$$\mathcal{B}(x) = \sum_{j=1}^{n} \mathbf{B}_{j,p,t}(x)\alpha_j \tag{8.6}$$

where

$$\mathbf{B}_{j,p,t}(x) \triangleq (t_{j+p} - t_j)[t_j, \ldots, t_{j+p}](\cdot - x)_+^{p-1} \tag{8.7}$$

is the j th B-spline of order p for the given knot sequence t. The notation $[t_j, \ldots, t_{j+p}]g$ stands for the divided difference of order p of g at the sites $t_j, \ldots, t_{j+p}$ and $(x)_+ = \max\{x, 0\}$ stands for the truncation function. The placeholder notation is used to indicate that the k th divided difference of the function $(t-x)_+^{p-1}$ of the two variables t and x is to be taken by fixing x and considering it as a function of t alone.

From definition, the first order B-spline is given by

$$\mathbf{B}_{j,1,t}(x) = \begin{cases} 1 & \text{if } t_j \leq x \leq t_{j+1} \\ 0 & \text{otherwise} \end{cases} \tag{8.8}$$

Starting from the first order, the higher order B-splines can be constructed using the recurrence relation

$$\mathbf{B}_{j,p,t} = w_{j,k}\mathbf{B}_{j,p-1,t} + (1 - w_{j+1,p})\mathbf{B}_{j+1,p-1,t} \tag{8.9}$$

with

$$w_{j,k}(x) \triangleq \frac{x - t_j}{t_{j+p-1} - t_j} \tag{8.10}$$

8.3.1 Positivity Property of B-splines

The B-spline $B_{j,p,t}$ is made up of at most p nontrivial polynomial pieces and vanishes outside the interval $[t_j \ldots t_{j+p}]$ and is positive on the interior of that interval [22], i.e.,

$$\mathbf{B}_{j,p,t}(x) > 0, \qquad t_j < x < t_{j+p} \tag{8.11}$$

while

$$t_j = t_{j+p} \quad \Longrightarrow \quad \mathbf{B}_{j,p,t}(x) = 0 \tag{8.12}$$

With the positivity property of B-splines and having $\alpha_i \geq 0$ for all i, we can ensure the spline representation of probability densities and likelihood functions are all positive.

8.3.2 Knot Insertion Property of B-splines

Additional knots can be inserted with the following knot insertion property [19]. If the knot sequence $\hat{t}$ is obtained from the knot sequence t by the insertion of just one term, say x, then for any $f \in \$$,

$$\sum_j \alpha_j \mathbf{B}_{j,p,t} = f = \sum_j \hat{\alpha}_j \mathbf{B}_{j,p,\hat{t}} \tag{8.13}$$

with

$$\hat{\alpha}_j = (1 - \hat{w}_j p(x))\alpha_{j-1} + \hat{w}_{jp}(x)\alpha_j, \qquad \text{for all } j \tag{8.14}$$

where $\$$ is the span$(\mathbf{B}_{j,p,t})$.

8.3.3 Stable Evaluation of B-splines

B-splines can be evaluated in a numerical stable way using de Boor algorithm [22]. This is done by using the recurrence relation (8.9) and first order B-spline (8.8) as follows.

Suppose $t_i < t_{i+1}$ and $x \in [t_i \;\; t_{i+1}]$. The values of all B-splines not automatically

zero at x fit into an array as follows.

$$
\begin{array}{cccccc}
 & & & & & 0 \\
 & & & & 0 & \\
 & & & \cdot & & \mathcal{B}_{i-k+1,k} \\
 & & 0 & & \mathcal{B}_{i-k+2,k-1} & \\
 & 0 & & \cdot & & \mathcal{B}_{i-k+2,k} \\
0 & & \mathcal{B}_{i-2,3} & & \mathcal{B}_{i-k+3,k-1} & \\
 & \mathcal{B}_{i-1,2} & & & \cdot & \mathcal{B}_{i-k+3,k} \\
\mathcal{B}_{i,1} & & \mathcal{B}_{i-1,3} & \cdot & & \cdot \\
 & \mathcal{B}_{i,2} & & & \cdot & \cdot \\
0 & & \mathcal{B}_{i,3} & & \mathcal{B}_{i-1,k-1} & \\
 & 0 & & \cdot & & \mathcal{B}_{i-1,k} \\
 & & 0 & & \mathcal{B}_{i,k-1} & \\
 & & & \cdot & & \mathcal{B}_{i,k} \\
 & & & & 0 & \\
 & & & & & 0
\end{array}
$$

Since we know the first column by (8.8), we can compute each entry in a subsequent column by (8.9). Suppose we have already computed the j numbers $\mathcal{B}_{i-j+1,j}(x) \ldots \mathcal{B}_{i,j}(x)$ and have saved them in the vector $\mathbf{b} \triangleq [b_1, \ldots b_j]$. If the vector $\mathbf{b}' \triangleq [b'_1, \ldots, b'_{j+1}]$ is to contain the $j+1$ numbers $\mathcal{B}_{i-j,j+1}(x), \ldots, \mathcal{B}_{i,j+1}(x)$ of the next column in that order, using (8.9), is given by,

$$b'_r = (x - t_{i-j+r-1}) \frac{b_{r-1}}{t_{i+r-1} - t_{i-j+r-1}} + (t_{i+r} - x) \frac{b_r}{t_{i+r} - t_{i-j+r}} \qquad (8.15)$$

for $r = 1, \ldots, j+1$ and with $b_0 \triangleq b_{j+1}$. We can introduce the quantities

$$\delta_{sj}^{L} \triangleq x - t_{i-j+s} \tag{8.16}$$

$$\delta_{s}^{R} \triangleq t_{i+s} - x \tag{8.17}$$

for $s = 1, \ldots, k-1$.

Then we can write (8.15) in the form

$$b'_r = \delta_{r-1,j}^{L} \frac{b_{r-1}}{\delta_{r-1,j}^{L} + \delta_{r-1}^{R}} + \delta_{r}^{R} \frac{b_r}{\delta_{rj}^{L} + \delta_{r}^{R}} \qquad \text{for } r = 1, \ldots, j+1 \tag{8.18}$$

8.3.4 Integral of B-splines

The integral of B-spline, for $t_1 \leq x \leq t_s$, is given by

$$\int_{t_1}^{x} \sum_{i=1}^{n} \alpha_i \mathbf{B}_{ip}(y) dy = \sum_{i=1}^{s-1} \left(\sum_{j=1}^{i} \alpha_j (t_{j+p} - t_j)/p \right) \mathbf{B}_{i,p+1}(x) \tag{8.19}$$

8.3.5 Construction of B-splines

The basis sequence of B-splines $\mathcal{B}_j$ for $j = 1, \ldots, n$ are linearly independent [22]. Thus the the space define the by the span of the B-spline basis functions $\mathcal{B}_j$'s is n-dimensional. If the strictly increasing sequence $t = \{\tau_1, \tau_2, \ldots, \tau_n\}$ of data sites is given, then for a given function g, the spline $f \triangleq \sum_{j=1}^{n} \alpha_j \mathcal{B}_j$ agrees with g at τ, if and only if

$$\sum_{j=1}^{n} \alpha_j \mathcal{B}_j(\tau_i) = g(\tau_i) \qquad \text{for } i = 1, \ldots, n \tag{8.20}$$

This is a linear system of n equations with n unknown α_i's with coefficient matrix $(\mathcal{B}_j(\tau_i))$, the spline collocation matrix. The matrix could be easily checked, if it is

invertible, using the following Schoenberg-Whitney theorem [93].

Schoenberg-Whitney Theorem

Let τ be strictly increasing and such that $a < t_i = \ldots = t_{i+r} = \tau_j < b$ implies $r < k-1$. The the matrix $A = (\mathcal{B}_j(\tau_i))$ of the linear system (8.20) is invertible if and only if

$$B_i(\tau_i) \neq 0 \qquad \text{for } i = 1, \ldots, n \tag{8.21}$$

8.4 Spline Filter

This section describes the prediction and update steps of the optimal recursive Bayesian filtering using spline representation of density and likelihood functions.

8.4.1 Prediction

Suppose we have the spline representation $\mathcal{B}_{k-1|k-1}(x_{k-1})$ of the prior density $p(x_{k-1}|z_{1:k-1})$ over the interval $[t_1(k-1) \ldots t_s(k-1)]$, i.e.,

$$\mathcal{B}_{k-1|k-1}(x_{k-1}) \approx p(x_{k-1}|z_{1:k-1}) \tag{8.22}$$

Then, the prediction step of the spline filter involves solving the integral (8.3) numerically. This can be done by transforming the spline interpolation of the prior density $\mathcal{B}_{k-1|k-1}(x_{k-1})$ using the target transition function $f_{k-1}(\cdot)$ and convoluting the transformed density with the density function of the additive noise v_k. However, this procedure is tedious since it involves finding the inverse function of $f_{k-1}(\cdot)$, which may not be available in general. We propose the following point mass approach to

solve the integral (8.3). It should be noted that the whole region of the state space is not represented by $\mathcal{B}_{k-1|k-1}(x_{k-1})$ and it is assumed to be zero outside the outside the interval $[t_1(k-1)\dots t_s(k-1)]$. The interval $[t_1(k-1)\dots t_s(k-1)]$ of the state space is decomposed into N_s cells $\{x_{k-1}^i : i = 1,\dots,N_s\}$ and $\mathcal{B}_{k-1|k-1}(x_{k-1})$ is evaluated at these points. The i th point is calculated by,

$$y_{k-1}^i = \mathcal{B}_{k-1|k-1}(x_{k-1}^i) \tag{8.23}$$

$$x_k^i = f_{k-1}(x_{k-1}^i) + n_{k-1}^i \tag{8.24}$$

Then spline interpolation of the predicted density $\mathcal{B}_{k|k-1}(x_{k-1})$ is constructed using $\left\{\{x_k^i, y_{k-1}^i\} : i = 1,\dots,N_s\right\}$. The new interval is set to $[t_1'(k-1)\dots t_s'(k-1)]$.

8.4.2 Update

Now we have the spline representation $\mathcal{B}_{k|k-1}(x_k)$ of the predicted density $p(x_k|z_{1:k-1})$ and the update step involve solving (8.4). This can be done by constructing the spline representation $\mathcal{B}_{l,k}(x_k)$ of the likelihood function $p(z_k|x_k)$ over the interval $[t_1'(k-1)\dots t_s'(k-1)]$. Here we select the same knot sequence to construct the spline interpolation of the likelihood function in order to reduce the computational cost of multiplying $\mathcal{B}_{k|k-1}(x_{k-1})$ and $\mathcal{B}_{l,k}(x_k)$.

The spline interpolation of the update density is then calculated by

$$\mathcal{B}_{k|k}(x_k) = \frac{\mathcal{B}_{k|k-1}(x_{k-1})\mathcal{B}_{l,k}(x_k)}{\int_{t_1(k-1)}^{t_s(k-1)} \mathcal{B}_{k|k-1}(x_{k-1})\mathcal{B}_{l,k}(x_k)} \tag{8.25}$$

Now we have the spline function for the updated posterior density $\mathcal{B}_{k|k}(x_k)$. However, in many instances, we can take off the insignificant end portion of the posterior

density that have lower target probability. This is done by finding the integral value of the $\mathcal{B}_{k|k}(x_k)$ over the interval $[t'_{1+i}(k-1) \;\; t'_{1+i+1}(k-1)]$ for each i starting from 1 until the first value exceeds a threshold value. This procedure is repeated for $[t'_{s-i}(k-1) \;\; t'_{s-i+1}(k-1)]$ for i starting from 1 and these intervals are removed. The new interval is then set to $[t_1(k) \ldots t_s(k)]$.

8.4.3 Find estimate

The mean of the state estimate $\hat{x}_k$ at time step k is calculated by

$$\hat{x}_k = \int_{t_1(k)}^{t_s(k)} x_k \mathcal{B}_{p,k}(x_k) dx_k \tag{8.26}$$

Finding the higher moments of the estimate is also straightforward with the spline filter approach.

8.5 Simulations

This sections presents a one dimensional nonlinear estimation problem that has been analyzed before in many publications 4, 46. Consider the following nonlinear state space model

$$x_k = f_k(x_{k-1}) + w_{k-1} \tag{8.27}$$

where

$$f_k(x_{k-1}) = \frac{x_{k-1}}{2} + \frac{25x_{k-1}}{1+x_{k-1}^2} + 8\cos(1.2k) \tag{8.28}$$

and where w_{k-1} is a zero mean Gaussian random variable with variance $Q_{k-1} = 10$.

The measurement model is given by

$$z_k = \frac{x_k^2}{20} + v_{k-1} \tag{8.29}$$

where v_{k-1} is a zero mean Gaussian random variable with variance $R_{k-1} = 1$. The initial target position $x_1 = 0.1$.

We compare spline filter and Sampling Importance Resampling (SIR) particle filter. The reader is referred to [4] for the results on various other nonlinear filtering algorithms for the given example. For reference, the true states x_k and measurements z_k are shown in Figures 8.1 and 8.2, respectively. The filters were initialized with same target state, with mean 0.1 and variance 3. The spline filter is constructed using order $p = 4$ and 10 knots. The performance and computational cost of the spline filter is compared with two SIR particle filters, one with 50 particles and another with 1000 particles, and the results are given in Table 8.1. Even though, Root Mean Squared Error (RMSE) is not exceptionally meaningful in this example since the densities here lead to multi-modal, it is included here for the reason that RMSE has been used extensively in the literature and facilitates quantitative comparison. The average RMSE of the spline filter estimates shows that the performance of the filter is as good as the particle filter, based on 100 Monte Carlo runs. The spline filter is computationally expensive than the particle filter of same performance. This is because the construction of splines requires a vast amount of computational power. It should be noted that the spline filter retains the probability density information and gives a more natural representation over the discrete approximation.

Figures 8.3 and 8.4 show the true states and, the filter estimate of posterior mean of spline filter and particle filter, respectively. The figure shows that the spline filter

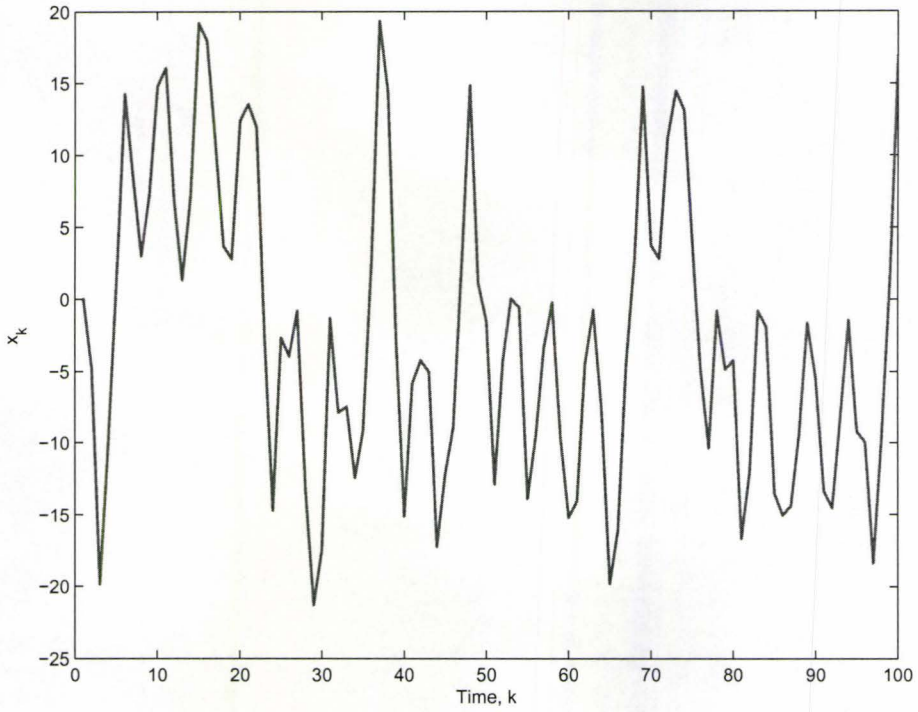


Figure 8.1: True values of x_k as a function of k

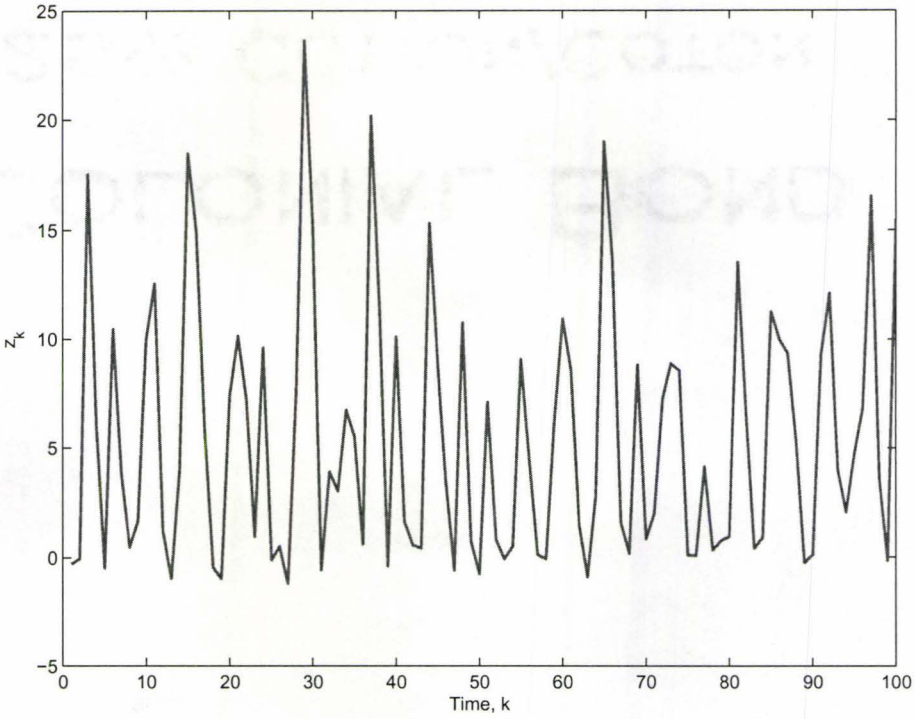


Figure 8.2: Measurements of z_k of the states x_k shown in Figure 8.1

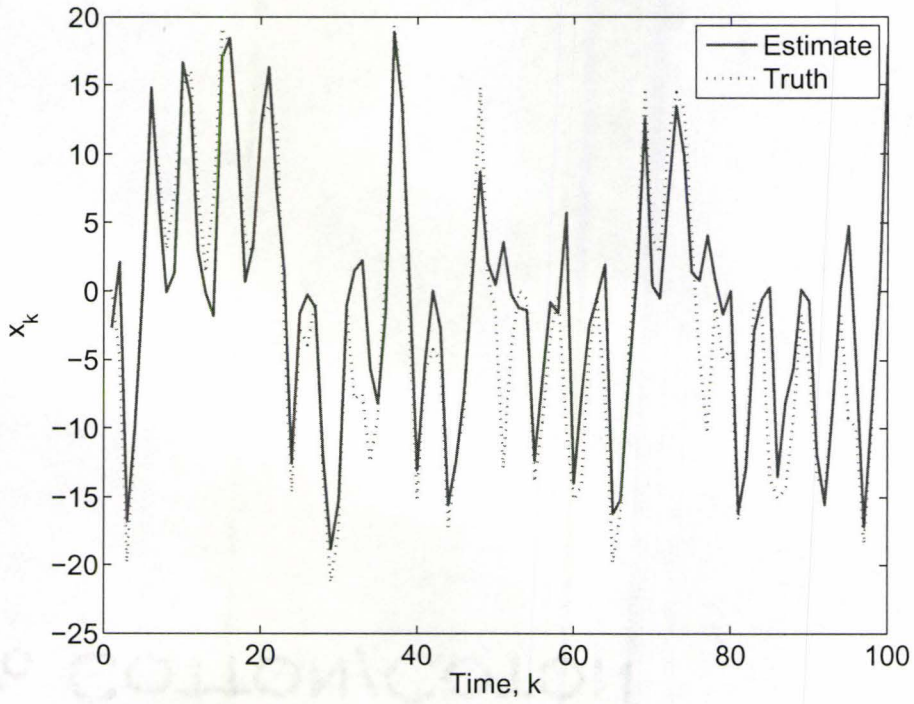


Figure 8.3: Spline filter estimate of posterior mean

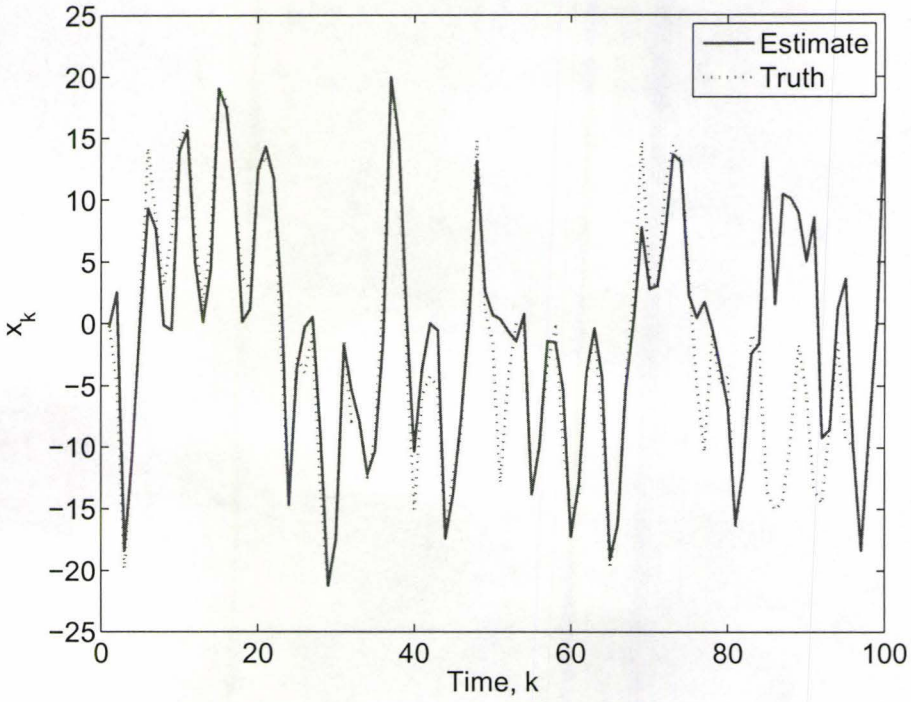


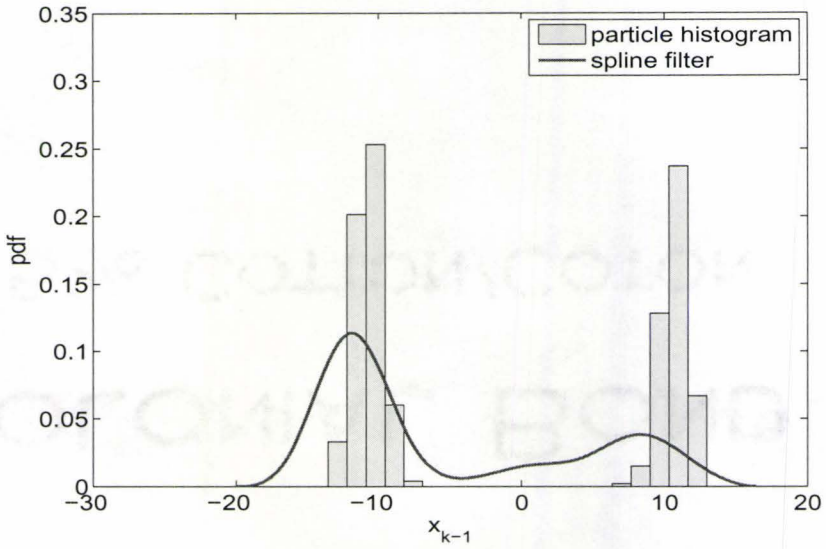
Figure 8.4: SIR particle filter estimate of posterior mean

Filter	RMSE	Average time per run (in seconds)
Spline filter	4.83	5.23
SIR particle filter (50 particles)	5.58	0.16
SIR particle filter (1000 particles)	4.85	3.86

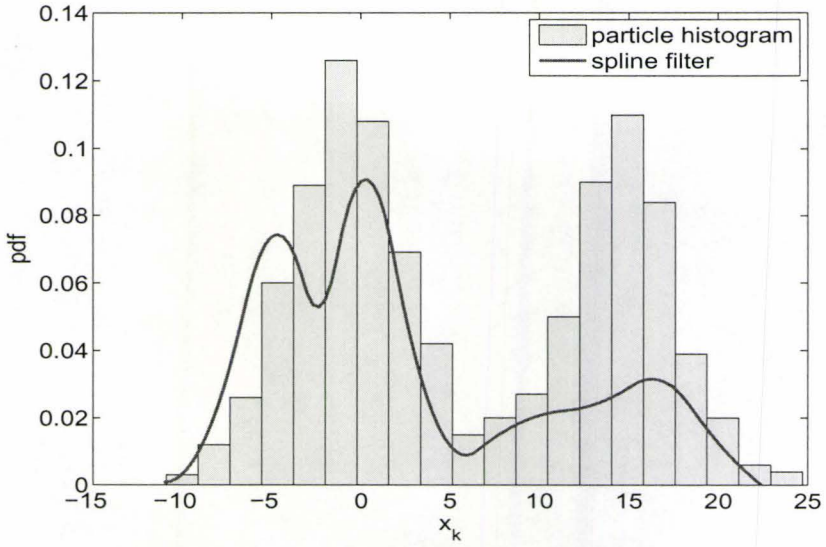
Table 8.1: Comparison of algorithms (based on 100 Monte Carlo runs)

and particle filter perform well and the mean of the estimates closely follows the true state, for both filters.

Figure 8.5 shows the probability densities and likelihood function in one cycle of the recursion of the spline filter at time step $k = 10$. Histogram of particles (for 1000 particles) also displayed for comparison. The spline interpolation of the density functions are closer to the particle filter histograms at each step of the recursion.

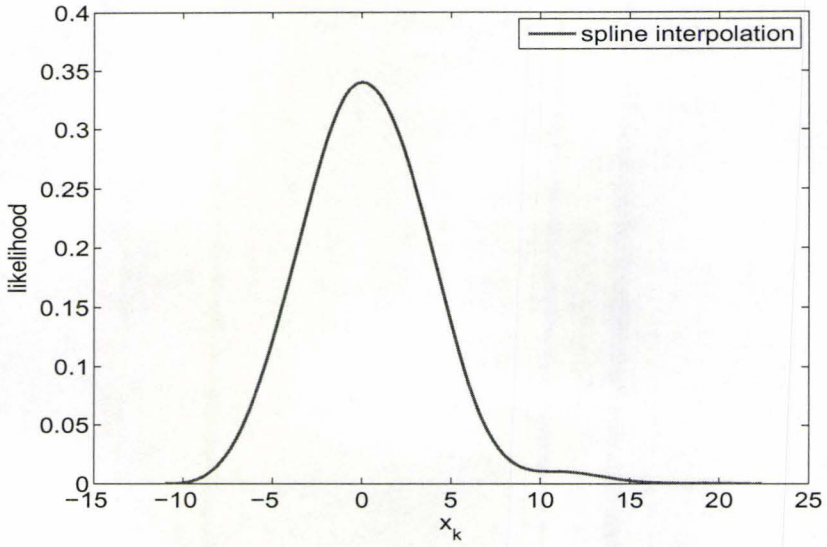


(a) Prior density

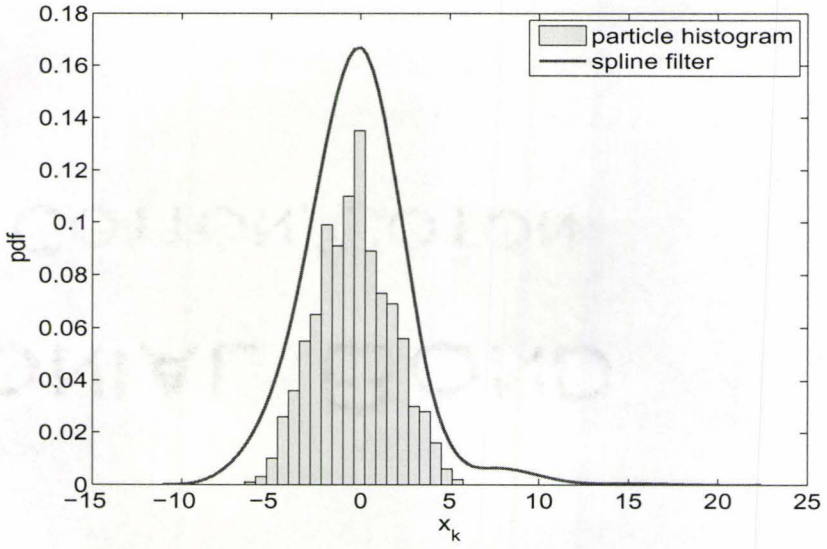


(b) Predicted density

Figure 8.5: One cycle of recursive estimation (at $k = 10$)



(a) Likelihood function



(b) Posterior density

Figure 8.6: One cycle of recursive estimation (at $k = 10$)

Chapter 9

Conclusions and Future Work

9.1 Conclusions

In this thesis, we considered the problem of tracking multiple targets in nonlinear/non-Gaussian systems in cluttered environments. Our main focus was on tracking maneuvering targets and stealthy targets. Multitarget tracking is a difficult problem since it involves the model-data association problem which becomes more challenging when the number of the targets vary with time. We proposed a PHD based filtering approach to solve many practical multitarget tracking problems. The PHD filter is a computationally efficient solution to multitarget tracking problems, and the SMC implementation makes the filter applicable in general scenarios with nonlinear/non-Gaussian systems.

We presented a new multiple-model approach to enhance the PHD filter's capability to handle target maneuvers. Maneuvering targets might switch between different modes of operation, and tracking using a single-model filter might fail since the filter may be accurate for only one mode of operation. It has been shown that

multiple-model PHD filter performs significantly better than the single-model filters. The single-model filters that matched only one kind of target dynamics showed poor adaptation to maneuvers, resulting in large estimation errors when the target maneuverability is high. We can conclude that a multiple-model approach is essential to ensure satisfactory performance in such cases.

Then, we considered the problem of tracking multiple low SNR targets, which cannot be simply tracked using conventional tracking algorithms, based on thresholded the sensor data. Thresholding throws away potential information and results in poor performance in tracking dim targets. We presented a sequential Monte Carlo probability hypothesis density filter for recursive Bayesian track-before-detect algorithm. The filtering algorithm operates on raw measurement frames in a recursive manner and it is computationally more efficient than conventional batch track-before-detect algorithms. The proposed algorithm can detect and track multiple weak targets in noise. The advantage of the TBD PHD filter over other methods is that it does not have a restriction on the maximum possible number of targets and offers flexibility in the target and noise models. Through simulation on a 20×20 grid of data, it was shown that multiple targets with SNR as low as 6 dB can be detected and tracked.

This thesis also addressed the problem of tracking a time varying number of targets in a distributed sensor network. The sensor network contains a large number of sensors and these sensors report observations to their associated computational nodes. The nature of the monitored system is nonlinear and non-Gaussian. A distributed implementation of an SMC PHD filter is proposed. The distributed implementation of the filter has the advantage over the centralized architecture that it does not suffer from a single point failure problem. Further, it reduces the effects related to latency, synchronization and imbalance workload in the network. The proposed algorithm

quantizes measurements and the histogram based on the SMC PHD filter samples is effectively used to encode the quantized measurement using a data-encoding scheme. Further, each node waits for several time steps before they broadcast measurements over the network and this measurement vectorization procedure saves a substantial amount of bits wasted on overhead bits. A simulation of a two-dimensional target tracking using a network of 4 computational nodes and 16 bearing-only sensors is presented with results to show the effectiveness of the proposed algorithm.

Finally, we presented a novel approach to optimal nonlinear/non-Gaussian Bayesian filtering using B-spline interpolation. The proposed algorithm uses a continuous approximation of the probability densities and likelihood functions using B-spline polynomial interpolation. Numerical methods to evaluate the prediction and update steps of the filter are illustrated. This method has the advantages over the widely used discrete approximation known as particle filters in that it retains accurate density information in a continuous form and thus evaluating the target probability over a region in state space is straightforward. Further, it does not suffer from the degeneracy problem that is common in discrete approximation approach of density functions. The knots of the spline interpolation are selected in a way that only the region of the state space, in which the target probability density is significant, is represented to reduce the computational costs. The simulations are presented to compare the spline filter with SIR particle filter and the spline filter performs as good as SIR particle filter at the expense of more computational power.

9.2 Future Work

In this thesis, the spline filter was demonstrated with a nonlinear example. The work could be extended to multi-dimensional case using tensor product approaches. One of the main challenges in applying tensor product approach is that the computational complexity increases exponentially with the dimension. However, using pruning techniques, it is possible approximate the filtering equations only to represent the state space in which the density is significant. This could significantly reduce the computational power needed.

Since the spline filter is general enough to apply for multi-modal densities, it could be a very good candidate to implement multi-target tracking filters such as the PHD filter. However, it is necessary to model the prediction and update operator of the PHD filter using spline approach.

In some systems, the measurements may not be received in the sequence as they originated due to network delays and the tracker may receive some of the measurements in an out-of-sequence fashion. The current PHD filter equations are only applicable to the measurement set that is in the correct chronological order. Deriving the PHD filter equations to handle out-of-sequence measurement could be a potential area of research for many practical systems.

Bibliography

[1] G. A. Ackerson and K.S. Fu, "On state estimation in switching environments", *IEEE Transactions on Automatic Control*, pp. 10–17, Vol. 15, No. 1, February 1970.

[2] D. L. Alspach abd H. W. Sorenson, "Nonlinear Bayesian estimation using Gaussian sum approximation", *IEEE Transactions on Automation and Control*, Vol. 17, No. 4, pp. 439–448, August 1972.

[3] J. Arnold, S. W. Shaw and H. Pasternack, "Efficient target tracking using dynamic programming", *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 29, No. 1, pp. 44–56, January 1993.

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

[5] D. J. Ballantyne, H. Y. Chan, and M. A. Kouritzin, "A branching particle-based nonlinear filter for multi-target tracking" *Proceedings of the International Conference on Information Fusion*, Vol. 1, pp. WeA2:3–10, Montreal, 2001.

[6] Ba-Ngu Vo, S. Singh and A. Doucet, "Sequential Monte Carlo implementation of the PHD filter for multi-target tracking", *Proceedings of the Sixth International Conference of Information Fusion*, Vol. 2, pp. 792–799, July 2003.

[7] B.-N. Vo, S. Singh and A. Doucet, "Sequential Monte Carlo methods for multitarget filtering with random finite sets", *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 41, No. 4, pp. 1224–1245, October 2005.

[8] B. Vo and S. Singh, Technical aspects of the Probability Hypothesis Density recursion, *Tech. Rep. TR05-006, EEE Dept., The University of Melbourne, Australia 2005*

[9] Ba-Ngu Vo and Wing-Kin Ma, "A closed-form solution for the probability hypothesis density filter", *Proceedings of the 8th International Conference on Information Fusion*, pp. 25–28, Vol. 2, Philadelphia, July 2005

[10] B.-N. Vo and W.-K. Ma, "The Gaussian Mixture Probability Hypothesis Density Filter", *IEEE Transactions on Signal Processing*, Vol. 54, Iss. 11, pp. 4091–4104, November 2006.

[11] Y. Bar-Shalom, K. C. Chang and H. A. P. Blom, "Tracking splitting targets in clutter by using an Interacting Multiple Model Joint Probabilistic Data Association filter", *Multitarget Multisensor Tracking: Applications and Advances*, pp. 93–110, Vol. II, Artech House, 1992.

[12] Y. Bar-Shalom and T. E. Fortmann, *Tracking and Data Association*, New York, Academic Press, January 1988.

[13] Y. Bar-Shalom, X. Li, and T. Kirubarajan, *Estimation with Application to Tracking and Navigation*, Wiley, 2001.

[14] Y. Bar-Shalom, *Multitarget-Multisensor Tracking: Applications and Advances*, Artech House, 1991

[15] Y. Barniv, "Dynamic programming solution for detecting dim moving targets", *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 21, pp. 144–156, January 1985.

[16] Samuel S. Blackman, *Multiple-Target Tracking with Radar Applications*, Artech House, December 1986.

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

[18] W. D. Blair, G. A. Watson, T. Kirubarajan and Y. Bar-Shalom, "Benchmark for radar resource allocation and tracking in the presence of ECM", *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 34, No. 4, pp. 1097–1114, October 1998.

[19] W. Boehm, "Inserting new knots into B-spline curves", *Computer Aided Design*, Vol. 12, No. 41, pp. 199–201, 1980.

[20] Y. Boers and J. N. Driessen, "Particle filter based detection for tracking", *Proceedings of American Control Conference*, pp. 3755–3760, Arlington, VA, June 2001.

[21] Y. Boers and J. N. Driessen, "Multitarget particle filter track before detect application", *IEE Proceedings on Radar, Sonar and Navigation*, Vol. 151, No. 6, pp. 351–357, December 2004.

[22] Carl de Boor, *A Practical Guide to Splines*, Springer-Verlag, New York, 2001.

[23] R. R. Brooks, P. Ramanathan and A. M. Sayeed, "Distributed target classification and tracking in sensor networks", Proceedings of the IEEE, Vol. 91, No. 8, pp. 1163–1171, August 2003.

[24] H. A. P. Blom, "A sophisticated tracking algorithm for ATC surveillance data", *Proceedings of the International Radar Conference*, pp. 393–398, Paris, May 1984.

[25] H. A. P. Blom and Y. Bar-Shalom, "The interacting multiple model algorithm for systems with Markovian switching coefficients", *IEEE Transactions on Automatic Control*, Vol. 33, No. 8, pp. 780–783, August 1988.

[26] H. A. P. Blom and E. A. Bloem, "Combining IMM and JPDA for tracking multiple maneuvering targets in clutter", *Proceedings of the Fifth International Conference on Information Fusion*, Vol. 1, pp. 705–712, July 2002.

[27] H. A. P. Blom and E. A. Bloem, "Joint IMMPDA particle filter", *Proceedings of the Sixth International Conference of Information Fusion*, Vol. 2, pp. 785–792, July 2003.

[28] Y. Boers and J. N. Driessen, "Interacting multiple model particle filter", *IEE Proceedings on Radar, Sonar and Navigation*, Vol. 150, No. 5, pp. 344–349, October 2003.

[29] R. S. Bucy and H. Youssef, "Nonlinear filter representation via spline functions", *Proceedings of the 5th Symposium on Nonlinear Estimation*, pp. 51-60, 1974.

[30] B. D. Carlson, E. D. Evans and S. L. Wilson, "Search radar detection and track with the Hough transform, part I: System concept", *IEEE Transactions on Aerospace and Electronic Systems*, pp. 102–108, Vol. 30, No. 1, January 1994.

[31] C. B. Chang and M. Athans, "State estimation for dicrete system with switching parameters", *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 14, Iss. 3, pp. 418–425, May 1978.

[32] B. Chen and J. K. Tugnait, "Multisensor tracking of multiple maneuvering targets in clutter using IMM/JPDA fixed-lag smoothing", *Proceedings of the 38th IEEE Conference on Decision and Control*, Vol. 5, pp. 5058–5063, December 1999.

[33] C.-Y. Chong and S. P. Kumar, "Sensor networks: Evolution, opportunities, and challenges", *Proceedings of the IEEE*, Vol. 91, pp. 1247-1256, August 2003.

[34] M. Coates, "Distributed particle filters for sensor networks," *Proceedings of theThird International Symposium on Information Processing in Sensor Networks*, pp. 99–107, Berkeley, CA, April 2004.

[35] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein, *Introduction to Algorithms*, MIT Press, Cambridge, MA, 2nd edition, 2001.

[36] D. J. Daley and D. Vere-Jones, *An Introduction to the Theory of Point Processes*, New York: Springer-Verlag, 1988.

[37] R. J. Dempster, S. S. Blackman and T. S. Nichols, "Combining IMM filtering and MHT data association for multitarget tracking", *Proceedings of the Twenty-Ninth Southeastern Symposium on System Theory*, pp. 123–127, March 1997.

[38] D. Docenko and K. Berzins, "Spline histogram method for reconstruction of probability density functions of clusters of galaxies", *Lecture Notes on Physics* Vol. 626, pp. 294-301, 2003.

[39] A. Doucet, S. J. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering", *Statistical Computing*, pp. 197-208, Vol. 10, No. 3, July 2000.

[40] A. Doucet, N. de Freitas, and N. J. Gordon, *Sequential Monte Carlo Methods in Practice*, New York: Springer-Verlag, May 2001.

[41] A. Doucet, B. Vo, C. Andrieu and M. Davy, "Particle filtering for multitarget tracking and sensor management", *Proceedings of the International Conference on Information Fusion*, pp. 474-481, Vol. 1, Annapolis, MD, 2002.

[42] A. El-Fallah, T. Zajic, R. Mahler, B. Ravichandran and R. Mehra, "Multitarget nonlinear filtering based on spectral compression and probability hypothesis density", *SPIE Proceedings on Signal Processing, Sensor Fusion and Target Recognition X*, Vol. 4380, pp. 207–216, 2001.

[43] Ozgur Erdinc, Peter Willett, and Yaakov Bar-Shalom, "A physical-space approach for the probability hypothesis density and cardinalized probability hypothesis density filters", *SPIE Proceedings on Signal and Data Processing of Small Targets*, Vol. 6236, pp. 623619:1–12, May 2006.

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

[45] I. R. Goodman, R. P. S. Mahler, and H. T. Nguyen, *Mathematics of Data Fusion*, Kluwer Academic Publishers, 1997.

[46] Gordon, N. J., Salmond, D., and Smith, A., "Novel approach to nonlinear/non-Gaussian Bayesian state estimation", *IEE Proceedings, Pt. F, Radar and Signal Processing*, pp. 107-113, Vol. 140, Iss. 2, April 1993.

[47] J. Goutsias, R. Mahler, H. Nguyen, *Random Sets: Theory and Applications*, New York: Springer-Verlag, 1997.

[48] D. L. Hall and J. Llinas, editors. *Handbook of Multisensor Data Fusion*, chapter 7, CRC Press, 2001.

[49] John A. Hartigan, *Clustering Algorithms*, John Wiley & Sons, 1975.

[50] J. C. Hassab, *Underwater Signal and Data Processing*, Boca Raton, FL, CRC Press, 1989.

[51] J. Hoffman and R. Mahler, "Multitarget miss distance via optimal assignment", *IEEE Transactions on System, Man, and Cybernetics–Part A*, Vol. 34, Iss. 3, pp. 327–336, 2004.

[52] T. T. Hsieh, "Using sensor networks for highway and traffic applications", *IEEE Potentials*, Vol. 23, pp. 13-16, AprilMay 2004.

[53] C. Hue, J.-P. Le Cadre and P. Perez, "Tracking multiple objects with particle filtering", *IEEE Transactions on Aerospace and Electronic Systems*, pp. 791–812, Vol. 38, Iss. 3, Jul 2002.

[54] G. Ing and M. J. Coates , "Parallel particle filters for tracking in wireless sensor networks", IEEE 6th Workshop on Signal Processing Advances in Wireless Communications,pp. 935–939, June 2005.

[55] A. Johansen, S. Singh, A. Doucet and B. Vo, "Convergence of the sequential implementation of the PHD filter", *Technical Report CUED/F-INFENG/TR-517*, 2005.

[56] Simon J. Julier and Jeffery K. Uhlmann, "A New Extension of the Kalman Filter to nonlinear Systems", *In The Proceedings of AeroSense: The 11th International Symposium on Aerospace/Defense Sensing, Simulation and Controls, Multi Sensor Fusion, Tracking and Resource Management II*, SPIE, 1997.

[57] R. E. Kalman, "A new approach to linear filtering and prediction problems", *Transactions of the ASME - Journal of Basic Engineering*, Vol. 82, pp. 35-45, 1960.

[58] E. W. Kamen, "Multiple target tracking based on symmetric measurement equations", *IEEE Transactions on Automation and Control*, pp. 371–374, Vol. AC-37, 1992.

[59] A. F. Karr, *Point Processes and Their Statistical Inference* (2nd ed.), Marcel Dekker, 1991.

[60] T. Kirubarajan and Y. Bar-Shalom, "Kalman filter versus IMM estimator: When do we need the latter?", *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 39, No. 4, pp. 1452–1457, October 2003.

[61] W. Koch, "Fixed-interval retrodiction approach to Bayesian IMM-MHT for maneuvering multiple targets", *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 36, No. 1, pp. 2–14, January 2000.

[62] L. Lin, Y. Bar-Shalom, T. Kirubarajan, "Track labeling and PHD filter for multitarget tracking", *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 42, No.3, pp. 778–795, July 2006.

[63] S. Lloyd, "Least squares quantization in PCM", *IEEE Transactions on Information Theory*, Vol. 28, pp. 129-137, March 1982.

[64] R. Mahler, "Detecting, tracking and classifying group targets: A unified approach", *SPIE Proceedings on Signal Processing, Sensor Fusion and Target Recognition X*, Vol. 4380, pp. 217–228, 2001.

[65] R. Mahler, "Multi-target moments and their application to multi-target tracking", *Proceedings of the Workshop on Estimation, Tracking and Fusion: A Tribute to Yaakov Bar-Shalom*, pp. 134–166, Monterey, CA, 2001.

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

[67] R. Mahler, Random set theory for target tracking and identification, In D. L. Hall and J. Lindas (Eds.), *Handbook of Multisensor Data Fusion*, Ch. 14, Boca Raton, FL, CRC Press, 2002.

[68] R. Mahler, "Bayesian cluster tracking using a generalized Cheeseman approach", *SPIE Proceedings on Signal Processing, Sensor Fusion and Target Recognition X*, Vol. 5096, pp. 334–345, 2003.

[69] R. P. S. Mahler, "Multitarget Bayes filtering via first-order multitarget moments", *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 39, No. 4, pp. 1152–1178, October 2003.

[70] R. Mahler, "Objective functions for Bayesian control-theoritic sensor management, I: Multitarget first-moment approximation", *Proceedings of the IEEE Conference on Aerospace and Electronic Systems*, Big Sky, MT, March 2003.

[71] R. Mahler, "A theoretical foundation for the Stein-Winter probability hypothesis density (PHD) multi-target tracking approach", *Proceedings of the 2002 MSS National Symposium on Sensor and Data Fusion*, Vol. 1, (unclassified), San Antonio, TX, June 2000.

[72] R. Mahler and T. Zajic, "Bulk multitarget tracking using a first-order multitarget tracking filter", *SPIE Proceedings on Signal Processing, Sensor Fusion and Target Recognition XI*, Vol. 4729, pp. 175–186, 2002.

[73] K. Martinez, J. K. Hart, and R. Ong, "Environmental sensor networks", *IEEE Computer*, Vol. 37, pp. 50-56, August 2004.

[74] S. Maskell, N. J. Gordon, M. Rollason and D. Salmond, "Efficient particle filtering for multi-target tracking with application to tracking in structured images", *Proceedings of SPIE on Signal and Data Processing of Small Targets*, Vol. 4728, 2002.

[75] J. Max, "Quantization for minimum distortion", *IRE Transactions on Information Theory*, Vol. 6, pp. 7-12, March 1960.

[76] S. McGinnity and G. W. Irwin, "Multiple model bootstrap filter for maneuvering target tracking", *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 36, No. 3, pp. 1006–1012, July 2000.

[77] S. Mori, C. Chong, E. Tse and R. Wishner, "Tracking and identifying multiple targets without apriori identifications", *IEEE Transactions on Automatic Control*, pp. 401–409, AC-21, 1986.

[78] S. Musick, J. Greenwald, C. Kreucher and K. Kastella, "Comparison of particle method and finite difference nonlinear filters for low SNR target tracking", *Proceedings of The Fourth Annual Conference on Information Fusion*, Montreal, Canada, August 2001.

[79] A. Mutambara, *Decentralized Estimation and Control for Multisensor Systems*, CRC Press, 1998.

[80] N. Nandakumaran, K. Punithakumar and T. Kirubarjan, "Improved Multitarget Tracking Using Probability Hypothesis Density Smoothing", *SPIE Proceedings on Signal and Data Processing of Small Targets*, in press, 2007.

[81] S. C. Nardone, A. G. Lindgren and K. F. Gong, "Fundamental properties and performance of conventional bearings-only target motion analysis", *IEEE Transactions on Automatic Control*, Vol. 29, No. 9, pp. 775–787, 1984.

[82] K. Panta, B. Vo, S. Singh, "Improved Probability Hypothesis Density (PHD) Filter for Multitarget Tracking", *Third International Conference on Intelligent Sensing and Information Processing*, pp. 213–218, December 2005.

[83] K. Panta, B. Vo, S. Singh, and A. Doucet, "Probability hypothesis density filter versus multiple hypothesis tracking", *SPIE Proceedings on Signal Processing, Sensor Fusion, and Target Recognition XIII*, Ivan Kadar, Editor, Volume 5429, pp. 284–295, August 2004.

[84] D. Read, "An algorithm for tracking multiple targets", *IEEE Transactions on Automation and Control*, pp. 84–90, Vol. 24, Iss. 6, 1979.

[85] B. Ristic, S. Arulampalam and N. Gordon, "Detection and tracking of stealthy targets", *Beyond the Kalman Filter: Particle Filters for Tracking Applications*, ch. 11, Artech House, 2004.

[86] M. Rollason and D. Salmond, "A particle filter for track-before-detect of a target with unknown amplitude" *IEE Int. Seminar on Target Tracking: Algorithms and Applications*, Vol. 1, pp. 14/1–14/4, Enschede, Netherlands, October 2001.

[87] M. Rosencrant, G. Gordon and S. Thrun, "Decentralized sensor fusion with distributed particle filters", *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, Acapulco, Mexico, August 2003.

[88] M. G. Rutten, N. J. Gordon and S. Maskell, "Particle-based track-before-detect in Rayleigh noise", *SPIE Proceedings on Signal and Data Processing of Small Targets*, Vol. 5428, pp. 509–519. August 2004.

[89] M. G. Rutten, B., Ristic, N. J. Gordon, "A comparison of particle filters for recursive track-before-detect", *8th International Conference on Information Fusion*, Vol.1, pp. 169–175, July 2005

[90] D. J. Salmond and H. Birch, "A particle filter for track-before-detect", *Proceedings of the 2001 American Control Conference*, Vol. 5, pp. 3755–3760, Arlington, VA, June 2001.

[91] I. J. Schoenberg and A. Whitney, "On Pólya frequency functions. III. The positivity of translation determinants with an application to the interpolating problem by spline curves", *Transactions on Americal Mathematics Society*, Vol. 74, pp. 246–259, 1953.

[92] X. Sheng, Y. H. Hu, and Parameswaran Ramanathan, "Distributed particle filter with GMM approximation for multiple targets localization and tracking in wireless sensor network", *Fourth International Symposium on Information Processing in Sensor Networks*, pp. 181–188, April 2005.

[93] I. J. Schoenberg and A. Whitney, "On Pólya frequency functions. III. The positivity of translation determinants with an application to the interpolating problem by spline curves", *Transactions on Americal Mathematics Society*, Vol. 74, pp. 246–259, 1953.

[94] H. Sidenbladh, "Multi-target particle filtering for the probability hypothesis density", *Proceedings of the Sixth International Conference of Information Fusion*, Vol. 2, pp. 800–806, July 2003.

[95] D. L. Snyder and M. I. Miller, *Random Point Processes in Time and Space* (2nd ed.), New York: Springer, 1991.

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

[97] M. Tobias and A. D. Lanterman, "A probability hypothesis density-based multitarget tracker using multiple bistatic range and velocity measurements", *Proceedings of the Thirty-Sixth Southeastern Symposium on System Theory*, pp. 205–209, March 2004.

[98] S. M. Tonissen and Y. Bar-Shalom, "Maximum likelihood track-before-detect with fluctuating target amplitude", *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 34, No. 3, pp. 796–809, July 1998.

[99] A. H. Wang and R. L. Klein, "Implementation of nonlinear estimation using monospline", *Proceedings of the 13th IEEE Conference on Decision and Control*, pp. 1305-1307, 1976.

[100] R.B. Washburn, "A Random Point Process Approach to Multiobject Tracking", *Proceedings of the 1987 American Control Conference*, Vol. 3, pp. 1846–1852, Minneapolis, MN, 1987.

[101] T. Zajic and R. Mahler, "A particle-systems implementation of the PHD multitarget tracking filter", *SPIE Proceedings on Signal Processing, Sensor Fusion and Target Recognition XII*, Vol. 5096, pp. 291–299, 2003.

[102] T. Zajic, R. Ravichandran, R. Mahler, R. Mehra, and M. Noviskey, "Joint tracking and identification with robustness against unmodeled targets", *SPIE Proceedings on Signal Processing, Sensor Fusion and Target Recognition XII*, Vol. 5096, pp. 279–290, 2003.

[103] J. Zhang and H. Yue, "Improved identification algorithm for B-spline modelling of output probability density functions", *Proceedings of the 2004 IEEE International Symposium on Intelligent Control*, Taipei, Taiwan, September 2004.