

Syracuse University

SURFACE

Electrical Engineering and Computer Science -
Dissertations

College of Engineering and Computer Science

2011

Conditional Posterior Cramer-Rao Lower Bound and Distributed Target Tracking in Sensor Networks

Long Zuo
Syracuse University

Follow this and additional works at: https://surface.syr.edu/eecs_etd



Part of the [Electrical and Computer Engineering Commons](#)

Recommended Citation

Zuo, Long, "Conditional Posterior Cramer-Rao Lower Bound and Distributed Target Tracking in Sensor Networks" (2011). *Electrical Engineering and Computer Science - Dissertations*. 301.
https://surface.syr.edu/eecs_etd/301

This Dissertation is brought to you for free and open access by the College of Engineering and Computer Science at SURFACE. It has been accepted for inclusion in Electrical Engineering and Computer Science - Dissertations by an authorized administrator of SURFACE. For more information, please contact surface@syr.edu.

Abstract

Sequential Bayesian estimation is the process of recursively estimating the state of a dynamical system observed in the presence of noise. Posterior Cramér-Rao lower bound (PCRLB) sets a performance limit on any Bayesian estimator for the given dynamical system. The PCRLB does not fully utilize the existing measurement information to give an indication of the mean squared error (MSE) of the estimator in the future. In many practical applications, we are more concerned with the value of the bound in the future than in the past. PCRLB is an offline bound, because it averages out the very useful measurement information, which makes it an off-line bound determined only by the system dynamical model, system measurement model and the prior knowledge of the system state at the initial time.

This dissertation studies the sequential Bayesian estimation problem and then introduces the notation of conditional PCRLB, which utilizes the existing measurement information up to the current time, and sets the limit on the MSE of any Bayesian estimators at the next time step. This work has two emphases: firstly, we give the mathematically rigorous formulation of the conditional PCRLB as well as the approximate recursive version of conditional PCRLB for nonlinear, possibly non-Gaussian dynamical systems. Secondly, we apply particle filter techniques to compute the numerical values of the conditional PCRLB approximately, which overcomes the integration problems introduced by nonlinear/non-Gaussian systems.

Further, we explore several possible applications of the proposed bound to find algorithms that provide improved performance. The primary problem of interest is the sensor selection problem for target tracking in sensor networks. Comparisons are also made between the performance of sensor selection algorithm based on the proposed bound and the existing approaches, such as information driven, nearest neighbor, and PCRLB with renewal strategy, to demonstrate the superior performances of the proposed approach.

This dissertation also presents a bandwidth-efficient algorithm for tracking a target in sensor networks using distributed particle filters.

This algorithm distributes the computation burden for target tracking over the sensor nodes. Each sensor node transmits a compressed local tracking result to the fusion center by a modified expectation-maximization (EM) algorithm to save the communication bandwidth. The fusion center incorporates the compressed tracking results to give the estimate of the target state.

Finally, the target tracking problem in heterogeneous sensor networks is investigated extensively. Extended Kalman Filter and particle filter techniques are implemented and compared for tracking a maneuvering target with the Interacting Multiple Model (IMM).

Conditional Posterior Cramér-Rao Lower Bound and Distributed Target Tracking in Sensor Networks

by

LONG ZUO

M.S., Institute of Automation, Chinese Academy of Science, 2002

B.S., Xi'an Jiaotong University, P.R. China 1999

THESIS

Submitted in partial fulfillment of the requirements for the degree of Doctor of
Philosophy in Electrical Engineering and Computer Science in the Graduate
School of Syracuse University

December, 2010

COMMITTEE IN CHARGE:

Professor Pramod K. Varshney
Research Assist. Professor Ruixin Niu
Professor Chilukuri Mohan
Professor Kishan Mehrotra
Professor Biao Chen
Associate Professor Lixin Shen

© Copyright 2010
Long Zuo
All Rights Reserved

Acknowledgements

I would like to express my sincere gratitude to my advisor, Professor Pramod K. Varshney for his invaluable guidance and support throughout my graduate study in Sensor Fusion Group at Syracuse University. I would also like to express the appreciation to my co-advisor, Dr. Ruixin Niu, for providing me the opportunity to work on the fantastic topics and guiding me throughout my Ph.D study with kindly help and inspiring discussions. Without his never ending patience this thesis would probably not exist today. In addition, I am also grateful to professors Chilukuri K. Mohan and Kishan Mehrotra for the comments, remarks and suggestions that significantly improved the quality of my research.

I would also like to thank all my friends and colleagues at Syracuse University with whom I have had the pleasure of working over the years. These include Hao Chen, Min Xu, Priyadip Ray, Nikhil Padhye, Onur Ozdemir, Dazhi Chen, Arun Subramanian, Swarnendu Kar, Ashok Sundaresan, Satish G Iyengar, Renbin Peng and all other members of the Sensor Fusion Group to create a creative, stimulating and professional atmosphere.

Finally, I would like to dedicate this thesis to my loving parents and extended family members for their patience and support during these years.

Table of Contents

1	Introduction	1
1.1	Background	1
1.2	Main Contributions	5
1.3	Dissertation Organization	5
2	Sequential Bayesian Estimation	7
2.1	Bayesian Estimation Problems	7
2.2	Bayesian Filtering	8
2.2.1	Kalman Filter	9
2.2.2	Extended Kalman Filter	11
2.3	Bayesian Estimation through Sequential Monte Carlo Approach	12
2.3.1	Background	12
2.3.2	Sequential Monte Carlo Sampling for Bayesian Estimation	12
2.3.2.1	Monte Carlo Sampling	12
2.3.2.2	Importance Sampling	14
2.3.2.3	Sequential Importance Sampling	14
2.3.2.4	Sampling Importance Resampling	15
2.4	Discussion	16
3	Conditional Posterior Cramér-Rao Lower Bounds for Sequential Bayesian Estimation	17
3.1	Motivation	17
3.2	Classical Cramer-Rao Lower Bounds	20
3.3	Conditional PCRLB for Nonlinear Dynamical Systems	22
3.4	A Sequential Monte Carlo solution for Conditional PCRLB	29

3.4.1	General Formulation	31
3.4.2	Additive Gaussian Noise Case	34
3.4.3	Linear System with Additive Gaussian Noise Case	37
3.5	Simulation Results for Comparison	39
3.5.1	Conditional PCRLB vs Conditional MSE for UNGM	40
3.5.2	Weakly UNGM results	40
3.5.3	Conditional PCRLB vs Unconditional PCRLB	41
3.5.4	Exact Conditional PCRLB vs Its Recursive Approximation	42
3.5.5	Unconditional PCRLB with Renewal Strategy	43
3.6	Discussion	46
4	Sensor Selection for Target Tracking in Sensor Networks	48
4.1	Motivation	48
4.2	Sensor Selection Approaches	51
4.2.1	Information Driven	52
4.2.2	Nearest Neighbor	55
4.2.3	Conditional PCRLB	56
4.3	Target Tracking Model in Sensor Networks	57
4.3.1	Target Motion Model	57
4.3.2	Sensor Measurement Model	57
4.4	Sensor Selection Based on Conditional PCRLB	58
4.5	Simulation Results	62
4.5.1	Sensor Selection with Analog Data	63
4.5.2	Sensor Selection with Quantized Data	63
4.6	Discussion	65
5	Bandwidth-Efficient Distributed Particle Filter for Target Tracking in Sensor Networks	71
5.1	Motivation	71
5.2	Distributed Target Tracking	72
5.3	Tracking Based on Particle Filters	73
5.4	Gaussian Mixture Approximation to Particle Filter Estimates	74
5.4.1	Expectation-Maximization Algorithm	74
5.4.2	MLE of Gaussian Mixture Densities Parameters via EM	75

5.4.3	Dynamic EM for GMM	77
5.4.4	Best Linear Unbiased Estimators for Centralized Estimation	78
5.5	Simulation Results	79
5.5.1	Target Motion Model	79
5.5.2	Measurement Model	80
5.6	Discussion	82
6	Target Tracking in Heterogeneous Sensor Networks	85
6.1	Motivation	85
6.2	Sensor Network Setup	86
6.3	System Models	87
6.3.1	Target Motion Model	87
6.3.2	Sensor Measurement Model	88
6.4	Target Tracking Algorithms	89
6.4.1	Extended Kalman Filter	90
6.4.2	Particle Filtering	91
6.5	Simulation Results	92
6.5.1	Target Tracking Using Two Bearing-only Sensors	93
6.5.2	Target Tracking Using One Radar Sensor	94
6.5.3	Target Tracking Using One Range Sensor and One Bearing- only Sensor	95
6.5.4	Target Tracking Using Two Range Sensors	95
6.5.5	Target Tracking Using Bearing Only, Range and Radar Sensors	104
6.6	Summary	105
7	Conclusions	108
A	Proof of Proposition 1 in Chapter 3	111
B	Proof of Proposition 2 in Chapter 3	114
C	Approximation of $B_k^{22,b}$ by Particle Filters in Chapter 3 Section 3.4.2	116

TABLE OF CONTENTS

References

124

List of Tables

3.1	Comparison between $L_A(x_1 z_1)$ and \tilde{J}_1	46
4.1	Comparison of average CPU computational times (Analog data with two active sensors selected at each time, 30 time steps with $N = 300$)	64

List of Figures

1.1	Wireless Sensor Networks (from http://www2.ece.ohio-state.edu/~ekici/res_wmsn.html)	3
3.1	Plot of the true state x_k and observations z_k	41
3.2	Plot of filtering results by Extended Kalman filter and by particle filter for Example I	42
3.3	Plot of conditional posterior CRLB and conditional MSE for Example I	43
3.4	Plot of filtering results by Extended Kalman filter and by particle filter for Example II	44
3.5	Plot of conditional posterior CRLB and conditional MSE for Example II	45
3.6	Comparison of conditional and unconditional PCRLBs for Example III.	46
3.7	Comparison of conditional PCRLB between the one with error propagation and the one without error propagation	47
4.1	Sensor management based on feedback from recursive estimator.	49
4.2	Conditional PCRLB Tracking results with analog data	63
4.3	Renewal PCRLB Tracking results with analog data	64
4.4	Nearest Neighbor Tracking results with analog data	65
4.5	Information based Tracking results with analog data	66
4.6	Comparison of x-MSEs with analog data	66
4.7	Comparison of y-MSEs with analog data	67
4.8	Conditional PCRLB Tracking results with quantized data	67
4.9	Renewal PCRLB Tracking results with quantized data	68

4.10 Nearest Neighbor Tracking results with quantized data	68
4.11 Information based Tracking results with quantized data	69
4.12 Comparison of x-MSEs with quantized data	69
4.13 Comparison of y-MSEs with quantized data	70
5.1 Distributed target tracking results	80
5.2 Number of GMM components	81
5.3 Plot of number of particles vs number of bits transmitted	82
5.4 Plot of number of bits transmitted vs RMSE	83
5.5 Plot of cumulative number of bits transmitted vs number of particles	84
5.6 Fixed number of GMM components vs dynamic number of GMM components	84
6.1 Centralized fusion process	87
6.2 Simulated trajectory in a multi-sensor environment	93
6.3 2 bearing sensors - CV model with glint measurement noise : track- ing results	94
6.4 2 bearing sensors - CV model with glint measurement noise: MSE	95
6.5 2 bearing sensors - Maneuvering target: tracking results	96
6.6 2 bearing sensors - Maneuvering target: MSE	96
6.7 One radar sensor - CV model with glint measurement noise: track- ing results	97
6.8 One radar sensor - CV model with glint measurement noise: MSE	97
6.9 One radar sensor - Maneuvering target: tracking results	98
6.10 One radar sensor - Maneuvering target: MSE	98
6.11 One range and One bearing - CV model with glint measurement noise: tracking results	99
6.12 One range and One bearing - CV model with glint measurement noise: MSE	100
6.13 One range and One bearing - Maneuvering target: tracking results	100
6.14 One range and One bearing - Maneuvering target: MSE	101
6.15 Two range sensors - CV model with glint measurement noise: tracking results	101
6.16 Two range sensors - CV model with glint measurement noise: MSE	102

6.17 Two range sensors - Maneuvering target: tracking results 102

6.18 Two range sensors - Maneuvering target: MSE 103

6.19 Bearing + range + radar - CV model with glint measurement noise:
tracking results 104

6.20 Bearing + range + radar - CV model with glint measurement noise:
MSE 105

6.21 Bearing + range + radar - Maneuvering target: tracking results . 106

6.22 Bearing + range + radar - Maneuvering target: MSE 107

Chapter 1

Introduction

1.1 Background

Over the past decade, wireless networks have been deployed that are able to provide a large number of users with the ability to move diverse forms of information readily and thus have revolutionized business, industry, defense, science, education, research, and human interactions. Recent technological improvements have made the deployment of small, inexpensive, low-power, distributed devices, which are capable of local processing and wireless communication, a reality. This has resulted in the proliferation of wireless sensor networks that combine distributed sensing, computing, and wireless communications into a powerful technology and offer unprecedented resolution, unobtrusiveness, and autonomous operation for countless applications. At the same time, they offer numerous challenges consisting of monitoring and collecting the data, assessing and evaluating the information, formulating meaningful user displays, and performing decision-making and alarm functions especially under strict energy constraints, distributed operation, and scalability. This has generated world-wide interest in the basic and applied research and deployment of sensor networks [1].

As shown in Fig 1.1, a sensor network is composed of a large number of sensor nodes that are densely deployed either inside the region where the phenomenon is taking place or very close to it [2]. Sensor nodes with various functionalities usually allow random deployment in inaccessible terrains or disaster relief operations. Another unique feature of sensor networks is the cooperative effort of

sensor nodes. Sensor nodes either have the ability to send raw data to the nodes responsible for further processing or have the processing abilities to locally carry out simple computations and transmit only the required and partially processed data over a flexible network architecture with dynamic topologies. Therefore, sensor networks have the great potential to enable a large class of applications ranging from military scenarios to environmental control, consumer electronics and industrial equipment, health monitoring, warehouse inventory, etc. [3]. Some important applications are as follows:

- **Military Monitoring.** Special sensors can be deployed in the battlefield to gain enemy information for surveillance, to detect and track the enemy target movements, explosions and other phenomena of interest.
- **Environmental Monitoring.** Environmental sensors can be used to detect and monitor environmental changes in plains, forests, oceans, etc. or to monitor disaster areas to detect and characterize Chemical, Biological, Radiological, Nuclear, and Explosive (CBRNE) attacks.
- **Building Monitoring.** Sensors can be used in large buildings to monitor climate changes or vibration that could damage the structure of a building.
- **Traffic Monitoring.** Sensors can be used to monitor vehicle traffic on highways or in congested parts of a city.
- **Health Care.** Sensors can be used in biomedical applications to improve the quality of the provided care. For example, sensors are implanted in the human body to monitor medical problems like cancer and help patients maintain their health.

This dissertation focuses on target tracking problems in sensor networks. Due to the energy or bandwidth limitations, one of the main goals of target tracking problems in sensor networks is to provide most informative or accurate information about the moving target over time with constrained resources. Maximizing the tracking accuracy requires collecting the measurements from all the sensors, whose sensing ranges cover the moving target. However, due to the energy or

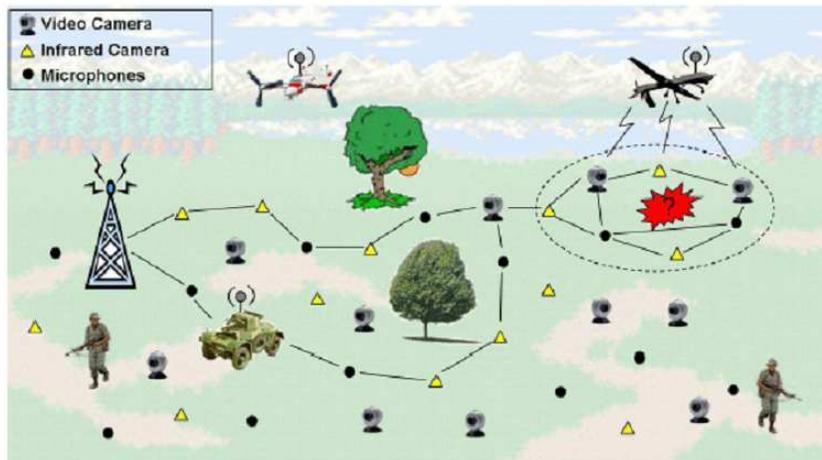


Figure 1.1: Wireless Sensor Networks (from http://www2.ece.ohio-state.edu/~ekici/res_wmsn.html)

bandwidth limitations, the number of active sensors should be kept to a minimum, while the requirement of the tracking accuracy is still satisfied. Sensor selection schemes that provide acceptable tracking accuracy and yet address energy and bandwidth limitations are, therefore, investigated in this dissertation.

The sensor selection problem can be defined as follows: given a set of sensors $\mathcal{S} = \{S_1, S_2, \dots, S_n\}$, a subset of k sensors need to be determined to provide the information in order to maximize a specific objective, for example, tracking accuracy. There are several existing approaches. One of the most widely used methods is the dynamic information-driven solution proposed by Zhao et al. [4]. They consider selecting the sensors that collectively have the maximum mutual information between the sensor measurements and the target state.

In the dissertation, a novel approach for evaluating the estimation performance in the sequential Bayesian estimation problems is proposed. Utilization of the novel conditional posterior Cramér-Rao lower bounds (PCRLBs) concept for the task of real-time sensor selection is proposed and compared with existing methods. PCRLBs for sequential Bayesian estimators provide performance bounds for general nonlinear filtering problems and their modified versions have

been used for sensor management in tracking and fusion systems. However, the unconditional PCRLB [5] is an off-line bound that is obtained by taking the expectation of the Fisher information matrix (FIM) with respect to the measurement and the state to be estimated. In other words, the standard unconditional PCRLB uses *a priori* information instead of currently sensed information. The novelty of the conditional PCRLB comes from the fact that it utilizes the information contained in the observation data up to the current time for a particular realization of the system state, which makes it an online bound adaptive to the particular track realization. This approach is expected to provide a more accurate and effective performance evaluation than the conventional unconditional PCRLB. However, analytical computation of this new bound is, in general, intractable except when the system is linear and Gaussian. In this dissertation, we utilize a sequential Monte Carlo solution to compute the conditional PCRLB for nonlinear, non-Gaussian sequential Bayesian estimation problems.

In addition, a novel algorithm for tracking a moving target in a multi-sensor environment using distributed particle filters (DPFs) is presented. In a sensor network, the implementation of distributed particle filters requires huge amount of communications between local sensor nodes and the fusion center. To make the DPF approach feasible for real time processing and to reduce communication requirements, we approximate the *a posteriori* distribution obtained from the local particle filters by a Gaussian Mixture Model (GMM). We propose a modified EM algorithm to estimate the parameters of GMMs obtained locally. These parameters are transmitted to the fusion center where the Best Linear Unbiased Estimator (BLUE) is used for fusion. Simulation results are presented to illustrate the performance of the proposed algorithm.

Beyond that, tracking a moving target in heterogeneous sensor networks is investigated. Algorithms based on the classical extended Kalman filter (EKF) and on the emerging non-Gaussian and nonlinear particle filtering (PF) techniques have been implemented. These algorithms are tested in the practical case where a target maneuvers from time to time and an Interacting Multiple Model (IMM) framework is used.

1.2 Main Contributions

In this section, we present an outline of the significant contributions presented in the dissertation:

- A new notion of conditional PCRLB is proposed, which is conditioned on the actual past measurement realizations and is, therefore, more suitable for online adaptive sensor management. Derivation and implementation of the conditional PCRLB is also presented. Extensive comparison of the proposed conditional PCRLB and the conventional PCRLB is carried out.
- The conditional PCRLB is applied to the sensor selection problem for target tracking in sensor networks. A particle filter is used to estimate the moving target states as well as recursively computing the conditional PCRLB. Simulations for both analog and quantized measurement data are presented and compared to existing state of art approaches.
- A novel algorithm is developed for target tracking in sensor networks with distributed particle filters in order to save the communication bandwidth. Instead of transmitting the raw particles, we use a Gaussian Mixture Model (GMM) to approximate the *a posteriori* distribution obtained from the local particle filters, and only transmit the parameters of the GMM to the fusion center. The optimal rule based on the best linear unbiased estimation (BLUE) method is developed to fuse the GMM parameters collected from local sensors.
- Maneuvering target tracking with glint noise in heterogeneous sensor networks is investigated. Several algorithms are presented and compared in the practical case where a target maneuvers from time to time and an Interacting Multiple Model (IMM) framework is used.

1.3 Dissertation Organization

The rest of the dissertation is organized as follows. Chapter 2 describes some commonly used methods in estimation theory. Mainly, the maximum likelihood

method and the Bayesian method are discussed. Bayesian framework for dynamical systems is highlighted and optimal Bayesian estimation is presented. Optimal and sub-optimal approaches based on Kalman Filter theory and particle filter theory are presented as feasible estimation methods.

In Chapter 3, the new concept of conditional PCRLB is proposed. The exact conditional PCRLB and its recursive evaluation approach including an approximation are derived. Further, a general sequential Monte Carlo solution is proposed to compute the conditional PCRLB recursively for nonlinear non-Gaussian sequential Bayesian estimation problems. The differences between this new bound and existing measurement dependent PCRLBs are investigated and discussed. Illustrative examples are also provided to show the performance of the proposed conditional PCRLB.

Chapter 4 applies the conditional PCRLB to sensor selection problems for target tracking in sensor networks. Comparison between the conditional PCRLB and other existing approaches, including information-driven, PCRLB with renewal and nearest neighbor, are presented in terms of the simulation results.

Chapter 5 proposes a distributed target tracking algorithm based on particle filters for sensor networks. A modified EM algorithm is proposed to estimate the parameters of GMMs based on particles obtained locally. These parameters are transmitted to the fusion center where the Best Linear Unbiased Estimator (BLUE) is used for fusion. Simulation results are presented to illustrate the performance of the proposed algorithm.

Chapter 6 discusses and compares several target tracking algorithms in heterogeneous sensor networks.

Finally, Chapter 7 provides concluding remarks as well a summary of the work and discusses future research.

Chapter 2

Sequential Bayesian Estimation

2.1 Bayesian Estimation Problems

The sequential Bayesian estimation problem is to find the estimate of the state from the measurements (observations) over time. The evolution of the state sequence \mathbf{x}_k is assumed to be an unobserved first order Markov process, $\mathbf{x}_k | \mathbf{x}_{k-1} \sim p(\mathbf{x}_k | \mathbf{x}_{k-1})$, and is modeled as

$$\mathbf{x}_{k+1} = f_k(\mathbf{x}_k, \mathbf{u}_k) \quad (2.1)$$

where $f_k : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_x}$ is, in general, a nonlinear function of state \mathbf{x} , and $\{\mathbf{u}_k, k \in \{0\} \cup \mathbb{N}\}$ is an independent white process noise. n_u is the dimension of the noise vector \mathbf{u}_k . The probability density function (PDF) of the initial state \mathbf{x}_0 is assumed to be known.

The observations are conditionally independent provided that $\mathbf{x}_0, \mathbf{x}_1, \dots$ are known. So the measurement equation is modeled as

$$\mathbf{z}_k = h_k(\mathbf{x}_k, \mathbf{v}_k) \quad (2.2)$$

where $h_k : \mathbb{R}^{n_x} \times \mathbb{R}^{n_v} \rightarrow \mathbb{R}^{n_z}$ is, in general, a nonlinear function, $\{\mathbf{v}_k, k \in \mathbb{N}\}$ is the measurement noise sequence, which is independent of \mathbf{x}_k as well as \mathbf{u}_k . n_v is the dimension of the noise vector \mathbf{v}_k . Since process noise and measurement noise are assumed to be independent, \mathbf{x}_{k+1} is independent of $\mathbf{z}_{1:k} \triangleq \{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_k\}$ given \mathbf{x}_k , which means that $p(\mathbf{x}_{k+1} | \mathbf{x}_k, \mathbf{z}_{1:k}) = p(\mathbf{x}_{k+1} | \mathbf{x}_k)$.

If we denote the states and measurements up to time k as $\mathbf{x}_{0:k}$ and $\mathbf{z}_{1:k}$, then the joint PDF of $(\mathbf{x}_{0:k}, \mathbf{z}_{1:k})$ can be determined from (2.1) and (2.2) with known initial PDF $p(\mathbf{x}_0)$ and noise models for \mathbf{u}_k and \mathbf{v}_k

$$p(\mathbf{x}_{0:k}, \mathbf{z}_{1:k}) = p(\mathbf{x}_0) \prod_{i=1}^k p(\mathbf{x}_i | \mathbf{x}_{i-1}) \prod_{j=1}^k p(\mathbf{z}_j | \mathbf{x}_j) \quad (2.3)$$

Given the above Equations (2.1), (2.2) and (2.3), the Bayesian estimation problems can be classified into three categories:

- **Prediction** is an operation that involves the estimation of the state at time $k + \tau$ ($\tau > 0$) by using the observations up to and including time k . It is an *a priori* form of estimation in that the observed data up to time k is used for estimating the state at a future time.
- **Filtering** corresponds to estimating the distribution of the current state based upon the observations received up to and including k .
- **Smoothing** corresponds to estimating the distribution of the state at a particular time k' given the observations up to some later time k , where $k > k'$. It is an *a posteriori* form of estimation in that the data measured after the time of interest are used for estimation.

2.2 Bayesian Filtering

Bayesian filtering is aimed to apply Bayesian statistics and Bayes rule to probabilistic inference problems, and specifically the stochastic filtering problem. In the past few decades, numerous authors have investigated the Bayesian filtering problem in a dynamic state space framework [6][7][8]. This section provides a Bayesian view of the existing methods by focusing on the approach for solving the estimation problems mentioned in the last section.

An optimal filter is said to be optimal usually in some specific sense [9]. A criterion should be defined to measure the optimality. Here are some criteria for measuring the optimality:

- Minimum mean-squared error (MMSE): It can be defined in terms of prediction or filtering error (or equivalently the trace of the MSE matrix of the state).
- Maximum a posteriori (MAP): It is aimed to find the mode of posterior probability.
- Maximum likelihood (ML): which is a special case of MAP where the prior is neglected.

The criterion of optimality for Bayesian filtering is the Bayesian MMSE. However, except in some special cases (e.g. linear Gaussian or conjugate family case), the analytical form of the state distribution is not obtainable so that Bayesian MMSE is not tractable either. Therefore, in general we can only seek to obtain suboptimal solutions (e.g. Extended Kalman Filter, Particle Filter, etc.).

2.2.1 Kalman Filter

The Kalman filter [10], or Kalman-Bucy filter [11] is the optimal Bayesian filter for dynamic systems with linear state equation and additive Gaussian noise for both the process Equation (2.1) and measurement Equation (2.2). It is optimal not only in the sense of MMSE, but also the filtering result gives the exact distribution of state \mathbf{x}_k given the measurements up to the current time k .

Assuming the linearity and additive Gaussian noise properties, Equations (2.1) and (2.2) become:

$$\mathbf{x}_{k+1} = F_k \mathbf{x}_k + \mathbf{u}_k \tag{2.4}$$

$$\mathbf{z}_k = H_k \mathbf{x}_k + \mathbf{v}_k \tag{2.5}$$

where the dynamics noise \mathbf{u}_k is a white Gaussian noise process, $\mathbf{u}_k \sim \mathcal{N}\{\mathbf{0}, Q_k\}$, measurement noise \mathbf{v}_k is also white Gaussian noise process, and $\mathbf{v}_k \sim \mathcal{N}\{\mathbf{0}, R_k\}$, which is uncorrelated with \mathbf{u}_k . The initial state \mathbf{x}_0 is assumed to be known and distributed as $\mathbf{x}_0 \sim \mathcal{N}\{\hat{\mathbf{x}}_{0|0}, P_{0|0}\}$. $\hat{\mathbf{x}}_{0|0}$ and $P_{0|0}$ are the mean and covariance respectively for \mathbf{x}_0 .

The Kalman filter consists of an iterative prediction-update process. In the prediction step, the one-step ahead prediction of state is calculated:

$$\hat{\mathbf{x}}_{k|k-1} = F_{k-1}\hat{\mathbf{x}}_{k-1|k-1} \quad (2.6)$$

$$P_{k|k-1} = F_{k-1}P_{k-1|k-1}F_{k-1}^T + Q_{k-1} \quad (2.7)$$

where $\hat{\mathbf{x}}_{k|k-1}$ is defined as the estimate of \mathbf{x}_k conditioned on measurements up to and including time $k-1$, while $P_{k|k-1}$ is the covariance matrix of the corresponding prediction error.

In the update step, the predicted state estimate is updated according the new measurement at time k .

$$S_k = H_k P_{k|k-1} H_k^T + R_k \quad (2.8)$$

$$K_k = P_{k|k-1} H_k^T S_k^{-1} \quad (2.9)$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + K_k(\mathbf{z}_k - H_k \hat{\mathbf{x}}_{k-1|k-1}) \quad (2.10)$$

$$P_{k|k} = P_{k|k-1} - K_k H_k P_{k|k-1} \quad (2.11)$$

where $\hat{\mathbf{x}}_{k|k}$ is the updated state estimate and $P_{k|k}$ is the updated estimate covariance. S_k is the innovation(or residual) covariance:

$$S_k = cov(\tilde{\mathbf{y}}_k) \quad (2.12)$$

where $\tilde{\mathbf{y}}_k$ is the innovation(or measurement) residual, which is defined as $\tilde{\mathbf{y}}_k = \mathbf{z}_k - H_k \hat{\mathbf{x}}_{k-1|k-1}$.

For some reason, if the observation is unavailable, the update may be skipped and multiple prediction steps performed. Likewise, if multiple independent observations are available at the same time, multiple update steps may be performed.

In a stationary situation. Kalman filter is precisely the Wiener filter for stationary least-squares smoothing. In other words, Kalman filter is a time-variant Wiener filter. The reader is referred to [12] and [9] for more in-depth treatments.

2.2.2 Extended Kalman Filter

The Kalman filter is only applicable to linear systems with additive Gaussian noise. For nonlinear systems, the solution is that a mild nonlinearity may be approximated as being linear about a nominal point through the Taylor series expansion [13]. The mean $\hat{\mathbf{x}}_{k|k}$ and covariance $P_{k|k}$ of the Gaussian approximation to the posterior distribution of the states can be derived as follows:

$$\hat{\mathbf{x}}_{k|k-1} = f(\hat{\mathbf{x}}_{k-1|k-1}, 0) \quad (2.13)$$

$$P_{k|k-1} = F_{k-1}P_{k-1}F_{k-1}^T + Q_{k-1} \quad (2.14)$$

$$K_k = P_{k|k-1}H_k^T(H_kP_{k|k-1}H_k^T)^{-1} \quad (2.15)$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + K_k(y_k - h(\hat{\mathbf{x}}_{k|k-1}, 0)) \quad (2.16)$$

$$P_{k|k} = P_{k|k-1} - K_kH_kP_{k|k-1} \quad (2.17)$$

where K_k is the Kalman gain, and Jacobians of the process model and measurement model are given by

$$F_{k-1} \triangleq \left. \frac{\partial f(\mathbf{x}_{k-1})}{\partial \mathbf{x}_{k-1}} \right|_{(\mathbf{x}_{k-1} = \hat{\mathbf{x}}_{k-1|k-1})} \quad (2.18)$$

$$H_k \triangleq \left. \frac{\partial h(\mathbf{x}_k)}{\partial \mathbf{x}_k} \right|_{(\mathbf{x}_k = \hat{\mathbf{x}}_{k|k-1})} \quad (2.19)$$

It can be seen that, in EKF, the function f is used to compute the predicted state from the previous estimate and similarly the function h is used to compute the predicted measurement from the predicted state. However, f and h cannot be applied to compute the covariance directly. Instead, at each time step, the process linearizes the non-linear function f , which is evaluated at the updated state estimate $\mathbf{x}_{k-1} = \hat{\mathbf{x}}_{k-1|k-1}$ as shown in Equation (2.18). Equation (2.19) shows the linearization process for the non-linear function h with Jacobian matrix evaluated with predicted states $\mathbf{x}_k = \hat{\mathbf{x}}_{k|k-1}$. These Jacobians are used in the Kalman filter equations.

2.3 Bayesian Estimation through Sequential Monte Carlo Approach

2.3.1 Background

Monte Carlo methods are a class of computational algorithms that rely on statistical sampling and estimation techniques to evaluate the solutions to mathematical problems. Monte Carlo methods can be classified into three categories: (i) Monte Carlo sampling, which is aimed at developing efficient (variance-reduction oriented) sampling algorithms for parameter estimation; (ii) Monte Carlo calculation, which is a class of computational algorithms that rely on repeated random sampling to compute their results; and (iii) Monte Carlo optimization, which is devoted to applying the Monte Carlo idea to optimize some (non-convex or non-differentiable) functions. In last decades, modern Monte Carlo techniques have attracted more and more attention and have been developed in different areas. A detailed background of Monte Carlo methods can be obtained from the books [14][15] and survey papers, e.g., [16].

2.3.2 Sequential Monte Carlo Sampling for Bayesian Estimation

2.3.2.1 Monte Carlo Sampling

Monte Carlo sampling is a bunch of random number generation algorithms that use Monte Carlo Markov Chain (MCMC) methods to generate samples for a certain distribution, from which it is usually difficult to generate random samples directly. Monte Carlo sampling provides a convenient way to compute the properties of distributions, such as mean or variance of the random variable. Expectation based on Monte Carlo sampling may be expressed as [15]

$$E[f(x)] = \int f(x)p(x)dx \approx \frac{1}{N} \sum_{i=1}^N f(x_i) \quad (2.20)$$

By the law of large numbers, as the number of samples goes to infinity, this estimate approaches the true value. Due to the discrete nature of Monte Carlo

2.3 Bayesian Estimation through Sequential Monte Carlo Approach

sampling, it is difficult to obtain the probability distribution. A crude approximation in term of discrete distributions, useful for building intuition, may be written as,

$$p(x) \approx \sum_{i=1}^N w_i \delta(x - x_i) \quad (2.21)$$

where x_i is the i -th sample that approximates the distribution. The coefficient w_i is the probability associated with each sample.

There are two fundamental problems arising in Monte Carlo sampling methods: (i) How to draw random samples $\{x_i\}$ from a probability distribution $p(x)$? and (ii) How to estimate the expectation of a function w.r.t. the distribution or density? The first problem is a design problem, and the second one is an inference problem invoking integration. Besides, there are several relevant issues that need to be considered in the context of parameter estimation:

- Consistency: An estimator is consistent if the estimator converges to the true value almost surely as the number of observations approaches infinity.
- Unbiasedness: An estimator is unbiased if its expected value is equal to the true value.
- Efficiency: An estimator is efficient if it produces the smallest error covariance matrix among all unbiased estimators, and it is also regarded as the one optimally using the information in the measurements. A well-known efficiency criterion is the Cramér-Rao bound.
- Robustness: An estimator is robust if it is insensitive to the gross measurement errors and the uncertainties of the model.
- Minimal variance: Variance reduction is the central issue of various Monte Carlo approximation methods, most improvement techniques are variance-reduction oriented.

2.3 Bayesian Estimation through Sequential Monte Carlo Approach

2.3.2.2 Importance Sampling

Importance sampling (IS) was first introduced by Marshall [17] and is a general technique for estimating the expectations of a particular function of a random variable or random vector, while samples are generated from a different distribution rather than the distribution of interest. The idea of importance sampling is to choose a proposal distribution $q(x)$ in place of the true probability distribution $p(x)$, which is hard-to-sample. The support of $q(x)$ is assumed to cover that of $p(x)$. The objective of importance sampling is aimed to sample the distribution in the region of importance in order to achieve computational efficiency.

Importance sampling is useful in two ways [15]: 1) it can be used when encountering the difficulty to sample from the true distribution directly; and 2) it provides an elegant way to reduce the variance of the estimator. The idea is that certain values of the input random variables in a simulation have more impact on the parameter being estimated than others. If these “important” values are emphasized by sampling more frequently, then the estimator variance can be reduced. Hence, the basic methodology in importance sampling is to choose a distribution which “encourages” the important values.

2.3.2.3 Sequential Importance Sampling

For the sequential estimation problems, a good proposal distribution is to construct the proposal distribution sequentially, which is the basic idea of sequential importance sampling (SIS). In particular, if the proposal distribution is chosen in a factorized form

$$q(x_{0:k}|z_{1:k}) = q(x_0)\prod_{t=1}^k q(x_t|x_{0:t-1}, z_{1:t}) \quad (2.22)$$

then the importance sampling can be performed recursively.

There are a number of potential problems with the SIS algorithm. One problem is sample impoverishment or weight degeneracy, which is a phenomenon where after a few iterations, the weights of most samples become insignificant, while only a few samples start to dominate the distribution [18]. Consequently, most samples have no influence on the posterior and distributions are then determined only by a few samples. This implies that a large computational resources will be wasted

2.3 Bayesian Estimation through Sequential Monte Carlo Approach

to update particles whose contribution has little or no relevance to the approximation. Therefore, this phenomenon may weaken the successful application of Monte Carlo sampling which relies on diversity of the samples. Degeneracy could be observed by monitoring the variance of the samples' weights.

Degeneracy is inevitable in SMC unless importance functions are selected such that the variance of samples' weights is minimized. In order to alleviate this problem, the importance density function has to be chosen very carefully. Another convenient approach to avoid degeneracy is to implement resampling whenever degeneracy develops. This approach involves drawing samples from the weighted sample pool. However, resampling consumes a substantial computation power.

2.3.2.4 Sampling Importance Resampling

The sampling-importance resampling (SIS) is an approach to avoid the problem of degeneracy. Resampling can be taken at every step or only taken if regarded necessary. In the resampling step, the particles and associated importance weights $\{x_i, \bar{w}_i\}$ are replaced by the new samples with equal importance weights (i.e. $\bar{w}_i = 1/N$). There are many types of resampling methods available in the literature:

- Multinomial resampling: Multinomial resampling uniformly generates N new independent particles from the old particle set.
- Residual resampling: Residual resampling, or remainder resampling is an efficient means to decrease the variance due to resampling. Residual resampling procedure is computationally cheaper than the conventional SIR, and it does not introduce additional bias.
- Systematic resampling: Systematic resampling treats the weights as continuous random variables in the interval $(0, 1)$, which are randomly ordered.
- Local Monte Carlo resampling: The samples are redrawn using rejection method or Metropolis-Hastings method.

- Stratified Sampling: The idea of stratified sampling is to distribute the samples evenly (or unevenly according to their respective variance) to the subregions dividing the whole space.

2.4 Discussion

In this dissertation, SIS particle filter is used extensively for all the target tracking problems. We also use particle filter to obtain the numerical values of the conditional posterior Cramér-Rao lower bound in Chapter 3. Extended Kalman Filter is used for comparison purpose. For most problems in this dissertation, due to the nonlinearity property, the particle filter yields more accurate simulation results than the extended Kalman Filter.

Chapter 3

Conditional Posterior Cramér-Rao Lower Bounds for Sequential Bayesian Estimation

3.1 Motivation

The conventional Cramér-Rao lower bound (CRLB) [19] on the variance of estimation error provides the performance limit for any unbiased estimator of a fixed parameter. For a random parameter, Van Trees presented an analogous bound, the posterior CRLB (PCRLB) [19], which is also referred to as the Bayesian CRLB. The PCRLB is defined as the inverse of the Fisher information matrix (FIM) for a random vector and provides a lower bound on the mean squared error (MSE) of any estimator of the random parameter, which in general is a vector. In [5], Tichavsky et al. derived an elegant recursive approach to calculate the sequential PCRLB for a general multi-dimensional discrete-time nonlinear filtering problem.

The PCRLB is a very important tool, since it provides a theoretical performance limit of any estimator for a nonlinear filtering problem under the Bayesian framework. In an unconditional PCRLB, the FIM is derived by taking the expectation with respect to the joint distribution of the measurements and the system states up to the current time. As a result, the very useful measurement information is averaged out and the unconditional PCRLB becomes an off-line bound.

It is determined only by the system dynamic model, system measurement model and the prior knowledge regarding the system state at the initial time, and is thus independent of any specific realization of the system state, as we will show later in the chapter. As a result, the unconditional PCRLB does not reflect the nonlinear filtering performance for a particular system state realization very faithfully. This is especially true when the uncertainty in the state model (or equivalently the state process noise) is high and thus the prior knowledge regarding the system state at the initial time quickly becomes irrelevant as the system state evolves over time.

Some attempts have been made in the literature to include the information obtained from measurements by incorporating the tracker's information into the calculation of the PCRLB. In [20], a renewal strategy has been used to restart the recursive unconditional PCRLB evaluation process, where the initial time is reset to a more recent past time, so that the prior knowledge of the initial system state is more useful and relevant to the sensor management problem. The resulting PCRLB is, therefore, conditioned on the measurements up to the reset initial time. Based on the PCRLB evaluated in this manner, a sensor deployment approach is developed to achieve better tracking accuracy which at the same time uses the limited sensor resources more efficiently. This approach is extended in [21] to incorporate sensor deployment and motion uncertainties, and to manage sensor arrays for multi-target tracking problems in [22, 23]. In the renewal strategy proposed in [20], using a particle filter, the posterior probability density function (PDF) of the system state at the reset initial time is represented non-parametrically by a set of random samples (particles), from which it is difficult to derive the exact Fisher information matrix. One solution is to use a Gaussian approximation, and in this case the FIM at the reset initial time can be taken as the inverse of the empirical covariance matrix estimated based on the particles. This, however, may incur large errors and discrepancy, especially in a highly nonlinear and non-Gaussian system. Once restarted, the renewal based approach recursively evaluates the PCRLB as provided in [5] till the next restart. Since the FIM at the reset initial time is evaluated based on filtering results rather than the previous FIM, this is not an entirely recursive approach. In contrast, in this chapter, we introduce the notion of conditional PCRLB, which is shown to be

different from the PCRLB based on renewal strategy presented in [20], through analysis and numerical examples. A systematic recursive approach to evaluate the conditional PCRLB with approximation is also presented.

Another related work is reported in [24], where a PCRLB based adaptive radar waveform design method for target tracking has been presented. In [24], for a system with a linear and Gaussian state dynamic model, but nonlinear measurement model, the framework of the unconditional recursive PCRLB derived in [5] has been retained. Only one term corresponding to the contribution of the future measurement to the Fisher information matrix (FIM) has been modified in an ad-hoc manner to include the measurement history, by taking the expectation of the second-order derivative of the log-likelihood function with respect to the joint probability density function (PDF) of the state and measurement at the next time step conditioned on the measurements up to the current time. The heuristically modified PCRLB calculated in this manner does not yield the exact conditional PCRLB, as shown later in this chapter.

In [25], for nonlinear target tracking problems, an algorithm is developed to select and configure radar waveforms to minimize the predicted MSE in the target state estimate, which is the expectation of the squared error over predicted states and observations given a past history of measurements. The predicted MSE is computationally intractable, and in [25] it has been approximated by the covariance update of the unscented Kalman filter.

Given the importance of the PCRLB based adaptive sensor management problem, to take advantage of the available measurement information, we have systematically developed the exact conditional PCRLB based on first principles. The proposed conditional PCRLB is dependent on the past data and hence implicitly dependent on the system state. The conditional PCRLB provides a bound on the conditional MSE of the system state estimate, based on the measurements up to the current time. In this chapter, we systematically derive an approximate recursive formula to calculate the conditional PCRLB for nonlinear/non-Gaussian Bayesian estimation problems. The cumulative error due to the approximation is not severe even for a highly nonlinear problem, as demonstrated in a simulation example. Further, we present numerical approximation approaches for the computation of the recursive formula through particle filters. Since the conditional

PCRLB is a function of the past history of measurements, which contains the information of the current realization of the system state, an approach based on it is expected to lead to much better solutions to the sensor resource management problem than those based on the unconditional PCRLB.

3.2 Classical Cramer-Rao Lower Bounds

We are interested in estimating the state \mathbf{x} given the observation \mathbf{z} , where \mathbf{x} and \mathbf{z} are both random vectors with dimensions n_x and n_z respectively, $n_x, n_z \in \mathbb{N}$, and \mathbb{N} is the set of natural numbers. Let $\hat{\mathbf{x}}(\mathbf{z})$ be an estimator of \mathbf{x} , which is a function of \mathbf{z} . The Bayesian Cramér-Rao inequality [19] shows that the mean squared error (MSE) of any estimator can not go below a bound, which is given by

$$E \{ [\hat{\mathbf{x}}(\mathbf{z}) - \mathbf{x}][\hat{\mathbf{x}}(\mathbf{z}) - \mathbf{x}]^T \} \geq J^{-1} \quad (3.1)$$

where J is the Fisher information matrix

$$J = E \{ -\Delta_{\mathbf{x}}^{\mathbf{x}} \log p(\mathbf{x}, \mathbf{z}) \} \quad (3.2)$$

and the expectation is taken with respect to $p(\mathbf{x}, \mathbf{z})$, which is the joint PDF of the pair (\mathbf{x}, \mathbf{z}) . Δ denotes the second-order derivative operator, namely

$$\Delta_{\mathbf{x}}^{\mathbf{y}} = \nabla_{\mathbf{x}} \nabla_{\mathbf{y}}^T \quad (3.3)$$

in which ∇ denotes the gradient operator. Unbiasedness of the estimator $\hat{\mathbf{x}}$ is not required for the Bayesian CRLB. The mild conditions and proof of this inequality can be found in [19].

The sequential Bayesian estimation problem is to find the estimate of the state from the measurements (observations) over time. The evolution of the state sequence \mathbf{x}_k is assumed to be an unobserved first order Markov process, and is modeled as

$$\mathbf{x}_{k+1} = f_k(\mathbf{x}_k, \mathbf{u}_k) \quad (3.4)$$

where $f_k : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_x}$ is, in general, a nonlinear function of state \mathbf{x} , and $\{\mathbf{u}_k, k \in \{0\} \cup \mathbb{N}\}$ is an independent white process noise. n_u is the dimension of

3.2 Classical Cramer-Rao Lower Bounds

the noise vector \mathbf{u}_k . The PDF of the initial state \mathbf{x}_0 is assumed to be known. The observations about the state are obtained from the measurement equation

$$\mathbf{z}_k = h_k(\mathbf{x}_k, \mathbf{v}_k) \quad (3.5)$$

where $h_k : \mathbb{R}^{n_x} \times \mathbb{R}^{n_v} \rightarrow \mathbb{R}^{n_z}$ is, in general, a nonlinear function, $\{\mathbf{v}_k, k \in \mathbb{N}\}$ is the measurement noise sequence, which is independent of \mathbf{x}_k as well as \mathbf{u}_k . n_v is the dimension of the noise vector \mathbf{v}_k . Since process noise and measurement noise are assumed to be independent, \mathbf{x}_{k+1} is independent of $\mathbf{z}_{1:k}$ given \mathbf{x}_k , which means that $p(\mathbf{x}_{k+1} | \mathbf{x}_k, \mathbf{z}_{1:k}) = p(\mathbf{x}_{k+1} | \mathbf{x}_k)$.

If we denote the states and measurements up to time k as $\mathbf{x}_{0:k}$ and $\mathbf{z}_{1:k}$, then the joint PDF of $(\mathbf{x}_{0:k}, \mathbf{z}_{1:k})$ can be determined from (3.4) and (3.5) with known initial PDF $p(\mathbf{x}_0)$ and noise models for \mathbf{u}_k and \mathbf{v}_k

$$p(\mathbf{x}_{0:k}, \mathbf{z}_{1:k}) = p(\mathbf{x}_0) \prod_{i=1}^k p(\mathbf{x}_i | \mathbf{x}_{i-1}) \prod_{j=1}^k p(\mathbf{z}_j | \mathbf{x}_j) \quad (3.6)$$

If we consider $\mathbf{x}_{0:k}$ as a vector with dimension $(k+1)n_x$, and define $J(\mathbf{x}_{0:k})$ to be the $(k+1)n_x \times (k+1)n_x$ Fisher information matrix of $\mathbf{x}_{0:k}$ derived from the joint PDF $p(\mathbf{x}_{0:k}, \mathbf{z}_{1:k})$, (3.1) becomes

$$E \{ [\hat{\mathbf{x}}_{0:k}(\mathbf{z}_{1:k}) - \mathbf{x}_{0:k}] [\hat{\mathbf{x}}_{0:k}(\mathbf{z}_{1:k}) - \mathbf{x}_{0:k}]^T \} \geq J^{-1}(\mathbf{x}_{0:k}) \quad (3.7)$$

Let us define J_k as the matrix whose inverse equals the $n_x \times n_x$ lower-right corner submatrix of $J^{-1}(\mathbf{x}_{0:k})$. Then, the MSE of the estimate for \mathbf{x}_k is bounded by J_k^{-1} .

J_k can be obtained directly from the computed inverse of the $(k+1)n_x \times (k+1)n_x$ matrix $J(\mathbf{x}_{0:k})$. However, this is not an efficient approach. In [5], Tichavsky et al. provide an elegant recursive approach to calculate J_k without manipulating the large matrices at each time k

$$J_{k+1} = D_k^{22} - D_k^{21} (J_k + D_k^{11})^{-1} D_k^{12} \quad (3.8)$$

where

$$D_k^{11} = E \{ -\Delta_{\mathbf{x}_k}^{\mathbf{x}_k} \log p(\mathbf{x}_{k+1} | \mathbf{x}_k) \} \quad (3.9)$$

$$D_k^{12} = E\{-\Delta_{\mathbf{x}_k}^{\mathbf{x}_{k+1}} \log p(\mathbf{x}_{k+1}|\mathbf{x}_k)\} = (D_k^{21})^T \quad (3.10)$$

$$\begin{aligned} D_k^{22} &= E\left\{-\Delta_{\mathbf{x}_{k+1}}^{\mathbf{x}_{k+1}} [\log p(\mathbf{x}_{k+1}|\mathbf{x}_k) + \log p(\mathbf{z}_{k+1}|\mathbf{x}_{k+1})]\right\} \\ &= D_k^{22,a} + D_k^{22,b} \end{aligned} \quad (3.11)$$

Conventional PCRLB considers the measurements as random vectors, and at any particular time k , the bound is calculated by taking the average of both the measurements and the states up to time k . In many cases, besides the two system equations, some of the measurements are available, for example, the measurements up to time $k - 1$, $\mathbf{z}_{1:k-1}$. In this chapter, we introduce the notion of conditional PCRLB, which utilizes the information contained in the available measurements. The proposed bound is an online bound, and it gives us more accurate indication on the performance of the estimator at the upcoming time than the conventional PCRLB.

3.3 Conditional PCRLB for Nonlinear Dynamical Systems

The conditional PCRLB sets a bound on the performance of estimating $\mathbf{x}_{0:k+1}$ when the new measurement \mathbf{z}_{k+1} becomes available given that the past measurements up to time k are all known. Here the measurements up to time k are taken as realizations rather than random vectors.

Definition 1 *Conditional estimator $\hat{\mathbf{x}}_{0:k+1}(\mathbf{z}_{k+1}|\mathbf{z}_{1:k})$ is defined as a function of the observed data \mathbf{z}_{k+1} given the existing measurements $\mathbf{z}_{1:k}$.*

Definition 2 *Mean squared error of the conditional estimator at time $k + 1$ is defined as follows*

$$\begin{aligned} MSE(\hat{\mathbf{x}}_{0:k+1}|\mathbf{z}_{1:k}) &\triangleq E\{\tilde{\mathbf{x}}_{0:k+1}\tilde{\mathbf{x}}_{0:k+1}^T|\mathbf{z}_{1:k}\} \\ &= \int \tilde{\mathbf{x}}_{0:k+1}\tilde{\mathbf{x}}_{0:k+1}^T p_{k+1}^c d\mathbf{x}_{0:k+1} d\mathbf{z}_{k+1} \end{aligned} \quad (3.12)$$

where $\tilde{\mathbf{x}}_{0:k+1} \triangleq \hat{\mathbf{x}}_{0:k+1} - \mathbf{x}_{0:k+1}$ is the estimation error, and $p_{k+1}^c \triangleq p(\mathbf{x}_{0:k+1}, \mathbf{z}_{k+1}|\mathbf{z}_{1:k})$.

3.3 Conditional PCRLB for Nonlinear Dynamical Systems

Definition 3 Let $I(\mathbf{x}_{0:k+1}|\mathbf{z}_{1:k})$ be the $(k+2)n_x \times (k+2)n_x$ conditional Fisher information matrix of the state vector $\mathbf{x}_{0:k+1}$ from time 0 to $k+1$:

$$\begin{aligned} & I(\mathbf{x}_{0:k+1}|\mathbf{z}_{1:k}) & (3.13) \\ \triangleq & E \left\{ - \left[\Delta_{\mathbf{x}_{0:k+1}}^{\mathbf{x}_{0:k+1}} \log p_{k+1}^c \right] \middle| \mathbf{z}_{1:k} \right\} \\ = & - \int \left[\Delta_{\mathbf{x}_{0:k+1}}^{\mathbf{x}_{0:k+1}} \log p_{k+1}^c \right] \times p_{k+1}^c d\mathbf{x}_{0:k+1} d\mathbf{z}_{k+1} \end{aligned}$$

With the above definitions, we give the conditional posterior CRLB inequality.

Proposition 1 The conditional mean squared error of the state vector $\mathbf{x}_{0:k+1}$ is lower bounded by the inverse of the conditional Fisher information matrix

$$E \left\{ \tilde{\mathbf{x}}_{0:k+1} \tilde{\mathbf{x}}_{0:k+1}^T \middle| \mathbf{z}_{1:k} \right\} \geq I^{-1}(\mathbf{x}_{0:k+1}|\mathbf{z}_{1:k}) \quad (3.14)$$

The proof of Proposition 1 is similar to the one for the unconditional PCRLB presented in [19]. See appendix A for details.

Definition 4 $L(\mathbf{x}_{k+1}|\mathbf{z}_{1:k})$ is defined as the conditional Fisher information matrix for estimating \mathbf{x}_{k+1} , and $L^{-1}(\mathbf{x}_{k+1}|\mathbf{z}_{1:k})$ is equal to the $n_x \times n_x$ lower-right block of $I^{-1}(\mathbf{x}_{0:k+1}|\mathbf{z}_{1:k})$.

By definition, $L^{-1}(\mathbf{x}_{k+1}|\mathbf{z}_{1:k})$ is a bound on the MSE of the estimate for \mathbf{x}_{k+1} given $\mathbf{z}_{1:k}$. At time k , the conditional PCRLB, $L^{-1}(\mathbf{x}_{k+1}|\mathbf{z}_{1:k})$, provides a predicted estimator performance limit for the upcoming time $k+1$, given the measurements up to time k . Therefore, it is very useful for the sensor/resource management for target tracking in sensor networks[26, 27]. Here, we propose an iterative approach to calculate $L^{-1}(\mathbf{x}_{k+1}|\mathbf{z}_{1:k})$ without manipulating the large matrix $I(\mathbf{x}_{0:k+1}|\mathbf{z}_{1:k})$. This iterative approach is facilitated by an auxiliary FIM, which is defined below.

Definition 5 The auxiliary Fisher information matrix for the state vector from time 0 to k is defined as

$$\begin{aligned} & I_A(\mathbf{x}_{0:k}|\mathbf{z}_{1:k}) & (3.15) \\ \triangleq & E_{p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})} \left\{ - \Delta_{\mathbf{x}_{0:k}}^{\mathbf{x}_{0:k}} \log p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k}) \right\} \\ = & - \int \left[\Delta_{\mathbf{x}_{0:k}}^{\mathbf{x}_{0:k}} \log p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k}) \right] p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k}) d\mathbf{x}_{0:k} \end{aligned}$$

3.3 Conditional PCRLB for Nonlinear Dynamical Systems

Definition 6 We define $L_A(\mathbf{x}_k|\mathbf{z}_{1:k})$ as the auxiliary Fisher information matrix for \mathbf{x}_k , and $L_A^{-1}(\mathbf{x}_k|\mathbf{z}_{1:k})$ is equal to the $n_x \times n_x$ lower-right block of $I_A^{-1}(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})$.

The matrix inversion formula [28] is heavily used for deriving the recursive version of the conditional PCRLB. We include it here for completeness

$$\begin{bmatrix} A & B \\ B^T & C \end{bmatrix}^{-1} = \begin{bmatrix} D^{-1} & -A^{-1}BE^{-1} \\ -E^{-1}B^TA^{-1} & E^{-1} \end{bmatrix} \quad (3.16)$$

where A, B and C are sub-matrices with appropriate dimensions, and $D = A - BC^{-1}B^T$, $E = C - B^TA^{-1}B$.

By definition, the inverse of $L(\mathbf{x}_{k+1}|\mathbf{z}_{1:k})$ is the lower-right block of $I^{-1}(\mathbf{x}_{0:k+1}|\mathbf{z}_{1:k})$. Instead of calculating $I^{-1}(\mathbf{x}_{0:k+1}|\mathbf{z}_{1:k})$ directly, the following theorem gives a simple approach for computing $L(\mathbf{x}_{k+1}|\mathbf{z}_{1:k})$.

Theorem 1 The sequence of conditional Fisher information $\{L(\mathbf{x}_{k+1}|\mathbf{z}_{1:k})\}$ for estimating state vectors $\{\mathbf{x}_{k+1}\}$ can be computed as follows

$$L(\mathbf{x}_{k+1}|\mathbf{z}_{1:k}) = B_k^{22} - B_k^{21} [B_k^{11} + L_A(\mathbf{x}_k|\mathbf{z}_{1:k})]^{-1} B_k^{12} \quad (3.17)$$

where

$$B_k^{11} = E_{p_{k+1}^c} \left\{ -\Delta_{\mathbf{x}_k}^{\mathbf{x}_{k+1}} \log p(\mathbf{x}_{k+1}|\mathbf{x}_k) \right\} \quad (3.18)$$

$$B_k^{12} = E_{p_{k+1}^c} \left\{ -\Delta_{\mathbf{x}_k}^{\mathbf{x}_{k+1}} \log p(\mathbf{x}_{k+1}|\mathbf{x}_k) \right\} = (B_k^{21})^T \quad (3.19)$$

$$B_k^{22} = E_{p_{k+1}^c} \left\{ -\Delta_{\mathbf{x}_{k+1}}^{\mathbf{x}_{k+1}} [\log p(\mathbf{x}_{k+1}|\mathbf{x}_k) + \log p(\mathbf{z}_{k+1}|\mathbf{x}_{k+1})] \right\} \quad (3.20)$$

Proof: The conditional Fisher information matrix can be decomposed as follows

$$\begin{aligned} I(\mathbf{x}_{0:k+1}|\mathbf{z}_{1:k}) &= E_{p_{k+1}^c} (-1) \begin{bmatrix} \Delta_{\mathbf{x}_{0:k-1}}^{\mathbf{x}_{0:k-1}} & \Delta_{\mathbf{x}_{0:k-1}}^{\mathbf{x}_k} & \Delta_{\mathbf{x}_{0:k-1}}^{\mathbf{x}_{k+1}} \\ \Delta_{\mathbf{x}_k}^{\mathbf{x}_{0:k-1}} & \Delta_{\mathbf{x}_k}^{\mathbf{x}_k} & \Delta_{\mathbf{x}_k}^{\mathbf{x}_{k+1}} \\ \Delta_{\mathbf{x}_{k+1}}^{\mathbf{x}_{0:k-1}} & \Delta_{\mathbf{x}_{k+1}}^{\mathbf{x}_k} & \Delta_{\mathbf{x}_{k+1}}^{\mathbf{x}_{k+1}} \end{bmatrix} \log p_{k+1}^c \\ &= \begin{bmatrix} A_k^{11} & A_k^{12} & \mathbf{0} \\ A_k^{21} & A_k^{22} + B_k^{11} & B_k^{12} \\ \mathbf{0} & B_k^{21} & B_k^{22} \end{bmatrix} \end{aligned} \quad (3.21)$$

3.3 Conditional PCRLB for Nonlinear Dynamical Systems

where

$$\begin{aligned} A_k^{11} &= E_{p_{k+1}^c} \left[-\Delta_{\mathbf{x}_{0:k-1}}^{\mathbf{x}_{0:k-1}} \log p_{k+1}^c \right] \\ &= E_{p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})} \left[-\Delta_{\mathbf{x}_{0:k-1}}^{\mathbf{x}_{0:k-1}} \log p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k}) \right] \end{aligned} \quad (3.22)$$

In a similar manner, A_k^{12} can be derived as

$$A_k^{12} = E_{p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})} \left[-\Delta_{\mathbf{x}_{0:k-1}}^{\mathbf{x}_k} \log p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k}) \right] = (A_k^{21})^T \quad (3.23)$$

$$A_k^{22} + B_k^{11} = E_{p_{k+1}^c} \left[-\Delta_{\mathbf{x}_k}^{\mathbf{x}_k} \log p_{k+1}^c \right] \quad (3.24)$$

where

$$A_k^{22} = E_{p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})} \left[-\Delta_{\mathbf{x}_k}^{\mathbf{x}_k} \log p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k}) \right] \quad (3.25)$$

and B_k^{11} has been defined in (3.18). The conditional Fisher information matrix $L(\mathbf{x}_{k+1}|\mathbf{z}_{1:k})$ is equal to the inverse of the lower-right sub-matrix of $I^{-1}(\mathbf{x}_{0:k+1}|\mathbf{z}_{1:k})$. So

$$\begin{aligned} &L(\mathbf{x}_{k+1}|\mathbf{z}_{1:k}) \quad (3.26) \\ &= B_k^{22} - \begin{bmatrix} \mathbf{0} & B_k^{21} \end{bmatrix} \begin{bmatrix} A_k^{11} & A_k^{12} \\ A_k^{21} & A_k^{22} + B_k^{11} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{0} \\ B_k^{12} \end{bmatrix} \\ &= B_k^{22} - B_k^{21} [B_k^{11} + L_A(\mathbf{x}_k|\mathbf{z}_{1:k})]^{-1} B_k^{12} \end{aligned}$$

where

$$L_A(\mathbf{x}_k|\mathbf{z}_{1:k}) = A_k^{22} - A_k^{21} (A_k^{11})^{-1} A_k^{12} \quad (3.27)$$

Q.E.D.

Theorem 1 indicates that the conditional Fisher information at the current time step, $L(\mathbf{x}_{k+1}|\mathbf{z}_{1:k})$, can not be directly calculated from that at the previous time step, $L(\mathbf{x}_k|\mathbf{z}_{1:k-1})$. Instead, its evaluation has to be facilitated by the auxiliary Fisher information $L_A(\mathbf{x}_k|\mathbf{z}_{1:k})$. This implies that the heuristically modified conditional PCRLB presented in [24], which has a direct recursion from $L(\mathbf{x}_k|\mathbf{z}_{1:k-1})$ to $L(\mathbf{x}_{k+1}|\mathbf{z}_{1:k})$, does not yield the exact conditional PCRLB as provided in Definitions 3 and 4.

3.3 Conditional PCRLB for Nonlinear Dynamical Systems

In Theorem 1, a recursive approach is provided to predict the performance of the nonlinear filter at the next time step, based on the measurements up to the current time. Now let us investigate the relationship between the conditional PCRLB presented in Theorem 1 and the unconditional PCRLB with renewal strategy proposed in [20]. In the unconditional PCRLB with renewal strategy, the counterpart of the one-step-ahead conditional PCRLB works as follows. At each time k , the system prior PDF is re-initialized with the posterior PDF $p_0(\mathbf{x}_k) = p(\mathbf{x}_k|\mathbf{z}_{1:k})$. Accordingly, $E\{-\Delta_{\mathbf{x}_k}^{\mathbf{x}_k} \log p(\mathbf{x}_k|\mathbf{z}_{1:k})\}$ takes the place of J_k in (3.8). The Fisher information J_{k+1} at time $k+1$ is then calculated by one-step recursion using Eqs. (3.8) through (3.11), where the expectations are taken with respect to $p(\mathbf{x}_{k:k+1}, \mathbf{z}_{k+1}|\mathbf{z}_{1:k})$.

We summarize the relationship between the one-step ahead conditional PCRLB and the recursive unconditional PCRLB that renews its prior at each time in the following Lemma.

Lemma 1 *The conditional Fisher information matrix $L(\mathbf{x}_{k+1}|\mathbf{z}_{1:k})$ provided in Theorem 1 is different from J_{k+1} , calculated by one-step recursion using Eqs. (3.8) through (3.11) and setting the system state prior PDF $p_0(\mathbf{x}_k)$ as $p(\mathbf{x}_k|\mathbf{z}_{1:k})$, provided that $L_A(\mathbf{x}_k|\mathbf{z}_{1:k})$ is different from \tilde{J}_k , which is defined as $E_{p(\mathbf{x}_k|\mathbf{z}_{1:k})}\{-\Delta_{\mathbf{x}_k}^{\mathbf{x}_k} \log p(\mathbf{x}_k|\mathbf{z}_{1:k})\}$.*

Proof: In the recursive unconditional PCRLB that renews its prior at each time, according to (3.8),

$$J_{k+1} = D_k^{22} - D_k^{21}(\tilde{J}_k + D_k^{11})^{-1}D_k^{12} \quad (3.28)$$

where

$$\tilde{J}_k = E_{p(\mathbf{x}_k|\mathbf{z}_{1:k})}\{-\Delta_{\mathbf{x}_k}^{\mathbf{x}_k} \log p(\mathbf{x}_k|\mathbf{z}_{1:k})\} \quad (3.29)$$

Based on Theorem 1, the conditional FIM is given as

$$L(\mathbf{x}_{k+1}|\mathbf{z}_{1:k}) = B_k^{22} - B_k^{21} [B_k^{11} + L_A(\mathbf{x}_k|\mathbf{z}_{1:k})]^{-1} B_k^{12} \quad (3.30)$$

Since in the unconditional PCRLB that renews its prior at each time, the expectations are taken with respect to $p(\mathbf{x}_{k:k+1}, \mathbf{z}_{k+1}|\mathbf{z}_{1:k})$. According to (3.9), it is

3.3 Conditional PCRLB for Nonlinear Dynamical Systems

easy to show that

$$\begin{aligned}
 D_k^{11} &= E_{p(\mathbf{x}_{k:k+1}, \mathbf{z}_{k+1} | \mathbf{z}_{1:k})} \{ -\Delta_{\mathbf{x}_k}^{\mathbf{x}_k} \log p(\mathbf{x}_{k+1} | \mathbf{x}_k) \} \\
 &= E_{p(\mathbf{x}_{0:k+1}, \mathbf{z}_{k+1} | \mathbf{z}_{1:k})} \{ -\Delta_{\mathbf{x}_k}^{\mathbf{x}_k} \log p(\mathbf{x}_{k+1} | \mathbf{x}_k) \} \\
 &= B_k^{11}
 \end{aligned} \tag{3.31}$$

Similarly, it can be proved that $B_k^{12} = D_k^{12}$, $B_k^{21} = D_k^{21}$, and $B_k^{22} = D_k^{22}$. The right hand sides of (3.28) and (3.30) differ by only one term, which is either $L_A(\mathbf{x}_k | \mathbf{z}_{1:k})$ or \tilde{J}_k . Hence, if $L_A(\mathbf{x}_k | \mathbf{z}_{1:k})$ is different from \tilde{J}_k , in general, the conditional Fisher information matrix $L(\mathbf{x}_{k+1} | \mathbf{z}_{1:k})$ is different from J_{k+1} , which is calculated using the unconditional PCRLB that renews its prior at each time. Q.E.D.

The auxiliary Fisher information matrix has been defined in a way such that its inverse, $L_A^{-1}(\mathbf{x}_k | \mathbf{z}_{1:k})$, is equal to the $n_x \times n_x$ lower-right block of $[E_{p(\mathbf{x}_{0:k} | \mathbf{z}_{1:k})} \{ -\Delta_{\mathbf{x}_{0:k}}^{\mathbf{x}_{0:k}} \log p(\mathbf{x}_{0:k} | \mathbf{z}_{1:k}) \}]^{-1}$. It can be shown that in a linear and Gaussian system, $L_A(\mathbf{x}_k | \mathbf{z}_{1:k})$ and \tilde{J}_k are equivalent, so that the conditional PCRLB and the unconditional PCRLB that renews its prior at each time are equivalent. For nonlinear/non-Gaussian systems, the calculation of $L_A(\mathbf{x}_k | \mathbf{z}_{1:k})$ and \tilde{J}_k involves complex integrations and analytical results are intractable in general. Hence, direct comparison is very difficult. However, we demonstrate their difference through a simulation for a particular nonlinear system. The results are shown in Experiment V in Section 3.5. From the numerical results, we can see that $L_A(\mathbf{x}_k | \mathbf{z}_{1:k})$ is not equal to $\tilde{J}_k = E\{-\Delta_{\mathbf{x}_k}^{\mathbf{x}_k} \log p(\mathbf{x}_k | \mathbf{z}_{1:k})\}$. This in turn implies that $L(\mathbf{x}_{k+1} | \mathbf{z}_{1:k})$ and J_{k+1} are different in general.

One problem that is left in the proof of Theorem 1 is the inverse of the auxiliary Fisher Information matrix, $L_A^{-1}(\mathbf{x}_k | \mathbf{z}_{1:k})$, which is equal to the $n_x \times n_x$ lower-right block of $I_A^{-1}(\mathbf{x}_{0:k} | \mathbf{z}_{1:k})$. Direct computation of $L_A(\mathbf{x}_k | \mathbf{z}_{1:k})$ involves the inverse of the matrix $I_A(\mathbf{x}_{0:k} | \mathbf{z}_{1:k})$ of size $(k+1)n_x \times (k+1)n_x$. Therefore, we provide a recursive method for computing $L_A(\mathbf{x}_k | \mathbf{z}_{1:k})$ approximately, which is much more efficient.

Now let us derive the approximate recursive formula to calculate $L_A(\mathbf{x}_k | \mathbf{z}_{1:k})$. $I_A(\mathbf{x}_{0:k-1} | \mathbf{z}_{1:k-1})$ can be decomposed as

$$I_A(\mathbf{x}_{0:k-1} | \mathbf{z}_{1:k-1}) = \begin{bmatrix} A_{k-1}^{11} & A_{k-1}^{12} \\ A_{k-1}^{21} & A_{k-1}^{22} \end{bmatrix} \tag{3.32}$$

3.3 Conditional PCRLB for Nonlinear Dynamical Systems

Taking the inverse of the above matrix and applying (3.16), we have

$$L_A(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1}) = A_{k-1}^{22} - A_{k-1}^{21} (A_{k-1}^{11})^{-1} A_{k-1}^{12} \quad (3.33)$$

Now consider $I_A(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})$. We have

$$\begin{aligned} I_A(\mathbf{x}_{0:k}|\mathbf{z}_{1:k}) &= \\ E_{p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})} &\left\{ (-1) \begin{bmatrix} \Delta_{\mathbf{x}_{0:k-2}}^{\mathbf{x}_{0:k-2}} & \Delta_{\mathbf{x}_{0:k-2}}^{\mathbf{x}_{k-1}} & \mathbf{0} \\ \Delta_{\mathbf{x}_{k-1}}^{\mathbf{x}_{0:k-2}} & \Delta_{\mathbf{x}_{k-1}}^{\mathbf{x}_{k-1}} & \Delta_{\mathbf{x}_{k-1}}^{\mathbf{x}_k} \\ \mathbf{0} & \Delta_{\mathbf{x}_k}^{\mathbf{x}_{k-1}} & \Delta_{\mathbf{x}_k}^{\mathbf{x}_k} \end{bmatrix} \log p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k}) \right\} \\ &= \begin{bmatrix} E_{p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})} \left[-\Delta_{\mathbf{x}_{0:k-2}}^{\mathbf{x}_{0:k-2}} \log p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k}) \right] & A_{k-1}^{12} & \mathbf{0} \\ A_{k-1}^{21} & A_{k-1}^{22} + S_k^{11} & S_k^{12} \\ \mathbf{0} & S_k^{21} & S_k^{22} \end{bmatrix} \end{aligned} \quad (3.34)$$

where $\mathbf{0}$ s stand for blocks of zeros of appropriate dimensions. In general, there is no recursive method to calculate $L_A(\mathbf{x}_k|\mathbf{z}_{1:k})$. This is because the measurement \mathbf{z}_k provides new information about the system state in the past ($\mathbf{x}_{0:k-1}$), which will affect the top-left part of $I_A(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})$. As we can see, $I_A(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})$ is a block tridiagonal matrix. The top-left sub-matrix of $I_A(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})$ is a function of \mathbf{z}_k , which can be approximated by its expectation with respect to $p(\mathbf{z}_k|\mathbf{z}_{1:k-1})$, if we take \mathbf{z}_k and $\mathbf{z}_{1:k-1}$ as random vector and measurement realizations respectively. So we have

$$\begin{aligned} &E_{p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})} \left[-\Delta_{\mathbf{x}_{0:k-2}}^{\mathbf{x}_{0:k-2}} \log p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k}) \right] \\ &\approx E_{p(\mathbf{z}_k|\mathbf{z}_{1:k-1})} \left\{ E_{p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})} \left[-\Delta_{\mathbf{x}_{0:k-2}}^{\mathbf{x}_{0:k-2}} \log p(\mathbf{x}_{0:k-1}|\mathbf{z}_{1:k-1}) \right] \right\} \\ &= E_{p(\mathbf{x}_{0:k-1}|\mathbf{z}_{1:k-1})} \left[-\Delta_{\mathbf{x}_{0:k-2}}^{\mathbf{x}_{0:k-2}} \log p(\mathbf{x}_{0:k-1}|\mathbf{z}_{1:k-1}) \right] \\ &= A_{k-1}^{11} \end{aligned} \quad (3.35)$$

where (3.6) has been used. Because the auxiliary Fisher information matrix $L_A(\mathbf{x}_k|\mathbf{z}_{1:k})$ is equal to the inverse of the lower-right block of $I_A^{-1}(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})$, we have

$$\begin{aligned} &L_A(\mathbf{x}_k|\mathbf{z}_{1:k}) \\ &\approx S_k^{22} - [\mathbf{0} \quad S_k^{21}] \begin{bmatrix} A_{k-1}^{11} & A_{k-1}^{12} \\ A_{k-1}^{21} & A_{k-1}^{22} + S_k^{11} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{0} \\ S_k^{12} \end{bmatrix} \\ &= S_k^{22} - S_k^{21} [S_k^{11} + A_{k-1}^{22} - A_{k-1}^{21} (A_{k-1}^{11})^{-1} A_{k-1}^{12}]^{-1} S_k^{12} \\ &= S_k^{22} - S_k^{21} [S_k^{11} + L_A(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})]^{-1} S_k^{12} \end{aligned} \quad (3.36)$$

where

$$S_k^{11} \triangleq E_{p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})} \left[-\Delta_{\mathbf{x}_{k-1}}^{\mathbf{x}_{k-1}} \log p(\mathbf{x}_k|\mathbf{x}_{k-1}) \right] \quad (3.37)$$

$$S_k^{12} \triangleq E_{p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})} \left[-\Delta_{\mathbf{x}_{k-1}}^{\mathbf{x}_k} \log p(\mathbf{x}_k|\mathbf{x}_{k-1}) \right] = (S_k^{21})^T \quad (3.38)$$

$$S_k^{22} \triangleq E_{p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})} \left\{ -\Delta_{\mathbf{x}_k}^{\mathbf{x}_k} [\log p(\mathbf{x}_k|\mathbf{x}_{k-1}) + \log p(\mathbf{z}_k|\mathbf{x}_k)] \right\} \quad (3.39)$$

In summary, the sequence of $\{L_A(\mathbf{x}_k|\mathbf{z}_{1:k})\}$ can be computed recursively as provided in the following approximation

Approximation 1

$$L_A(\mathbf{x}_k|\mathbf{z}_{1:k}) \approx S_k^{22} - S_k^{21} [S_k^{11} + L_A(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})]^{-1} S_k^{12} \quad (3.40)$$

In the recursive evaluation approach, the approximation made in (3.35) may cause cumulative error. The theoretical analysis of the cumulative approximation error is very difficult. In Section 3.5, this approximation method is justified through simulation experiments for a highly nonlinear system. In the experiments, the conditional PCRLB evaluated using Theorem 1 and the method with the approximated $L_A(\mathbf{x}_k|\mathbf{z}_{1:k})$ provided by Approximation 1 and that evaluated based on Theorem 1 and the exact $L_A(\mathbf{x}_k|\mathbf{z}_{1:k})$ by calculating (3.34) without approximation yield results that are very close to each other.

3.4 A Sequential Monte Carlo solution for Conditional PCRLB

In Section 3.3, we have shown that given the available measurement data $\mathbf{z}_{1:k}$, the conditional Fisher information matrix $L(\mathbf{x}_{k+1}|\mathbf{z}_{1:k})$ can be recursively calculated according to Theorem 1 and Approximation 1. However, in most cases, direct computation of B_k^{11} , B_k^{12} , B_k^{22} , S_k^{11} , S_k^{12} , and S_k^{22} involves high-dimensional integration, and in general analytical solutions do not exist. Here sequential Monte Carlo methods, or particle filters, are proposed to evaluate these terms. For nonlinear non-Gaussian Bayesian recursive estimation problems, the particle filter is a very

3.4 A Sequential Monte Carlo solution for Conditional PCRLB

popular and powerful tool. Based on importance sampling techniques, particle filters approximate the high-dimension integration using Monte Carlo simulations and interested readers are referred to [16, 29] for details. For nonlinear dynamic systems that use particle filters for state estimation, the proposed particle filter based conditional PCRLB evaluation solution is very convenient, since the auxiliary Fisher information matrix $L_A(\mathbf{x}_k|\mathbf{z}_{1:k})$ and the conditional Fisher information matrix $L(\mathbf{x}_{k+1}|\mathbf{z}_{1:k})$ can be evaluated online as by-products of the particle filter state estimation process, as shown later in the chapter.

Under the assumptions that the states evolve according to a first-order Markov process and the observations are conditionally independent given the states, the PDF $p(\mathbf{x}_{0:k+1}, \mathbf{z}_{k+1}|\mathbf{z}_{1:k})$ can be factorized as

$$\begin{aligned} p_{k+1}^c &= p(\mathbf{z}_{k+1}|\mathbf{x}_{k+1})p(\mathbf{x}_{k+1}|\mathbf{x}_k) \\ &\times p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k}) \end{aligned} \quad (3.41)$$

Letting N denote the number of particles used in the particle filter, the posterior PDF $p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})$ at time k can be approximated by the particles[16]

$$p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k}) \approx \frac{1}{N} \sum_{l=1}^N \delta(\mathbf{x}_{0:k} - \mathbf{x}_{0:k}^l) \quad (3.42)$$

where we assume that the resampling has been performed at time k , so that each particle has an identical weight $\frac{1}{N}$. With (3.41) and (3.42), we can readily show that

$$p_{k+1}^c \approx \frac{1}{N} p(\mathbf{z}_{k+1}|\mathbf{x}_{k+1})p(\mathbf{x}_{k+1}|\mathbf{x}_k) \sum_{l=1}^N \delta(\mathbf{x}_{0:k} - \mathbf{x}_{0:k}^l)$$

We also derive another approximation for $p(\mathbf{x}_{0:k+1}, \mathbf{z}_{k+1}|\mathbf{z}_{1:k})$, which is given by the following proposition.

Proposition 2

$$\begin{aligned} &p(\mathbf{x}_{0:k+1}, \mathbf{z}_{k+1}|\mathbf{z}_{1:k}) \\ &\approx \frac{1}{N} \sum_{l=1}^N \delta(\mathbf{x}_{0:k+1} - \mathbf{x}_{0:k+1}^l) p(\mathbf{z}_{k+1}|\mathbf{x}_{k+1}^l) \end{aligned} \quad (3.43)$$

Proof: See Appendix B.

Note that even though approximations in (3.42) and (3.43) require that each particle represents one system state realization from time 0 to time k ($\mathbf{x}_{0:k}$), we will show later that for calculating conditional PCRLB at time step k , it is sufficient for each particle to keep system state realization at time steps $k-1$ and k only, which means that we only need to keep $\mathbf{x}_{k-1:k}$ for computation. This results in a significantly reduced burden for system memory.

In this section, we will consider the general form of the conditional PCRLB for any nonlinear/non-Gaussian dynamic system, as well as two special cases.

3.4.1 General Formulation

The general form is given to calculate each component in (3.17) and (3.40) for any nonlinear/non-Gaussian system. In the following equations, the superscripts represent the particle index. We also assume that the derivatives and expectations exist and the integration and derivatives are exchangeable. For \mathbf{B}_k^{11} , we have

$$\begin{aligned} B_k^{11} &= E_{p_{k+1}^c} \left\{ -\Delta_{\mathbf{x}_k}^{\mathbf{x}_k} \log p(\mathbf{x}_{k+1}|\mathbf{x}_k) \right\} \\ &= E_{p_{k+1}^c} \left[\frac{\nabla_{\mathbf{x}_k} p(\mathbf{x}_{k+1}|\mathbf{x}_k) \nabla_{\mathbf{x}_k}^T p(\mathbf{x}_{k+1}|\mathbf{x}_k)}{p^2(\mathbf{x}_{k+1}|\mathbf{x}_k)} - \frac{\Delta_{\mathbf{x}_k}^{\mathbf{x}_k} p(\mathbf{x}_{k+1}|\mathbf{x}_k)}{p(\mathbf{x}_{k+1}|\mathbf{x}_k)} \right] \end{aligned} \quad (3.44)$$

First, it is easy to show that

$$E_{p_{k+1}^c} \left[\frac{\Delta_{\mathbf{x}_k}^{\mathbf{x}_k} p(\mathbf{x}_{k+1}|\mathbf{x}_k)}{p(\mathbf{x}_{k+1}|\mathbf{x}_k)} \right] = 0 \quad (3.45)$$

Now let us define

$$g_1(\mathbf{x}_k, \mathbf{x}_{k+1}) \triangleq \frac{\nabla_{\mathbf{x}_k} p(\mathbf{x}_{k+1}|\mathbf{x}_k) \nabla_{\mathbf{x}_k}^T p(\mathbf{x}_{k+1}|\mathbf{x}_k)}{p^2(\mathbf{x}_{k+1}|\mathbf{x}_k)} \quad (3.46)$$

By substituting (3.43), (3.45), and (3.46) into (3.44), we have

$$\begin{aligned} B_k^{11} &= E_{p_{k+1}^c} [g_1(\mathbf{x}_k, \mathbf{x}_{k+1})] \\ &\approx \frac{1}{N} \sum_{l=1}^N g_1(\mathbf{x}_k^l, \mathbf{x}_{k+1}^l) \end{aligned} \quad (3.47)$$

3.4 A Sequential Monte Carlo solution for Conditional PCRLB

\mathbf{B}_k^{12} Following a similar procedure, we have

$$B_k^{12} \approx \frac{1}{N} \sum_{l=1}^N \left. \frac{\nabla_{\mathbf{x}_k} p(\mathbf{x}_{k+1} | \mathbf{x}_k) \nabla_{\mathbf{x}_{k+1}}^T p(\mathbf{x}_{k+1} | \mathbf{x}_k)}{p^2(\mathbf{x}_{k+1} | \mathbf{x}_k)} \right|_{\{\mathbf{x}_k, \mathbf{x}_{k+1}\} = \{\mathbf{x}_k^l, \mathbf{x}_{k+1}^l\}} \quad (3.48)$$

$$\mathbf{B}_k^{22} = \mathbf{B}_k^{22,a} + \mathbf{B}_k^{22,b}$$

$$B_k^{22,a} \approx \frac{1}{N} \sum_{l=1}^N \left. \frac{\nabla_{\mathbf{x}_{k+1}} p(\mathbf{x}_{k+1} | \mathbf{x}_k) \nabla_{\mathbf{x}_{k+1}}^T p(\mathbf{x}_{k+1} | \mathbf{x}_k)}{p^2(\mathbf{x}_{k+1} | \mathbf{x}_k)} \right|_{\{\mathbf{x}_k, \mathbf{x}_{k+1}\} = \{\mathbf{x}_k^l, \mathbf{x}_{k+1}^l\}} \quad (3.49)$$

$$\begin{aligned} B_k^{22,b} &= E_{p_{\mathbf{x}_{k+1}}^c} \left[-\Delta_{\mathbf{x}_{k+1}}^{\mathbf{x}_{k+1}} \log p(\mathbf{z}_{k+1} | \mathbf{x}_{k+1}) \right] \\ &\approx \frac{1}{N} \sum_{l=1}^N g_2(\mathbf{x}_{k+1}^l) \end{aligned} \quad (3.50)$$

where

$$g_2(\mathbf{x}_{k+1}) \triangleq \int \frac{\nabla_{\mathbf{x}_{k+1}} p(\mathbf{z}_{k+1} | \mathbf{x}_{k+1}) \nabla_{\mathbf{x}_{k+1}}^T p(\mathbf{z}_{k+1} | \mathbf{x}_{k+1})}{p(\mathbf{z}_{k+1} | \mathbf{x}_{k+1})} d\mathbf{z}_{k+1} \quad (3.51)$$

For the cases where the integration in (3.51) does not have a closed-form solution, it can be approximated by numerical integration approaches.

$$\begin{aligned} S_k^{11} &= E_{p(\mathbf{x}_{0:k} | \mathbf{z}_{1:k})} \left[-\Delta_{\mathbf{x}_{k-1}}^{\mathbf{x}_{k-1}} \log p(\mathbf{x}_k | \mathbf{x}_{k-1}) \right] \\ &= E_{p(\mathbf{x}_{0:k} | \mathbf{z}_{1:k})} \left[\frac{\nabla_{\mathbf{x}_{k-1}} p(\mathbf{x}_k | \mathbf{x}_{k-1}) \nabla_{\mathbf{x}_{k-1}}^T p(\mathbf{x}_k | \mathbf{x}_{k-1})}{p^2(\mathbf{x}_k | \mathbf{x}_{k-1})} - \frac{\Delta_{\mathbf{x}_{k-1}}^{\mathbf{x}_{k-1}} p(\mathbf{x}_k | \mathbf{x}_{k-1})}{p(\mathbf{x}_k | \mathbf{x}_{k-1})} \right] \end{aligned} \quad (3.52)$$

Since $\mathbf{z}_{1:k}$ are available measurement data, the posterior PDF $p(\mathbf{x}_{0:k} | \mathbf{z}_{1:k})$ can be approximated through sequential Monte Carlo approaches. Plugging (3.42) into the above equation, we have

$$S_k^{11} \approx \frac{1}{N} \sum_{l=1}^N g_3(\mathbf{x}_{k-1}^l, \mathbf{x}_k^l) \quad (3.53)$$

where

$$g_3(\mathbf{x}_{k-1}, \mathbf{x}_k) \triangleq \frac{\nabla_{\mathbf{x}_{k-1}} p(\mathbf{x}_k | \mathbf{x}_{k-1}) \nabla_{\mathbf{x}_{k-1}}^T p(\mathbf{x}_k | \mathbf{x}_{k-1})}{p^2(\mathbf{x}_k | \mathbf{x}_{k-1})} - \frac{\Delta_{\mathbf{x}_{k-1}}^{\mathbf{x}_{k-1}} p(\mathbf{x}_k | \mathbf{x}_{k-1})}{p(\mathbf{x}_k | \mathbf{x}_{k-1})} \quad (3.54)$$

3.4 A Sequential Monte Carlo solution for Conditional PCRLB

Following a similar procedure as in calculating S_k^{11} , we have

$$S_k^{12} \approx \frac{1}{N} \sum_{l=1}^N g_4(\mathbf{x}_{k-1}^l, \mathbf{x}_k^l) \quad (3.55)$$

where

$$g_4(\mathbf{x}_{k-1}, \mathbf{x}_k) \triangleq \frac{\nabla_{\mathbf{x}_{k-1}} p(\mathbf{x}_k | \mathbf{x}_{k-1}) \nabla_{\mathbf{x}_k}^T p(\mathbf{x}_k | \mathbf{x}_{k-1})}{p^2(\mathbf{x}_k | \mathbf{x}_{k-1})} - \frac{\Delta_{\mathbf{x}_{k-1}}^{\mathbf{x}_k} p(\mathbf{x}_k | \mathbf{x}_{k-1})}{p(\mathbf{x}_k | \mathbf{x}_{k-1})} \quad (3.56)$$

S_k^{22} consists of two parts, $S_k^{22} = S_k^{22,a} + S_k^{22,b}$, where

$$S_k^{22,a} \approx \frac{1}{N} \sum_{l=1}^N \left[\frac{\nabla_{\mathbf{x}_k} p(\mathbf{x}_k | \mathbf{x}_{k-1}) \nabla_{\mathbf{x}_k}^T p(\mathbf{x}_k | \mathbf{x}_{k-1})}{p^2(\mathbf{x}_k | \mathbf{x}_{k-1})} - \frac{\Delta_{\mathbf{x}_k}^{\mathbf{x}_k} p(\mathbf{x}_k | \mathbf{x}_{k-1})}{p(\mathbf{x}_k | \mathbf{x}_{k-1})} \right] \Bigg|_{\{\mathbf{x}_{k-1}, \mathbf{x}_k\} = \{\mathbf{x}_{k-1}^l, \mathbf{x}_k^l\}} \quad (3.57)$$

and

$$S_k^{22,b} \approx \frac{1}{N} \sum_{l=1}^N \left[\frac{\nabla_{\mathbf{x}_k} p(\mathbf{z}_k | \mathbf{x}_k) \nabla_{\mathbf{x}_k}^T p(\mathbf{z}_k | \mathbf{x}_k)}{p^2(\mathbf{z}_k | \mathbf{x}_k)} - \frac{\Delta_{\mathbf{x}_k}^{\mathbf{z}_k} p(\mathbf{z}_k | \mathbf{x}_k)}{p(\mathbf{z}_k | \mathbf{x}_k)} \right] \Bigg|_{\mathbf{x}_k = \mathbf{x}_k^l} \quad (3.58)$$

Taking a closer look at approximations made in this subsection, it is clear that at time step k , for the calculation of the conditional PCRLB at time $k+1$, only the values of system states from time $k-1$ to time $k+1$ ($\mathbf{x}_{k-1:k+1}^l$) are needed. Moreover, when the system transits from step k to step $k+1$, it is sufficient to propagate and update the particle set from $\{\mathbf{x}_{k-1:k}^l\}$ to $\{\mathbf{x}_{k:k+1}^l\}$, where $l = 1, \dots, N$.

With numerical integrations provided by the particle filter, the approach for evaluating the conditional PCRLB works recursively as follows. At time k , when the measurement \mathbf{z}_k is available, the weights of the particle sets $\{\mathbf{x}_{k-1:k}^l\}$ are updated, which is followed by a re-sampling procedure. Then each particle $\mathbf{x}_{k-1:k}^l$ has an equal constant weight of $1/N$. $\{\mathbf{x}_{k-1:k}^l\}$ will be used for the calculation of S_k^{11} , S_k^{12} and S_k^{22} . Then only the particles $\{\mathbf{x}_k^l\}$ are propagated to the next time step according to (3.4). The particle set $\{\mathbf{x}_{k:k+1}^l\}$ is used to evaluate B_k^{11} , B_k^{12} and B_k^{22} . At the end of the k th time step, for the l th particle, only $\mathbf{x}_{k:k+1}^l$ will be preserved and passed to the next $(k+1)$ time step.

Note that the particle filter is an approximate solution to the optimal non-linear estimator. At the $(k-1)$ th iteration, based on the information state

3.4 A Sequential Monte Carlo solution for Conditional PCRLB

$p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})$, which is a function of $\mathbf{z}_{1:k-1}$ and completely summarizes the past of the system in a probabilistic sense [30], the optimal nonlinear estimator calculates the new information state $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ by incorporating the new measurement \mathbf{z}_k . As a result, the optimal nonlinear estimator of \mathbf{x}_k at time k is a function of all the measurements up to time k , namely $\mathbf{z}_{1:k}$. The particle filter is nothing but a numerical approximation to the optimal estimator, which recursively updates the particle weights using arriving new measurements, and hence is a function of the measurements up to the current time. Therefore, the conditional PCRLB approximated by sequential Monte-Carlo methods depends on the history of the measurements. Details of the optimal estimator and the information state, and the particle filter can be found in [30] and [16] respectively.

Now let us investigate the computational complexities of the recursive conditional FIM, which can be evaluated using Theorem 1 and Approximation 1, and the recursive unconditional PCRLB that renews its prior at each iteration. Lemma 1 shows that these two methods differ only in the computation of \tilde{J}_k and $L_A(\mathbf{x}_k|\mathbf{z}_{1:k})$. When applying the particle filter, at each time k , the complexity for computing the common terms (B_k^{11} , B_k^{12} , and B_k^{22}) in Lemma 1 is linear in the number of particles (N). The terms used in Approximation 1 to recursively compute $L_A(\mathbf{x}_k|\mathbf{z}_{1:k})$ (S_k^{11} , S_k^{12} , and S_k^{22}) also have complexities that are linear in N . Since $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ has been represented by a set of particles and associated weights, $\tilde{J}_k = E_{p(\mathbf{x}_k|\mathbf{z}_{1:k})}\{-\Delta_{\mathbf{x}_k}^{\mathbf{x}_k} \log p(\mathbf{x}_k|\mathbf{z}_{1:k})\}$ could only be evaluated numerically, with a complexity at least linear in N . Thus, the computation of \tilde{J}_k has a complexity that is at least in the same order of that of $L_A(\mathbf{x}_k|\mathbf{z}_{1:k})$.

3.4.2 Additive Gaussian Noise Case

Here we consider a special case of nonlinear dynamic systems with additive Gaussian noises. It is assumed that the dynamic system has the following state and measurement equations:

$$\mathbf{x}_{k+1} = f_k(\mathbf{x}_k) + \mathbf{u}_k \tag{3.59}$$

$$\mathbf{z}_k = h_k(\mathbf{x}_k) + \mathbf{v}_k \tag{3.60}$$

3.4 A Sequential Monte Carlo solution for Conditional PCRLB

where $f_k(\cdot)$ and $h_k(\cdot)$ are nonlinear state transition and measurement functions respectively, \mathbf{u}_k is the white Gaussian state process noise with zero mean and covariance matrix Q_k , and \mathbf{v}_k is the white Gaussian measurement noise with zero mean and covariance matrix R_k . The sequences $\{\mathbf{u}_k\}$ and $\{\mathbf{v}_k\}$ are mutually independent. With these assumptions and notations, the transition prior of the state can be written as

$$p(\mathbf{x}_{k+1}|\mathbf{x}_k) = \frac{1}{(2\pi)^{\frac{n_x}{2}} |Q_k|^{\frac{1}{2}}} \exp \left\{ -\frac{1}{2} [\mathbf{x}_{k+1} - f_k(\mathbf{x}_k)]^T Q_k^{-1} [\mathbf{x}_{k+1} - f_k(\mathbf{x}_k)] \right\} \quad (3.61)$$

Taking the logarithm of the above PDF, we have

$$\begin{aligned} & -\log p(\mathbf{x}_{k+1}|\mathbf{x}_k) \\ &= c_0 + \frac{1}{2} [\mathbf{x}_{k+1} - f_k(\mathbf{x}_k)]^T Q_k^{-1} [\mathbf{x}_{k+1} - f_k(\mathbf{x}_k)] \end{aligned} \quad (3.62)$$

where c_0 denotes a constant independent of \mathbf{x}_k and \mathbf{x}_{k+1} . Then the first and second-order partial derivatives of $\log p(\mathbf{x}_{k+1}|\mathbf{x}_k)$ with respect to \mathbf{x}_k can be derived respectively as

$$\nabla_{\mathbf{x}_k} \log p(\mathbf{x}_{k+1}|\mathbf{x}_k) = [\nabla_{\mathbf{x}_k} f_k(\mathbf{x}_k)] Q_k^{-1} (\mathbf{x}_{k+1} - f_k(\mathbf{x}_k)) \quad (3.63)$$

and

$$\begin{aligned} & -\Delta_{\mathbf{x}_k}^{\mathbf{x}_k} \log p(\mathbf{x}_{k+1}|\mathbf{x}_k) = \\ & [\nabla_{\mathbf{x}_k} f_k(\mathbf{x}_k)] Q_k^{-1} [\nabla_{\mathbf{x}_k}^T f_k(\mathbf{x}_k)] - [\Delta_{\mathbf{x}_k}^{\mathbf{x}_k} f_k(\mathbf{x}_k)] \tilde{\Sigma}_{\mathbf{u}_k}^{-1} \Upsilon_k^{11} \end{aligned} \quad (3.64)$$

where

$$\tilde{\Sigma}_{\mathbf{u}_k}^{-1} = \begin{bmatrix} Q_k^{-1} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & Q_k^{-1} & \mathbf{0} & \vdots \\ \vdots & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{0} & Q_k^{-1} \end{bmatrix}_{n_x^2 \times n_x^2} \quad (3.65)$$

and

$$\Upsilon_k^{11} = \begin{bmatrix} \mathbf{x}_{k+1} - f_k(\mathbf{x}_k) & \dots & \mathbf{0} \\ \vdots & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{x}_{k+1} - f_k(\mathbf{x}_k) \end{bmatrix}_{n_x^2 \times n_x} \quad (3.66)$$

3.4 A Sequential Monte Carlo solution for Conditional PCRLB

For vector-valued functions $f(\cdot) = [f_1, f_2, \dots, f_{n_x}]^T$, the first order and second order derivatives of $f(\cdot)$ are defined respectively as

$$\nabla_{\mathbf{x}} f(\mathbf{x}) = [\nabla_{\mathbf{x}} f_1, \nabla_{\mathbf{x}} f_2, \dots, \nabla_{\mathbf{x}} f_{n_x}]_{n_x \times n_x} \quad (3.67)$$

$$\Delta_{\mathbf{x}}^{\mathbf{x}} f(\mathbf{x}) = [\Delta_{\mathbf{x}}^{\mathbf{x}} f_1, \Delta_{\mathbf{x}}^{\mathbf{x}} f_2, \dots, \Delta_{\mathbf{x}}^{\mathbf{x}} f_{n_x}]_{n_x \times n_x^2} \quad (3.68)$$

By substituting (3.43) and (3.64) into (3.44), we have

$$\begin{aligned} B_k^{11} &= E_{p_{k+1}^c} \{-\Delta_{\mathbf{x}_k}^{\mathbf{x}_k} \log p(\mathbf{x}_{k+1} | \mathbf{x}_k)\} \\ &\approx \frac{1}{N} \sum_{l=1}^N \left([\nabla_{\mathbf{x}_k} f(\mathbf{x}_k)] Q_k^{-1} [\nabla_{\mathbf{x}_k}^T f(\mathbf{x}_k)] \right) \Big|_{\mathbf{x}_k = \mathbf{x}_k^l} \end{aligned} \quad (3.69)$$

where the following identity has been used

$$E_{p(\mathbf{x}_{k+1} | \mathbf{x}_k)} \{\Upsilon_k^{11}\} = \mathbf{0} \quad (3.70)$$

From (3.62), we have

$$-\Delta_{\mathbf{x}_{k+1}}^{\mathbf{x}_k} \log p(\mathbf{x}_{k+1} | \mathbf{x}_k) = -Q_k^{-1} \nabla_{\mathbf{x}_k}^T f_k(\mathbf{x}_k)$$

Similarly, we have

$$\begin{aligned} B_k^{21} &= E_{p_{k+1}^c} \{-\Delta_{\mathbf{x}_{k+1}}^{\mathbf{x}_k} \log p(\mathbf{x}_{k+1} | \mathbf{x}_k)\} \\ &\approx -\frac{1}{N} \sum_{l=1}^N \left(Q_k^{-1} \nabla_{\mathbf{x}_k}^T f_k(\mathbf{x}_k) \right) \Big|_{\mathbf{x}_k = \mathbf{x}_k^l} \end{aligned} \quad (3.71)$$

As for $B_k^{22,a}$ and $B_k^{22,b}$, we have

$$B_k^{22,a} = Q_k^{-1} \quad (3.72)$$

$$\begin{aligned} B_k^{22,b} &= \frac{1}{N} \sum_{l=1}^N \\ &\left([\nabla_{\mathbf{x}_{k+1}} h(\mathbf{x}_{k+1})] R_{k+1}^{-1} [\nabla_{\mathbf{x}_{k+1}}^T h(\mathbf{x}_{k+1})] \right) \Big|_{\mathbf{x}_{k+1} = \mathbf{x}_{k+1}^l} \end{aligned} \quad (3.73)$$

whose derivation is provided in Appendix C.

3.4 A Sequential Monte Carlo solution for Conditional PCRLB

The approximations for S_k^{11} , S_k^{21} , and S_k^{22} can be derived similarly. Using (3.42), we have

$$S_k^{11} \approx \frac{1}{N} \sum_{l=1}^N g_5(\mathbf{x}_{k-1}^l, \mathbf{x}_k^l) \quad (3.74)$$

where

$$g_5(\mathbf{x}_{k-1}, \mathbf{x}_k) = [\nabla_{\mathbf{x}_{k-1}} f_{k-1}(\mathbf{x}_{k-1})] Q_{k-1}^{-1} [\nabla_{\mathbf{x}_{k-1}}^T f_{k-1}(\mathbf{x}_{k-1})] - [\Delta_{\mathbf{x}_{k-1}}^{\mathbf{x}_{k-1}} f_{k-1}(\mathbf{x}_{k-1})] \tilde{\Sigma}_{\mathbf{u}_{k-1}}^{-1} \Upsilon_{k-1}^{11}(\mathbf{x}_{k-1}, \mathbf{x}_k) \quad (3.75)$$

$$S_k^{21} \approx -\frac{1}{N} \sum_{l=1}^N \left[Q_{k-1}^{-1} \nabla_{\mathbf{x}_{k-1}}^T f_{k-1}(\mathbf{x}_{k-1}) \right] \Big|_{\mathbf{x}_{k-1}=\mathbf{x}_{k-1}^l} \quad (3.76)$$

$$S_k^{22,a} = Q_{k-1}^{-1} \quad (3.77)$$

and

$$S_k^{22,b} \approx \frac{1}{N} \sum_{l=1}^N \left\{ [\nabla_{\mathbf{x}_k} h_k(\mathbf{x}_k)] R_k^{-1} [\nabla_{\mathbf{x}_k}^T h_k(\mathbf{x}_k)] - \Delta_{\mathbf{x}_k}^{\mathbf{x}_k} h_k(\mathbf{x}_k) \tilde{\Sigma}_{\mathbf{v}_k}^{-1} \Upsilon_k^{22,b} \right\} \Big|_{\mathbf{x}_k=\mathbf{x}_k^l} \quad (3.78)$$

where $\tilde{\Sigma}_{\mathbf{v}_k}^{-1}$ and $\Upsilon_k^{22,b}$ are defined in Appendix C.

3.4.3 Linear System with Additive Gaussian Noise Case

The Gaussian dynamic system is characterized by its system state equation and measurement equation:

$$\mathbf{x}_{k+1} = F_k \mathbf{x}_k + \mathbf{u}_k \quad (3.79)$$

$$\mathbf{z}_k = H_k \mathbf{x}_k + \mathbf{v}_k \quad (3.80)$$

where \mathbf{u}_k and \mathbf{v}_k have been defined in Subsection 3.4.2, and F_k and H_k are known matrices with proper dimensions. In such a linear Gaussian system, we have the following theorem

3.4 A Sequential Monte Carlo solution for Conditional PCRLB

Theorem 2 *If the initial conditions for both the conditional PCRLB and the unconditional PCRLB are the same, namely*

$$J_0 = I_A(\mathbf{x}_0) \quad (3.81)$$

then the conditional PCRLB and PCRLB are equivalent for linear Gaussian dynamic systems, and all the three Fisher information matrices, namely the unconditional Fisher information, the conditional Fisher information and the auxiliary Fisher information, are equivalent. Mathematically, we have

$$J_k = I_A(\mathbf{x}_k | \mathbf{z}_{1:k}) = I(\mathbf{x}_k | \mathbf{z}_{1:k-1}) \quad (3.82)$$

Proof: In a linear Gaussian system, for the unconditional PCRLB, it can be shown [5] that

$$\begin{aligned} J_{k+1} &= D_k^{22} - D_k^{21}(J_k + D_k^{11})^{-1}D_k^{12} \\ &= H_{k+1}^T R_{k+1}^{-1} H_{k+1} + (Q_k + F_k J_k^{-1} F_k^T)^{-1} \end{aligned} \quad (3.83)$$

which is nothing but the recursive formula for the inverse covariance matrix in an information filter [30]. Based on results in Section 3.4.2, it can be proved that

$$\begin{aligned} S_k^{11} &= B_{k-1}^{11} = F_{k-1}^T Q_{k-1}^{-1} F_{k-1} = D_{k-1}^{11} \\ S_k^{12} &= B_{k-1}^{12} = -F_{k-1}^T Q_{k-1}^{-1} = D_{k-1}^{12} \\ S_k^{22} &= B_{k-1}^{22} = Q_{k-1}^{-1} + H_k^T R_k^{-1} H_k = D_{k-1}^{22} \end{aligned} \quad (3.84)$$

According to Theorem 1, we have the recursive formula for the auxiliary Fisher information matrix

$$\begin{aligned} I_A(\mathbf{x}_k | \mathbf{z}_{1:k}) &= \\ D_{k-1}^{22} - D_{k-1}^{21} [D_{k-1}^{11} + I_A(\mathbf{x}_{k-1} | \mathbf{z}_{1:k-1})]^{-1} D_{k-1}^{12} \end{aligned} \quad (3.85)$$

Comparing (3.83) and (3.86), it is clear that J_k and $I_A(\mathbf{x}_k | \mathbf{z}_{1:k})$ have the same recursive formula. Since they start from the same initial conditions ($J_0 = I_A(\mathbf{x}_0)$), we have

$$J_k = I_A(\mathbf{x}_k | \mathbf{z}_{1:k}) \quad (3.86)$$

Now using Theorem 1, we have

$$\begin{aligned} I(\mathbf{x}_{k+1}|\mathbf{z}_{1:k}) &= D_k^{22} - D_k^{21} [D_k^{11} + I_A(\mathbf{x}_k|\mathbf{z}_{1:k})]^{-1} D_k^{12} \\ &= I_A(\mathbf{x}_{k+1}|\mathbf{z}_{1:k+1}) = J_{k+1} \end{aligned} \quad (3.87)$$

Q.E.D.

Theorem 2 indicates that in a linear Gaussian system, there is no need to use an online conditional PCRLB bound, which is equivalent to the unconditional PCRLB. Note that in such a case, the Kalman filter is the optimal estimator, where the recursive calculations of filter gains and covariance matrices can be performed offline, since they are independent of the state [30]. In addition, Theorem 2 provides the insight that the approximation provided in Theorem 1 yields the exact result when the system is linear and Gaussian. Therefore, one can expect that for a system with weak nonlinearity and non-Gaussianity, the approximation error incurred by the recursive conditional PCRLB evaluation approach provided in Theorem 1 will be smaller than that in a highly nonlinear system.

3.5 Simulation Results for Comparison

In this section, we present some illustrative examples to demonstrate the accuracy of the computed bounds. Here we consider the univariate non-stationary growth model (UNGM), a highly nonlinear and bimodal model. The UNGM is very useful in econometrics, and has been used in [29, 31, 32]. In a UNGM, the dynamic state space equations are given by

$$x_{k+1} = \alpha x_k + \beta \frac{x_k}{1 + x_k^2} + \gamma \cos(1.2k) + u_k \quad (3.88)$$

$$z_k = \kappa x_k^2 + v_k \quad (3.89)$$

where u_k and v_k are the state process noise and measurement noise respectively, and they are white Gaussian with zero means and variances σ_u^2 and σ_v^2 .

In the simulations, the conditional MSE is obtained recursively as follows. At time k , the posterior PDF is calculated using a particle filter given the measurement $z_{1:k}$. 1000 Monte Carlo trials are performed to generate independent

realizations of z_{k+1} according to the measurement Equation (3.89). The conditional MSE, $\text{MSE}(\hat{x}_{k+1}|z_{1:k})$, is obtained based on the 1000 Monte-Carlo trials. At the next time step ($k + 1$), a single realization of z_{k+1} is picked randomly among the 1000 realizations, and concatenated with the past measurement history to form $z_{1:k+1}$. The particles and weights corresponding to this particular z_{k+1} are stored and used for the $(k + 1)$ th iteration. The recursive conditional PCRLB with approximations mentioned in Theorem 1 and Approximation 1 is used throughout the experiments, unless otherwise specified. The same particle filter can be used to evaluate both the conditional MSE and the conditional PCRLB.

3.5.1 Conditional PCRLB vs Conditional MSE for UNGM

We set parameters $\alpha = 1, \beta = 5, \gamma = 8, \sigma_u^2 = 1, \sigma_v^2 = 1$, and $\kappa = 1/20$ for UNGM. Fig. 3.1 shows the system states and measurements over a period of 20 discrete time steps. Due to the measurement equation of the UNGM specified in (3.89), there is bi-modality inherent in the filtering problem. As a result, the observation does not follow the system state very closely, as shown in Fig. 3.1. In such a case, it is very difficult to track the state using conventional methods, and the particle filter demonstrates better tracking performance than the extended Kalman filter, as illustrated in Fig. 3.2.

Fig. 3.3 shows the conditional posterior CRLB and the conditional MSE. It is clearly shown that the conditional PCRLB gives a lower bound on the conditional MSE that an estimator can achieve. It is also clear that the conditional PCRLB and the conditional MSE follow the same trend.

3.5.2 Weakly UNGM results

In Section 3.5.1, the choice of parameters for the UNGM makes it highly nonlinear, so that the MSE of the particle filter does not converge to the conditional PCRLB. In Experiment II, we set $\beta = 0.1$, implying a much smaller nonlinear component in the state equation, and set the measurement noise variance as

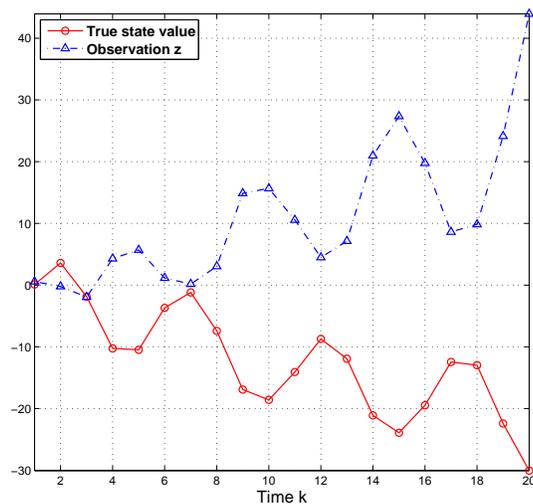


Figure 3.1: Plot of the true state x_k and observations z_k

$\sigma_v^2 = 0.01$, meaning a much higher signal to noise ratio (SNR) for the observation. We keep other parameters the same as in Experiment I. In such a case, the UNGM is weakly nonlinear. As illustrated in Fig. 3.4, the EKF achieves a much better tracking performance than in Experiment I, but the particle filter still outperforms the EKF due to the nonlinearity inherent in this problem. The conditional PCRLB and MSE in Experiment II are shown in Fig. 3.5. As we can see, the gap between the conditional MSE and the conditional PCRLB is much smaller than that in Experiment I.

3.5.3 Conditional PCRLB vs Unconditional PCRLB

In this experiment, we set the parameters in the UNGM the same as those in Experiment I, and compare the conditional and unconditional PCRLBs in Fig. 3.6. The conditional PCRLB and conditional MSE are drawn based on a particular realization of the measurement $\mathbf{z}_{1:k}$, and the unconditional PCRLB is obtained by taking the expectation with respect to both the measurements $\mathbf{z}_{1:k}$ and states $\mathbf{x}_{0:k}$. It can be seen that the conditional PCRLB is much tighter than the unconditional PCRLB for the conditional MSE, and it follows the trends of the conditional MSE more faithfully, since the proposed bound utilizes the available

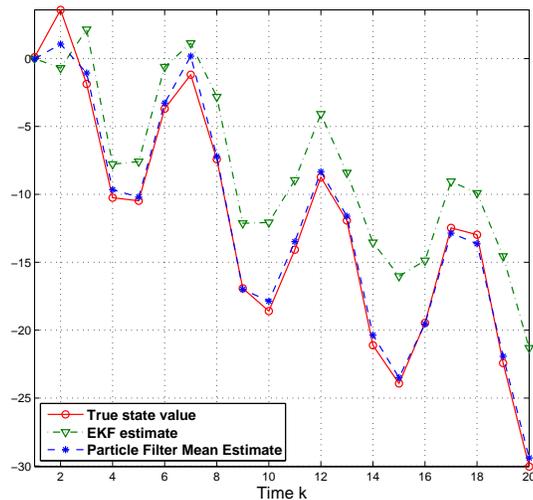


Figure 3.2: Plot of filtering results by Extended Kalman filter and by particle filter for Example I

measurement information. As a result, the conditional PCRLB can be used as a criterion for managing sensors dynamically for the next time step so that its value is minimized.

3.5.4 Exact Conditional PCRLB vs Its Recursive Approximation

In order to recursively calculate the conditional PCRLB, the top-left sub-matrix of $I_A(\mathbf{x}_{0:k}|\mathbf{z}_{0:k})$ is replaced by its expectation in Equation (3.34). This might cause propagation of errors due to approximation. Since it is very difficult to analyze the cumulative error theoretically, an experiment is designed to illustrate the approximation errors. In this experiment, the parameters in the UNGM are the same as those in Experiment I. In Fig. 3.7, the approximate conditional PCRLB evaluated based on Theorem 1 and the approximate recursive method provided by Approximation 1 is compared to the exact conditional PCRLB evaluated using Theorem 1 alone. In the evaluation of the exact conditional PCRLB, by using particle filters, we calculate the complete matrix $I_A(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})$ first, then $L_A^{-1}(\mathbf{x}_k|\mathbf{z}_{1:k})$ can be obtained from the lower-right sub-matrix of $I_A^{-1}(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})$. It

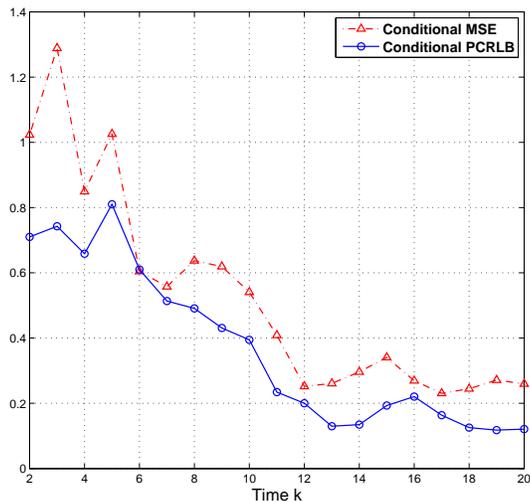


Figure 3.3: Plot of conditional posterior CRLB and conditional MSE for Example I

is clear from Fig. 3.7 that the error propagation is not severe even for the highly nonlinear filtering problem. Further, as time increases, the difference between the exact conditional PCRLB and its recursive approximation is getting smaller. Note that the recursive approach requires much less computational effort.

3.5.5 Unconditional PCRLB with Renewal Strategy

To show the difference between the conditional PCRLB and the unconditional PCRLB with renewal strategy, we choose the following system equations in the numerical example

$$\begin{aligned}
 x_{k+1} &= x_k^2 + u_k \\
 z_k &= x_k + v_k
 \end{aligned}
 \tag{3.90}$$

where x_k and z_k are both scalars, and x_0 , u_k , and v_k are independent and identically distributed (i.i.d.) Gaussian random variables with zero mean and unit variance. From Lemma 1, we know that the conditional PCRLB $L(x_{k+1}|z_{1:k})$ and the PCRLB with renewal strategy J_{k+1} are different if and only if $L_A(x_k|z_{1:k})$

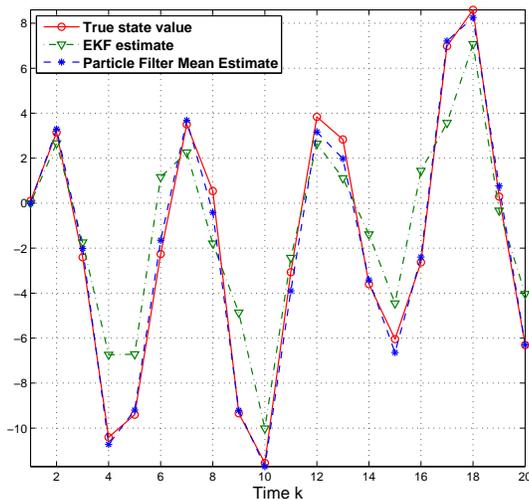


Figure 3.4: Plot of filtering results by Extended Kalman filter and by particle filter for Example II

and \tilde{J}_k are different. For simplicity, we only consider the case of $k = 1$ in the experiment. According to Definition 5, we have

$$\begin{aligned} I_A(x_{0:1}|z_1) &= E_{p(x_{0:1}|z_1)}[-\Delta_{x_{0:1}}^{x_{0:1}} \log p(x_{0:1}|z_1)] \\ &= \begin{bmatrix} a & b \\ b & c \end{bmatrix} \end{aligned} \quad (3.91)$$

With the model used in this experiment and according to Definition 6, the auxiliary FIM $L_A(x_1|z_1) = c - b^2/a$, where

$$\begin{aligned} a &= 1 - 2E_{p(x_{0:1}|z_1)}\{x_1\} + 6E_{p(x_{0:1}|z_1)}\{x_0^2\} \\ b &= -2E_{p(x_{0:1}|z_1)}\{x_0\} \\ c &= 2 \end{aligned} \quad (3.92)$$

The evaluation of $L_A(x_1|z_1)$ can be obtained with the help of particle filters. For the unconditional PCRLB with renewal strategy, at $k = 1$ after the re-initialization,

$$\begin{aligned} \tilde{J}_1 &= E_{p(x_1|z_1)}[-\Delta_{x_1}^{x_1} \log p(x_1|z_1)] \\ &= E_{p(x_1|z_1)}[-\Delta_{x_1}^{x_1} \log p(z_1|x_1)p(x_1)] \\ &= 1 + E_{p(x_1|z_1)}\{-\Delta_{x_1}^{x_1} \log p(x_1)\} \end{aligned} \quad (3.93)$$

3.5 Simulation Results for Comparison

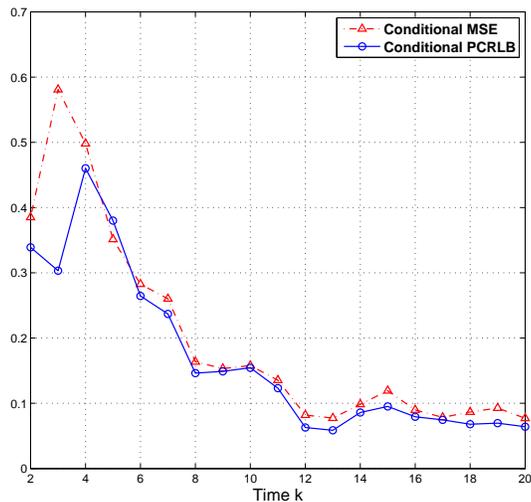


Figure 3.5: Plot of conditional posterior CRLB and conditional MSE for Example II

Given the system Equation (3.90), the PDF of x_1 can be derived

$$\begin{aligned}
 p(x_1) &= \frac{1}{2\pi} e^{-\frac{x_1^2}{2}} \int_0^\infty t^{-\frac{1}{2}} e^{-\frac{t^2}{2} + (x_1 - \frac{1}{2})t} dt \\
 &= \frac{1}{\pi} e^{-\frac{x_1^2}{2}} g(x_1)
 \end{aligned} \tag{3.94}$$

where due to the change of variable,

$$g(x_1) \triangleq \int_0^\infty e^{-\frac{t^4}{2} + (x_1 - \frac{1}{2})t^2} dt \tag{3.95}$$

Finally, we have

$$\Delta_{x_1}^{x_1} \log p(x_1) = -1 + \frac{\Delta_{x_1}^{x_1} g(x_1)}{g(x_1)} - \left[\frac{\nabla_{x_1} g(x_1)}{g(x_1)} \right]^2 \tag{3.96}$$

$p(x_1|z_1)$ and $p(x_{0:1}|z_1)$ are posterior PDFs, which can be calculated from the particle filter. So given a particular measurement z_1 , the value of $L_A(x_1|z_1)$ and \tilde{J}_1 through numerical simulation can be obtained.

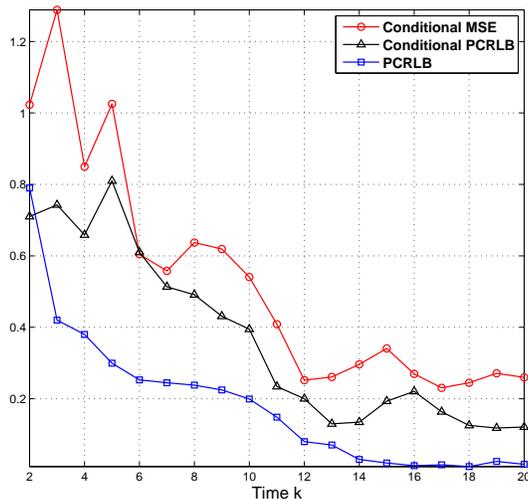


Figure 3.6: Comparison of conditional and unconditional PCRLBs for Example III.

Table 3.1: Comparison between $L_A(x_1|z_1)$ and \tilde{J}_1

z_1	-1.1414	2.3827	-0.0536	1.3337	-0.4035	0.9550	-0.7795	0.5070	1.2737	-1.9947
$L_A(x_1 z_1)$	1.9988	1.9594	1.9955	1.9998	1.9989	1.9977	1.9794	1.9763	1.9972	1.9955
\tilde{J}_1	1.8436	1.3275	1.7662	1.5069	1.7493	1.5863	1.8203	1.6576	1.5560	1.8769

The simulation results are shown in Table 3.1. Given a variety of measurements z_1 's, it is clear that $L_A(x_1|z_1)$ have different values from \tilde{J}_1 . It can also be seen that $L_A(x_1|z_1)$ is greater than \tilde{J}_1 , which indicates that in this particular case the conditional PCRLB is lower than the PCRLB that renews the prior at each time.

3.6 Discussion

In this chapter, we presented the new notation of PCRLB, which is conditioned on the actual past measurement realizations and is, therefore, suitable for online adaptive sensor management. The exact conditional PCRLB and its approximate recursive evaluation formula were theoretically derived. Further, the sequential Monte Carlo approximation for this bound were proposed to provide a conve-

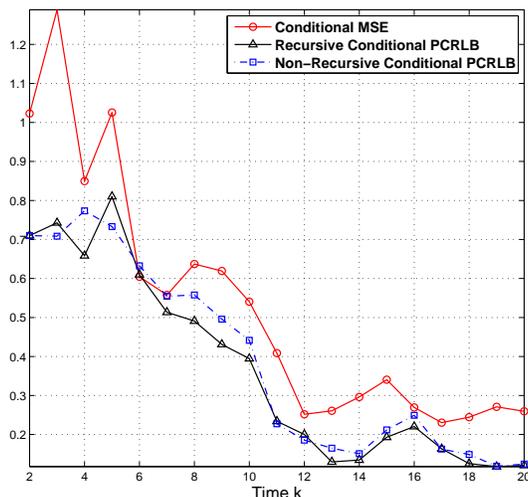


Figure 3.7: Comparison of conditional PCRLB between the one with error propagation and the one without error propagation

nient numerical evaluation solution, as a by-product of the particle filtering process. The conditional PCRLB was compared to existing measurement dependent PCRLBs and shown to be different from them.

Simulation results were provided to demonstrate the effectiveness of the conditional PCRLB in providing online estimation performance prediction, as opposed to the unconditional PCRLB. And the conditional PCRLB derived in this chapter provides an approach to recursively predict the MSE one-step ahead. It can be extended to multi-step ahead cases in the future.

The applications of the proposed bound will be numerous. One possible application area will be a variety of sensor management problems in sensor networks. Choosing the most informative set of sensors will improve the tracking performance, while at the same time reduce the requirement for communication bandwidth and the energy needed by sensors for sensing, local computation and communication. In the next chapter, we show one application of the proposed bound for the sensor selection problems for target tracking in sensor networks.

Chapter 4

Sensor Selection for Target Tracking in Sensor Networks

4.1 Motivation

In sensor networks, the sensors are used to gain information about the kinematic state (moving angle, position and velocity, etc.) of moving targets. The problem of sensor selection for target tracking in sensor networks is to determine the optimal way to select a subset of sensors over time to minimize a cost function considering the constraints, which might be sensor lifetime, bandwidth, communication, etc. As illustrated in Figure 4.1, typically, there is a fusion center that collects the information from active sensors via a data-link with low and/or time-varying bandwidth, and sends commands to sensors. There are some other practical concerns such as limited sensing resources per target, sensor energy consumption, etc. Due to these considerations, the fusion center must dynamically decide which sensor's data are the most valuable to transfer to the tracking system under the currently available data link capacity constraint during each measurement interval. Or, more generally the fusion center needs to decide on how the communication bandwidth should be allocated among the sensors so that the most informative data are transmitted to it as shown in Figure 4.1. The sensor selection scheme is usually based on the kinematic state of the target, error covariances of the estimators or some other predictive information about the target.

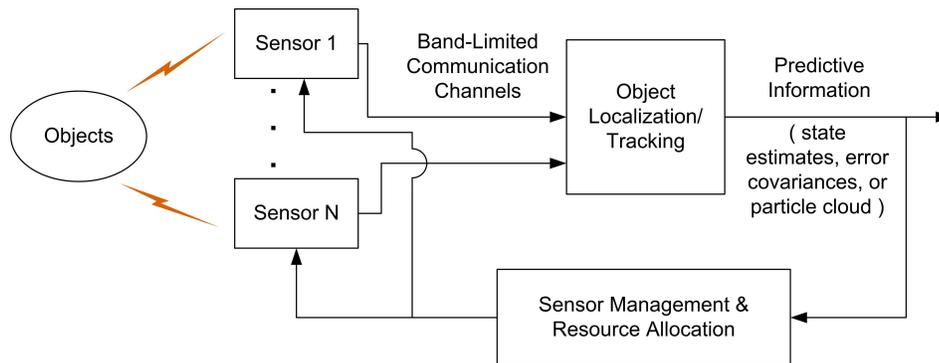


Figure 4.1: Sensor management based on feedback from recursive estimator.

There are several existing approaches to solve sensor management for target tracking. The state of art approach is based on information theoretic measures, such as entropy, relative entropy and Rényi Divergence. In the context of Bayesian estimation, a good measure of the quality of a sensing action is the reduction in entropy of the posterior distribution that is expected to be induced by the measurement. Therefore, information theoretic methods choose the sensing action that maximizes the expected gain in information. Authors in [33] focus on using the expected change in Shannon entropy when tracking a single target. In [4, 34], authors have compared several sensor selection approaches involving entropy and relative entropy. Kreucher et al. [35, 36] have proposed sensor management schemes that maximize the Rényi divergence between the current target state probability density and the density after a new measurement arrives. In [37, 38], sensors are selected that maximize the mutual information between the sensor measurements and the target state. But one problem with the information theoretic measures based approaches is that computational complexity of the mutual information or Rényi divergence is large, especially when the number (N_s) of sensors to be selected at each step is large. If the sensors provide quantized data, it can be shown that the computational complexity of the mutual information is exponential in N_s , whereas the complexity of PCRLB or conditional PCRLB is linear in N_s [39]. If the sensors provide analog data, it could be shown [40] that

the computation of the mutual information involves a $N_s \times n_z$ fold integration, where n_z is the dimensionality of the sensor measurements, whereas the complexity of the recursive conditional PCRLB involves only a n_z fold integration. This fact makes the information measures based sensor management impractical when N_s is large.

Another sensor management approach is based on PCRLB. The PCRLB provides a theoretical performance limit of any estimator for a nonlinear filtering problem. Tichavsky et al. [5] derived an elegant recursive approach to calculate the sequential PCRLB for a general multi-dimensional discrete-time nonlinear filtering problem. This algorithm makes it possible to obtain the PCRLB sequentially and in real time. However, the PCRLB is determined only by the system dynamic model, system measurement model and the prior knowledge regarding the system state at the initial time. As a result, sensor management problems solved by using PCRLB do not utilize the specific realization of the measurements so that the PCRLB does not reflect the filtering performance for the system state realization very faithfully.

Some attempts have been made in the literature to include the information obtained from measurements by incorporating the tracker's information into the calculation of the modified PCRLB. In [20], a renewal strategy has been used to restart the recursive PCRLB evaluation process, where the initial time is reset to a more recent past time, so that the prior knowledge of the initial system state is more useful and relevant to the sensor management problem. This approach is extended in [21] to incorporate sensor deployment and motion uncertainties, and to manage sensor arrays for multi-target tracking problems in [22, 23]. For the renewal strategy proposed in [20], there exists an intrinsic difficulty of calculating the PCRLB from the filtering results, which may incur large errors and discrepancy, especially in a highly nonlinear and non-Gaussian system.

In [24], the authors include the measurement history to calculate the modified PCRLB in an ad-hoc manner for the adaptive radar waveform design method used for target tracking. However, the heuristically modified PCRLB does not yield the exact conditional PCRLB. In [25], for nonlinear target tracking problems, an algorithm is developed to select and configure radar waveforms to minimize the predicted MSE in the target state estimate, which is the expectation of

the squared estimation error over predicted states and observations given a past history of measurements. The predicted MSE is, in general, computationally intractable, so it has been approximated by the covariance update of the unscented Kalman filter.

In this chapter, we consider the application of conditional PCRLB to the problem of tracking a single target traversing through a sensor field and develop an approach for sensor selection for this task. In addition to analog sensor data, we also investigate the sensor selection problems with quantized measurement data for the situations where the bandwidth for transmitting the data between the sensors and the fusion center is constrained.

4.2 Sensor Selection Approaches

Sensor networks consist of a large number of small sensor devices that have the capability to take various measurements of their environment. These measurements can include seismic, acoustic, magnetic, IR and video information. Each of these devices is equipped with a small processor and wireless communication antenna and is powered by a battery making it very resource constrained. Typically, sensors are scattered around a sensing field to collect information about their surroundings. For example, sensors can be used in a battlefield to gather information about enemy troops, detect events such as explosions, and track and localize targets. Upon deployment in a field, they form an ad hoc network and communicate with each other and with data processing centers.

Sensor networks are usually intended to last for long periods of time, such as several days or even months. However, due to the limited energy available on board, if a sensor remains active continuously, its energy will be depleted quickly leading to its death. To prolong the network lifetime, sensors alternate between being active and sleeping. There are several sensor selection algorithms to achieve this while still achieving the goal of continuously monitoring the environment. The decision as to which sensor should be activated takes into account a variety of factors such as residual energy, required coverage, or the type of information required. Sensors are selected to do one or multiple missions. These missions can be general and related to the function of the network, such as monitoring

the whole field by ensuring complete coverage, or more specific and application-oriented, such as tracking the movement of a target. At a given time, the system might be required to do multiple missions such as monitoring an event and, at the same time, track a single or multiple moving objects. Sensor selection schemes are used to allocate the sensor resources to different tasks while at the same time maximize the tracking or detection performances.

Here, we consider two existing approaches and one novel approach based on conditional PCRLB proposed in Chapter 3.

4.2.1 Information Driven

Zhao et al. [4] proposed a sensor selection scheme for target tracking whose performance is compared with our proposed algorithm in Section 4.5. They consider the problem of selecting a sensor $S^{(j)}$, which provides the greatest improvement in the estimate of a target location. This is solved as an optimization problem defined in terms of information gain and cost. The goal is to improve: (1) detection quality, (2) track quality, (3) scalability, (4) survivability and (5) resource usage.

The proposed scheme selects a single sensor node (the leader) at initial time by predicting the location of the target. The leader is activated and collects the required measurements about the target and run a tracking algorithm. From that point on, the leader selects the next node that it believes to be the most informative and passes its tracking information to it. That node becomes the new leader, collects measurements, and runs the tracking algorithm. This continues as long as needed to track a target. When deciding on the next leader, the current leader considers the information utility value of candidate sensors. This value is based only on available information such as a sensor's location, its modality and the current tracking results. The authors consider two possible definitions of information utility; one based on entropy and another based on a distance measure. Although an entropy based definition is mathematically more precise, it is very difficult to compute in a practical setting. This is because the entropy approach requires knowing a sensor's measurement before making any decision, which is very difficult. With distance based measure, the leader node measures

the utility of other sensors based on how far they are located from the estimated target position. This provides a good approximation of the sensor's utility.

One of the drawbacks of above approach is that its accuracy depends on the quality of the choice of the first leader. If the first leader is not close to the target location, due to an error in prediction, the overall tracking quality might degrade and the whole process might even fail. Also, this scheme only selects a single sensor (leader) at a time, so although it may be energy efficient, it might not provide information that is as good as if more sensors are used. When extending the entropy approach to the scenario where more than one sensor is selected, the computational complexity of the entropy approach becoming a problem for a practical application.

For comparing purpose, we implemented entropy based selection approach. In our simulation, we assume that there exists a fusion center, which collects measurement information from active local sensors, and selects those sensors that maximize the mutual information between the moving target and the sensor measurements. We assume that a particle filter is used to tracking the target and use particles to calculate the numerical value of the mutual information between the target state and the sensor measurements in the upcoming step.

In a particle filter, we know that

$$p(\mathbf{x}_{k+1}|\mathbf{z}_{1:k}) \approx \frac{1}{N} \sum_{i=1}^N \delta(\mathbf{x}_{k+1} - \mathbf{x}_{k+1}^i) \quad (4.1)$$

$$\begin{aligned} p(\mathbf{z}_{k+1}|\mathbf{z}_{1:k}) &= \int p(\mathbf{z}_{k+1}|\mathbf{x}_{k+1})p(\mathbf{x}_{k+1}|\mathbf{z}_{1:k})d\mathbf{x}_{k+1} \\ &\approx \frac{1}{N} \sum_{i=1}^N p(\mathbf{z}_{k+1}|\mathbf{x}_{k+1}^i) \end{aligned} \quad (4.2)$$

Therefore,

$$\begin{aligned}
 & \text{MI}(\mathbf{z}_{k+1}; \mathbf{x}_{k+1} | \mathbf{z}_{1:k}) \\
 &= \int p(\mathbf{z}_{k+1} | \mathbf{x}_{k+1}) p(\mathbf{x}_{k+1} | \mathbf{z}_{1:k}) \log \frac{p(\mathbf{z}_{k+1} | \mathbf{x}_{k+1})}{p(\mathbf{z}_{k+1} | \mathbf{z}_{1:k})} d\mathbf{x}_{k+1} d\mathbf{z}_{k+1} \\
 &\approx \frac{1}{N} \sum_{i=1}^N \int p(\mathbf{z}_{k+1} | \mathbf{x}_{k+1}^{(i)}) \log \frac{p(\mathbf{z}_{k+1} | \mathbf{x}_{k+1}^{(i)})}{p(\mathbf{z}_{k+1} | \mathbf{z}_{1:k})} d\mathbf{z}_{k+1} \\
 &\approx \frac{1}{N} \sum_{i=1}^N \int p(\mathbf{z}_{k+1} | \mathbf{x}_{k+1}^i) \log \frac{p(\mathbf{z}_{k+1} | \mathbf{x}_{k+1}^i)}{\frac{1}{N} \sum_{l=1}^N p(\mathbf{z}_{k+1} | \mathbf{x}_{k+1}^l)} d\mathbf{z}_{k+1} \quad (4.3)
 \end{aligned}$$

The above equation could be used for selecting the best sensor each at a time. However, it is not difficult to derive the equation for selecting more than one best sensors. The mutual information for a two-sensor case becomes

$$\begin{aligned}
 & \text{MI}(\mathbf{z}_{k+1}^{S_a}, \mathbf{z}_{k+1}^{S_b}; \mathbf{x}_{k+1} | \mathbf{z}_{1:k}) \\
 &\approx \frac{1}{N} \sum_{i=1}^N \int p(\mathbf{z}_{k+1}^{S_a}, \mathbf{z}_{k+1}^{S_b} | \mathbf{x}_{k+1}^i) \log \frac{p(\mathbf{z}_{k+1}^{S_a}, \mathbf{z}_{k+1}^{S_b} | \mathbf{x}_{k+1}^i)}{p(\mathbf{z}_{k+1}^{S_a}, \mathbf{z}_{k+1}^{S_b} | \mathbf{z}_{1:k})} d\mathbf{z}_{k+1}^{S_a} d\mathbf{z}_{k+1}^{S_b} \quad (4.4)
 \end{aligned}$$

where $\mathbf{z}_{k+1}^{S_a}$ and $\mathbf{z}_{k+1}^{S_b}$ represent the measurement taking from sensor a and b respectively.

For nonlinear/non-Gaussian systems, there is no analytical closed-form expression due to the integration over \mathbf{z}_{k+1} in the above equation. However, if we know the target state at $k+1$, \mathbf{x}_{k+1} , then according to the measurement equation (3.5), the measurement at time $k+1$ can be estimated from the predicted state $\hat{\mathbf{x}}_{k+1}$. Then,

$$p(\mathbf{z}_{k+1}^{S_a, S_b} | \mathbf{x}_{k+1}) \approx \frac{1}{M} \sum_{m=1}^M \delta(\mathbf{z}_{k+1}^{S_a, S_b} - \mathbf{z}_{k+1}^{(m)}) \quad (4.5)$$

where $\mathbf{z}_{k+1}^{S_a, S_b} \triangleq (\mathbf{z}_{k+1}^{S_a}, \mathbf{z}_{k+1}^{S_b})$, and M is the number of measurement samples taken from the measurement space at time $k+1$. Then the mutual information can be calculated according to the following expression:

$$\begin{aligned}
 & \text{MI}(\mathbf{z}_{k+1}^{S_a}, \mathbf{z}_{k+1}^{S_b}; \mathbf{x}_{k+1} | \mathbf{z}_{1:k}) \\
 &\approx \frac{1}{N} \sum_{i=1}^N \frac{1}{M} \sum_{m=1}^M \log \frac{p(\mathbf{z}_{k+1}^m | \mathbf{x}_{k+1}^i)}{\frac{1}{N} \sum_{l=1}^N p(\mathbf{z}_{k+1}^m | \mathbf{x}_{k+1}^l)} \quad (4.6)
 \end{aligned}$$

We must note that for multiple sensors, the mutual information can not be decoupled, which means that the mutual information between the target state and the measurements taken from more than one sensor is not equal to the summation of the mutual information between the target and each sensor measurement individually:

$$\text{MI}(\mathbf{z}_{k+1}^{S_a}, \mathbf{z}_{k+1}^{S_b}; \mathbf{x}_{k+1} | \mathbf{z}_{1:k}) \neq \text{MI}(\mathbf{z}_{k+1}^{S_a}; \mathbf{x}_{k+1} | \mathbf{z}_{1:k}) + \text{MI}(\mathbf{z}_{k+1}^{S_b}; \mathbf{x}_{k+1} | \mathbf{z}_{1:k}) \quad (4.7)$$

Therefore, the computation complexity of the information based approach is $O(NM)$, assuming the equal sample size M for each sensor, and N is the number of particles used by the fusion center. According to Equation (4.5), it can be seen that the number of generated samples should scale exponentially as N_s increases to guarantee an accurate estimation of the mutual information, where N_s is the number of active sensors.

It can also be seen that maximizing the mutual information is equivalent to minimizing the conditional entropy for the target state \mathbf{x}_{k+1} given the measurement $\mathbf{z}_{1:k}$ and \mathbf{z}_{k+1} . Here the existing measurements $\mathbf{z}_{1:k}$ are considered as realizations, while \mathbf{z}_{k+1} is considered as a random variable (vector).

$$\begin{aligned} & \text{MI}(\mathbf{z}_{k+1}, \mathbf{x}_{k+1} | \mathbf{z}_{1:k}) \\ &= H(\mathbf{x}_{k+1} | \mathbf{z}_{1:k}) - H(\mathbf{x}_{k+1} | \mathbf{z}_{1:k}, \mathbf{z}_{k+1}) \end{aligned} \quad (4.8)$$

The first term in the above equation, $H(\mathbf{x}_{k+1} | \mathbf{z}_{1:k})$, has the same value for all the different sensor selection solutions. Therefore, maximizing $\text{MI}(\mathbf{z}_{k+1}, \mathbf{x}_{k+1} | \mathbf{z}_{1:k})$ is equivalent to minimizing $H(\mathbf{x}_{k+1} | \mathbf{z}_{1:k}, \mathbf{z}_{k+1})$.

4.2.2 Nearest Neighbor

The nearest neighbor algorithm always updates the tracking results with the measurement closest to the predicted state.

The cost function for the nearest neighbor approach is:

$$\text{dist}^2(S^{(j)}, \hat{\mathbf{x}}_{k+1} | \mathbf{z}_{1:k}) = (S_x^{(j)} - \hat{x}_{k+1} | \mathbf{z}_{1:k})^2 + (S_y^{(j)} - \hat{y}_{k+1} | \mathbf{z}_{1:k})^2 \quad (4.9)$$

In the above equation, $(S_x^{(j)}, S_y^{(j)})$ represents the position for the sensor node j . $(\hat{x}_{k+1}, \hat{y}_{k+1} | \mathbf{z}_{1:k})$ represents the predicted target position.

The sensors that minimize the above cost function will be selected for the tracking task in the next time step $k+1$. Obviously, the nearest neighbor approach is a heuristic method. The advantage of nearest neighbor approach is that it is easy to implement and very fast. But for some kinds of sensors, such as bearing sensors, the sensor closest to the predicted target state is not always the best sensor to minimize the tracking error, which is shown in the simulations later.

4.2.3 Conditional PCRLB

The PCRLB considers the measurements as random vectors, and at any particular time k , the bound is calculated by taking the average of both the measurements and the states up to time k . In practice, besides the two system equations, some of the measurements are available. More particularly, the measurements up to time $k-1$, $\mathbf{z}_{1:k-1}$, which provide extra information beyond the two dynamic equations. The conditional PCRLB utilizes the information contained in the available measurements, and it gives us more accurate indication on the performance of the estimator at the upcoming time than the regular PCRLB.

The conditional PCRLB can be used as a criterion to select the sensors for target tracking in sensor networks, since it can provide a tracking performance lower bound for the sensor to be selected. The lower the conditional PCRLB is, the more is the potential for those sensors to provide more informative measurements to reduce the MSE, especially for the cases where the MSE can reach or is close to the conditional PCRLB.

The conditional PCRLB is a matrix if the state \mathbf{x}_k is a vector. For tracking problems, we are more interested in the position of the moving target, so the summation of the position bounds along the x and y axes are chosen as the criterion function to be minimized. We can also choose the determinant or the trace of the conditional PCRLB, but the simulation results are quite similar.

4.3 Target Tracking Model in Sensor Networks

4.3.1 Target Motion Model

In this chapter, we consider a single target moving in a 2-D Cartesian coordinate plane according to a dynamic white noise acceleration model [30]:

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{v}_k \quad (4.10)$$

where the constant parameter \mathbf{F} models the state kinematics

$$\mathbf{F} = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.11)$$

T is the time interval between two consecutive sampling points, and in simulation we set it equal to 1 s. The target state at time k is defined as $\mathbf{x}_k = [x_k \dot{x}_k y_k \dot{y}_k]^T$, x_k and y_k denote the target position and \dot{x}_k , and \dot{y}_k denote the velocity. \mathbf{v}_k is white Gaussian noise with covariance matrix \mathbf{Q} .

4.3.2 Sensor Measurement Model

We assume that a large number of homogenous bearing-only sensors are randomly deployed. There exists a fusion center that is responsible for collecting information from each sensor and providing the estimate of the target state. The fusion center has knowledge about the individual sensors, such as their positions and measurement accuracy. At each time, only a small number of sensors are activated to perform the sensing task and providing their observations to the fusion center. For the sensors providing the analog data, the measurement model is given by

$$\mathbf{z}_k^j = h(\mathbf{x}_k) + \mathbf{w}_k^j = \tan^{-1} \left(\frac{y_k - y^{s_j}}{x_k - x^{s_j}} \right) + \mathbf{w}_k^j \quad (4.12)$$

where \mathbf{z}_k^j is the measurement from sensor j , x^{s_j} and y^{s_j} represent the corresponding position of sensor j , and \mathbf{w}_k^j is the white Gaussian noise with covariance matrix \mathbf{R} .

Considering the situation of sensors providing quantized data, the measurement model is given by

$$\theta_k^j = h(\mathbf{x}_k) + \mathbf{w}_k^j = \tan^{-1} \left(\frac{y_k - y^{s_j}}{x_k - x^{s_j}} \right) + \mathbf{w}_k^j \quad (4.13)$$

$$\mathbf{z}_k^j = \mathcal{Q}(\theta_k^j \bmod 2\pi) \quad (4.14)$$

where θ_k^j is the original sensor measurement. The remainder after θ_k^j is divided by 2π is sent to the quantizer. \mathcal{Q} is a m -bit uniform quantizer on $(-\pi, \pi)$. And \mathbf{z}_k^j is the quantized measurement data.

4.4 Sensor Selection Based on Conditional PCRLB

The system model we presented in Section 4.3.1 is a dynamic system with additive Gaussian noises. The conditional Fisher information matrix $L(\mathbf{x}_{k+1}|\mathbf{z}_{1:k})$ can be recursively calculated accordingly as described in Section 3.3. Here we propose the particle filter approach to evaluate B_k^{11} , B_k^{12} , B_k^{22} , S_k^{11} , S_k^{12} , and S_k^{22} as well as providing the tracking results for the state \mathbf{x}_k .

Given the two equations (4.10) and (4.12), we have

$$\begin{aligned} B_k^{11} &= E_{p_{k+1}^c} \{ -\Delta_{\mathbf{x}_k}^{\mathbf{x}_{k+1}} \log p(\mathbf{x}_{k+1}|\mathbf{x}_k) \} \\ &\approx \frac{1}{N} \sum_{l=1}^N \left([\nabla_{\mathbf{x}_k} f(\mathbf{x}_k)] Q_k^{-1} [\nabla_{\mathbf{x}_k}^T f(\mathbf{x}_k)] \right) \Big|_{\mathbf{x}_k = \mathbf{x}_k^l} \\ &= F Q_k^{-1} F^T \end{aligned} \quad (4.15)$$

where $p_{k+1}^c \triangleq p(\mathbf{x}_{0:k+1}, \mathbf{z}_{k+1}|\mathbf{z}_{1:k})$, $\nabla_{\mathbf{x}_k} f(\mathbf{x}_k) = F$

$$\begin{aligned} B_k^{21} &= E_{p_{k+1}^c} \{ -\Delta_{\mathbf{x}_{k+1}}^{\mathbf{x}_k} \log p(\mathbf{x}_{k+1}|\mathbf{x}_k) \} \\ &\approx -\frac{1}{N} \sum_{l=1}^N \left(Q_k^{-1} \nabla_{\mathbf{x}_k}^T f_k(\mathbf{x}_k) \right) \Big|_{\mathbf{x}_k = \mathbf{x}_k^l} \\ &= -Q^{-1} F \end{aligned} \quad (4.16)$$

As for $B_k^{22,a}$ and $B_k^{22,b}$, we have

$$B_k^{22,a} = Q^{-1} \quad (4.17)$$

4.4 Sensor Selection Based on Conditional PCRLB

$$B_k^{22,b} = \frac{1}{N} \sum_{l=1}^N \left([\nabla_{\mathbf{x}_{k+1}} h(\mathbf{x}_{k+1})] R_{k+1}^{-1} [\nabla_{\mathbf{x}_{k+1}}^T h(\mathbf{x}_{k+1})] \right) \Big|_{\mathbf{x}_{k+1}=\mathbf{x}_{k+1}^l} \quad (4.18)$$

where $\nabla_{\mathbf{x}_k}^T h(\mathbf{x}_k) = [-\frac{y_k - y^{S_j}}{D^{S_j}} \ 0 \ \frac{x_k - x^{S_j}}{D^{S_j}} \ 0]$, $D^{S_j} \triangleq (x_k - x^{S_j})^2 + (y_k - y^{S_j})^2$. The approximations for S_k^{11} , S_k^{21} , and S_k^{22} can be derived similarly.

$$S_k^{11} \approx \frac{1}{N} \sum_{l=1}^N g(\mathbf{x}_{k-1}^l, \mathbf{x}_k^l) = F^T Q^{-1} F \quad (4.19)$$

because

$$\begin{aligned} g(\mathbf{x}_{k-1}, \mathbf{x}_k) &= \\ & [\nabla_{\mathbf{x}_{k-1}} f_{k-1}(\mathbf{x}_{k-1})] Q_{k-1}^{-1} [\nabla_{\mathbf{x}_{k-1}}^T f_{k-1}(\mathbf{x}_{k-1})] \\ & - [\Delta_{\mathbf{x}_{k-1}}^{\mathbf{x}_{k-1}} f_{k-1}(\mathbf{x}_{k-1})] \tilde{\Sigma}^{-1} \Upsilon_{k-1}^{11}(\mathbf{x}_{k-1}, \mathbf{x}_k) \\ & = F^T Q^{-1} F - [\Delta_{\mathbf{x}_{k-1}}^{\mathbf{x}_{k-1}} f(\mathbf{x}_{k-1})] \tilde{\Sigma}^{-1} \Upsilon_{k-1}^{11}(\mathbf{x}_{k-1}, \mathbf{x}_k) \\ & = F^T Q^{-1} F \end{aligned} \quad (4.20)$$

$$(4.21)$$

since $f(\mathbf{x}_{k-1}) = F\mathbf{x}_{k-1}$, we have $\Delta_{\mathbf{x}_{k-1}}^{\mathbf{x}_{k-1}} f(\mathbf{x}_{k-1}) = \mathbf{0}$ and

$$\tilde{\Sigma}^{-1} = \begin{bmatrix} Q^{-1} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & Q^{-1} & \mathbf{0} & \vdots \\ \vdots & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{0} & Q^{-1} \end{bmatrix}_{n_x^2 \times n_x^2} \quad (4.22)$$

and

$$\Upsilon_k^{11} = \begin{bmatrix} \mathbf{x}_{k+1} - F\mathbf{x}_k & \dots & \mathbf{0} \\ \vdots & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{x}_{k+1} - F\mathbf{x}_k \end{bmatrix}_{n_x^2 \times n_x} \quad (4.23)$$

$$S_k^{21} = -Q^{-1} F \quad (4.24)$$

$$S_k^{22,a} = Q^{-1} \quad (4.25)$$

and

$$\begin{aligned} S_k^{22,b} &\approx \frac{1}{N} \sum_{l=1}^N \{ [\nabla_{\mathbf{x}_k} h(\mathbf{x}_k)] R^{-1} [\nabla_{\mathbf{x}_k}^T h(\mathbf{x}_k)] \\ &- \Delta_{\mathbf{x}_k}^{\mathbf{x}_k} h(\mathbf{x}_k) \tilde{\Sigma}_v^{-1} \Upsilon_k^{22,b} \} \Big|_{\mathbf{x}_k=\mathbf{x}_k^l} \end{aligned} \quad (4.26)$$

where

$$\tilde{\Sigma}_{\mathbf{v}}^{-1} = \begin{bmatrix} R^{-1} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & R^{-1} & \mathbf{0} & \vdots \\ \vdots & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{0} & R^{-1} \end{bmatrix}_{(n_x n_z) \times (n_x n_z)} \quad (4.27)$$

and

$$\Upsilon_k^{22,b} = \begin{bmatrix} \mathbf{z}_k - h_k(\mathbf{x}_k) & \dots & \mathbf{0} \\ \vdots & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{z}_k - h_k(\mathbf{x}_k) \end{bmatrix}_{(n_x n_z) \times n_x} \quad (4.28)$$

The above derivation for conditional PCRLB is for one sensor measurement case. For selecting multiple sensors (M_k) on every tracking snapshot at time k , we need to include all the measurements from active sensors to calculate the conditional PCRLB, such that $\mathbf{Z}_k \triangleq \{\mathbf{z}_k^j, j \in M_k\}$. We assume that the sensor measurements are independent from each other conditioned on \mathbf{x}_k . Now the recursive conditional PCRLB can be evaluated with the help of particle filtering.

The derivation of conditional PCRLB for quantized data measurements is similar to that for analog data. The only difference lies in the likelihood function, and when the periodicity of bearings around 2π is taken into account, the likelihood function for each quantization level l can be found by

$$\begin{aligned} Pr\{\mathbf{z}_{k+1}^j = l | \mathbf{x}_{k+1}\} &= \sum_{n=-\infty}^{\infty} Pr\{(l-1)\eta + 2n\pi \\ &< \tan^{-1} \frac{\Delta y_{k+1}^{S_j}}{\Delta x_{k+1}^{S_j}} + \mathbf{w}_{k+1}^j < l\eta + 2n\pi\} \end{aligned} \quad (4.29)$$

$$\begin{aligned} Pr(\mathbf{z}_{k+1}^j = l | \mathbf{x}_{k+1}) &= \sum_{n=-\infty}^{\infty} \left\{ \Phi \left(\frac{l\eta + \beta_{k+1,n}^{S_j}}{\sigma} \right) - \right. \\ &\quad \left. \Phi \left(\frac{(l-1)\eta + \beta_{k+1,n}^{S_j}}{\sigma} \right) \right\} \end{aligned} \quad (4.30)$$

where $\Delta y_{k+1}^{S_j} \triangleq y_{k+1} - y^{S_j}$, and $\Delta x_{k+1}^{S_j} \triangleq x_{k+1} - x^{S_j}$, $\beta_{k+1,n}^{S_j} = 2n\pi - \tan^{-1} \frac{\Delta y_{k+1}^{S_j}}{\Delta x_{k+1}^{S_j}}$, $l = -L/2 + 1, -L/2 + 2, \dots, L/2$, and $L = 2^m$. $\eta = 2\pi/L$, σ is the standard deviation of the measurement noise, Φ is a cumulative Gaussian distribution with mean 0 and variance 1.

The partial derivatives in above equations can be found by

$$\frac{\partial p(\mathbf{z}_{k+1}^j | \mathbf{x}_{k+1})}{\partial x_{k+1}} = \frac{\Delta y_{k+1}^{S_j} \sum_{n=-\infty}^{\infty} \gamma(k+1, n, l, S_j)}{\sqrt{2\pi\sigma} \left[(\Delta x_{k+1}^{S_j})^2 + (\Delta y_{k+1}^{S_j})^2 \right]} \quad (4.31)$$

$$\frac{\partial p(\mathbf{z}_{k+1}^j | \mathbf{x}_{k+1})}{\partial y_{k+1}} = \frac{-\Delta x_{k+1}^{S_j} \sum_{n=-\infty}^{\infty} \gamma(k+1, n, l, S_j)}{\sqrt{2\pi\sigma} \left[(\Delta x_{k+1}^{S_j})^2 + (\Delta y_{k+1}^{S_j})^2 \right]} \quad (4.32)$$

where

$$\gamma(k+1, n, l, S_j) \triangleq e^{-\frac{l\eta + \beta_{k+1,n}^{S_j}}{\sigma}} - e^{-\frac{(l-1)\eta + \beta_{k+1,n}^{S_j}}{\sigma}}$$

Due to quantization, the likelihood function $p(\mathbf{z}_{k+1} | \mathbf{x}_{k+1})$ becomes a probability mass function and the PDF $p(\mathbf{x}_{k+1} | \mathbf{z}_{1:k})$ can be represented approximately by propagating the samples $\{\mathbf{x}_k^{(i)}\}$ from time k to $k+1$ according to the particle filter theory.

$$p(\mathbf{x}_{k+1} | \mathbf{z}_{1:k+1}) \approx \sum_{i=1}^N \omega_k^{(i)} \cdot \delta(\mathbf{x}_{k+1} - \mathbf{x}_{k+1}^{(i)}) \quad (4.33)$$

Therefore, the integrals due to expectation can be converted into summation and further can be evaluated approximately by particle filters only if we know the current PDF $p(\mathbf{x}_k | \mathbf{z}_{1:k})$, which can be easily derived by particle filter theory and represented approximately by the following equation

$$p(\mathbf{x}_k | \mathbf{z}_{1:k}) \approx \sum_{i=1}^N \omega_k^{(i)} \cdot \delta(\mathbf{x}_k - \mathbf{x}_k^{(i)}) \quad (4.34)$$

where N is the number of particles.

For the target tracking problems, we are more concerned with the target position. So we choose the summation of the position bounds along each axis as the cost function for time $k+1$

$$\mathcal{C}_{k+1} = L_{k+1}^{-1}(1, 1) + L_{k+1}^{-1}(3, 3) \quad (4.35)$$

where $L_{k+1} \triangleq L(\mathbf{x}_{k+1} | \mathbf{z}_{1:k})$, and $L_{k+1}^{-1}(1, 1)$ and $L_{k+1}^{-1}(3, 3)$ are the bounds on the MSE corresponding to position coordinates x_{k+1} and y_{k+1} respectively. Those sensors that collectively minimize the above cost function will be activated at the

next time $k + 1$. In this chapter, we use the optimal enumerative search method to determine the combination of sensors, which minimizes the cost function.

$$M_s^{k+1,*} \triangleq \underset{M_s^{k+1} \subset \mathcal{S}}{\operatorname{argmin}} \mathcal{C}_{k+1}(M_s^{k+1}) \quad (4.36)$$

where \mathcal{S} denotes the set containing all the sensors, M_s^{k+1} is a pair of sensors chosen from \mathcal{S} .

4.5 Simulation Results

In this section, the performance of the proposed sensor selection approach in this chapter is evaluated in terms of the MSEs of the state vector. In the simulations, we consider a scenario where 30 homogenous bearing-only sensors are randomly deployed in a 500×500 field. A single target moves in the field for 60 seconds according to the white noise acceleration model (4.10). At each time, two sensors are activated to report the information of the target to fusion center according to Equation (4.12). The measurement noise variance is set to $R = 0.005$, and the system noise covariance matrix Q is chosen as

$$Q = \begin{bmatrix} 0.3333 & 0.5000 & 0 & 0 \\ 0.5000 & 1.0000 & 0 & 0 \\ 0 & 0 & 0.3333 & 0.5000 \\ 0 & 0 & 0.5000 & 1.0000 \end{bmatrix}$$

The prior PDF of the target state is assumed Gaussian with mean $[0 \ 10 \ 0 \ 10]^T$ and covariance $P_0 = \operatorname{diag}(1, 0.5, 1, 0.5)$. For simplicity and illustration purposes, the transition PDF $p(\mathbf{x}_{k+1}|\mathbf{x}_k)$ is chosen as the proposal density function $\pi(\mathbf{x}_k | \mathbf{x}_{0:k-1}, \mathbf{z}_{0:k})$. We implement our approach by using $N = 500$ particles, and 100 Monte Carlo repetitions are performed for each experiment.

For comparison purposes, we also consider three other selection methods. 1) PCRLB with renewal strategy, in which the prior pdf of the target state is updated at each time after we get the state estimate, which uses similar selection criterion as the conditional PCRLB to try to minimize position error; 2) Information-driven approach, where the selection schemes aim to minimize the entropy of the measurement; and 3) Nearest neighbor approach, where the sensors that are closest to the predicted position of the target are selected.

4.5.1 Sensor Selection with Analog Data

Figures 4.2 - 4.5 demonstrate the tracking results where true target trajectory and estimated trajectories by different sensor selection methods are compared. We can see that the proposed selection method achieves more accurate tracking results. Figure 4.6 and Figure 4.7 show the MSEs of target position in x and y coordinates respectively. The proposed sensor selection method by minimizing the conditional PCRLB offers a significant error reduction for most of the tracking time compared to other existing methods.

For analog data, Table 4.1 shows the time complexities of different sensor selection approaches. It can be seen that nearest neighbor, PCRLB with renewal prior and conditional PCRLB has the same order of time complexity, which is liner in the number of particles and the number of active sensors. However, information based approach has a much higher order of time complexity than the other three.

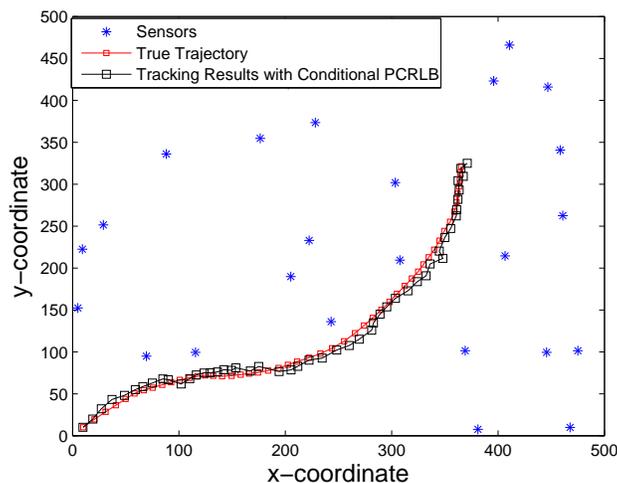


Figure 4.2: Conditional PCRLB Tracking results with analog data

4.5.2 Sensor Selection with Quantized Data

In the simulation, we choose a quantization of measurement with $m = 5$ bits for the Equation (4.14). Figures 4.8 - 4.11 show the tracking results under the same

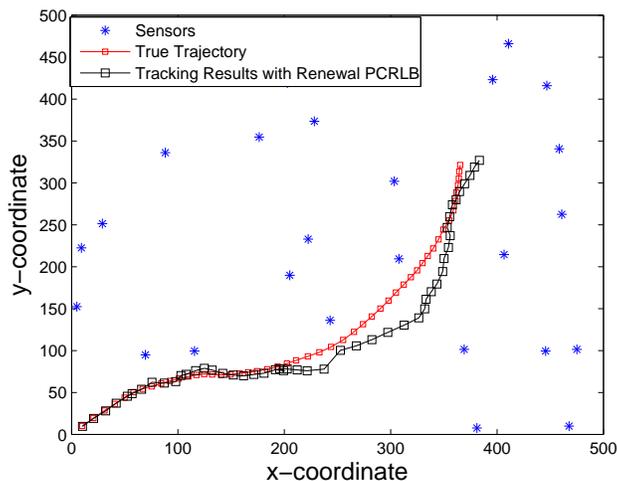


Figure 4.3: Renewal PCRLB Tracking results with analog data

Table 4.1: Comparison of average CPU computational times (Analog data with two active sensors selected at each time, 30 time steps with $N = 300$)

	Time (s)
C-PCRLB	5.102285
PCRLB with renewal prior	3.642564
Nearest neighbor	1.159977
Mutual information	276.001549

simulation configuration as the analog data. The MSE results are illustrated in Figures 4.12 and 4.13. It can be seen that the proposed method shows better tracking results than the nearest neighbor, information-driven and PCRLB with renewal strategy approaches.

In the experiment, we have observed that due to the quantization procedure, the Fisher information is smaller and the conditional PCRLB is higher compared to the analog data. The tracking error is also increased.

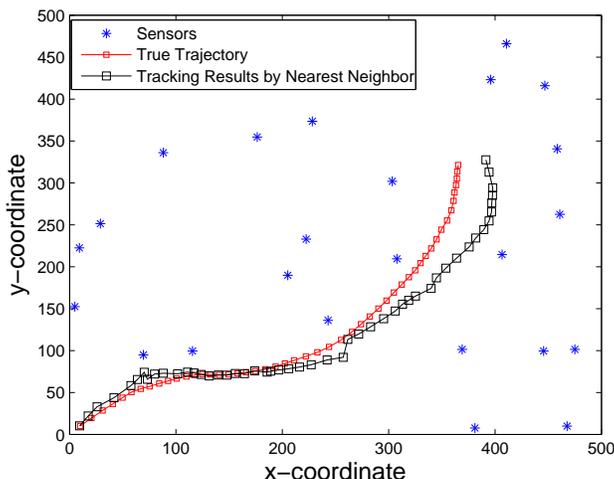


Figure 4.4: Nearest Neighbor Tracking results with analog data

4.6 Discussion

In this chapter, we considered a sensor selection problem for tracking a single target in sensor networks. The conditional PCRLB method is approximated recursively by using a particle filter without the knowledge of future measurements. Those sensors that collectively minimized the cost function established on conditional PCRLB are activated, while other sensors are in the idle state. Simulation results for both analog and quantized measurement data were presented to illustrate the improved performance of our proposed sensor selection approach, which outperforms other existing methods.

We also use conditional PCRLB as one of the objectives for the multi-objective target tracking problems. Such kind of problems involves simultaneously maximizing target tracking accuracy and minimizing querying cost. The querying cost could consist of computation, sensing range, communication bandwidth, and energy consumption. The tracking task tends to be sequentially identifying active subset sensor while simultaneously addressing two conflicting objectives: cost and tracking accuracy. For more in-depth information on these topics, the reader is referred to [41]

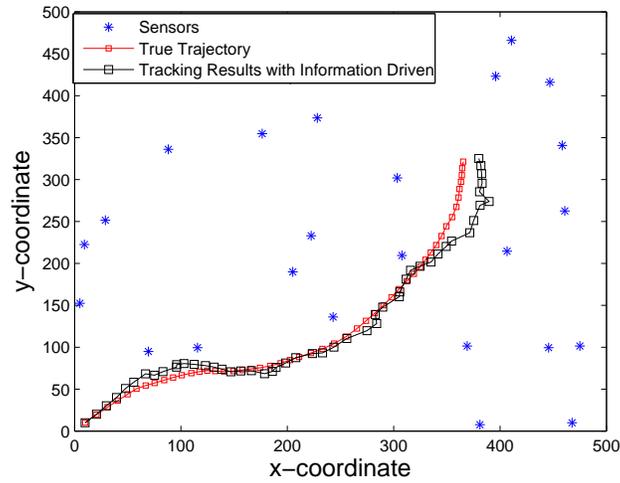


Figure 4.5: Information based Tracking results with analog data

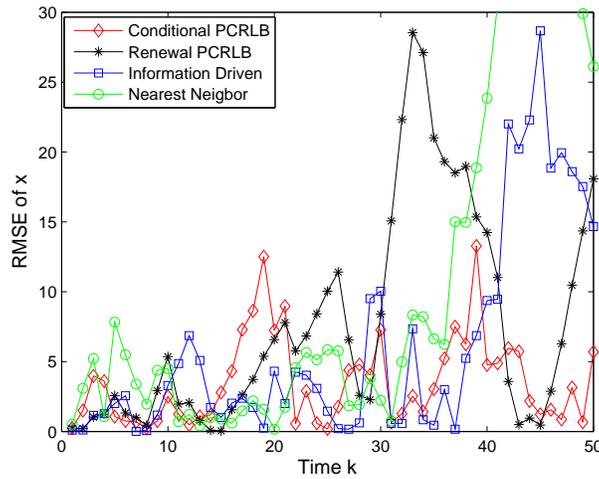


Figure 4.6: Comparison of x-MSEs with analog data

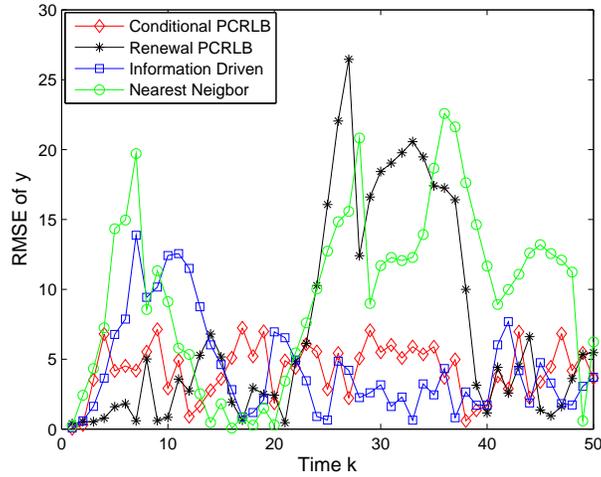
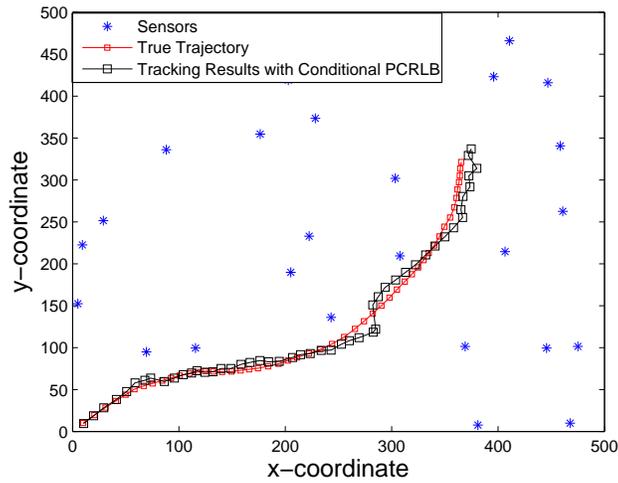
Figure 4.7: Comparison of y -MSEs with analog data

Figure 4.8: Conditional PCRLB Tracking results with quantized data

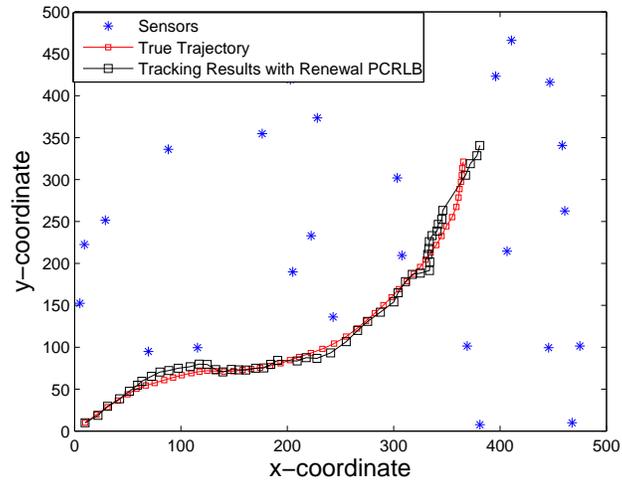


Figure 4.9: Renewal PCRLB Tracking results with quantized data

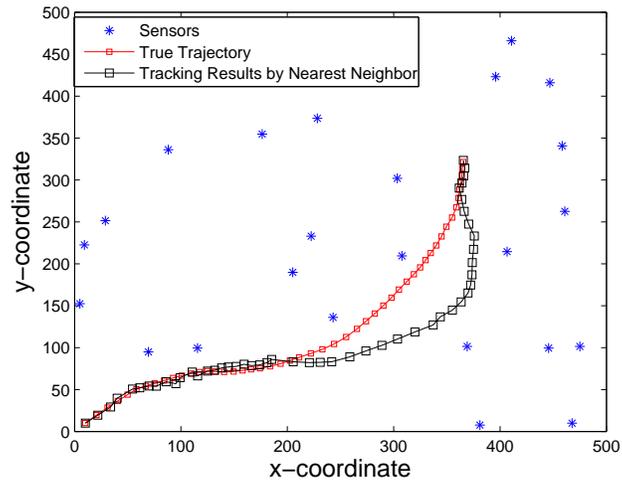


Figure 4.10: Nearest Neighbor Tracking results with quantized data

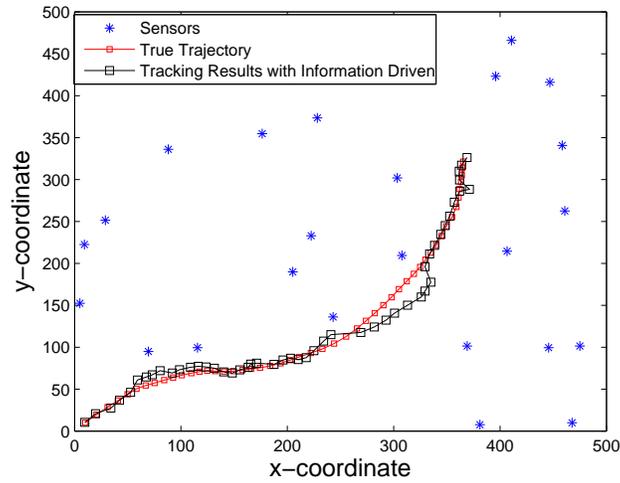


Figure 4.11: Information based Tracking results with quantized data

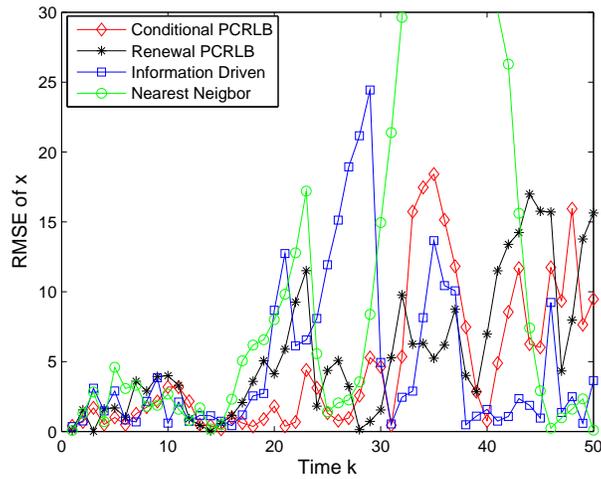
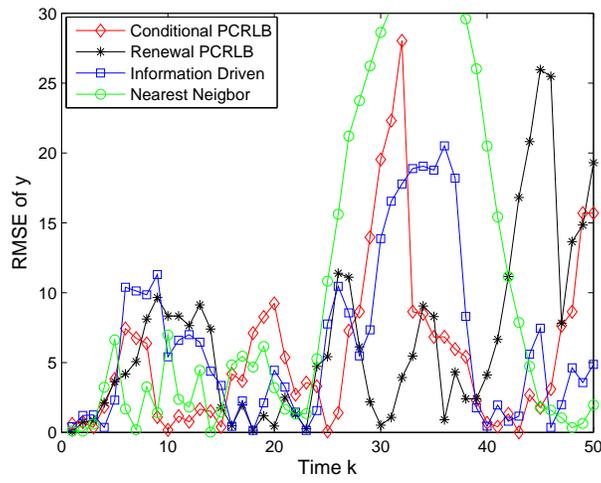


Figure 4.12: Comparison of x-MSEs with quantized data

Figure 4.13: Comparison of y -MSEs with quantized data

Chapter 5

Bandwidth-Efficient Distributed Particle Filter for Target Tracking in Sensor Networks

5.1 Motivation

Wireless sensor networks composed of miniature devices that integrate physical sensing, data processing and communication capabilities present great opportunities for a wide range of applications [42]. The technology lends itself well to surveillance and monitoring tasks, including target tracking. Unfortunately, the sensors used for these tasks are inherently limited, and individually incapable of estimating the target state. However, fusing measurements from multiple sensors for improving tracking performance has been the subject of significant research [30]. The focus has been on combining measurements from sensors (radars, bearing sensors, etc.) individually capable of estimating the target state (position, velocity, etc.).

As opposed to centralized computation based on measurements available from all the sensors [13][43], distributed processing has many advantages: 1) Distributed architecture is more robust. 2) Sensor nodes can have computation power. Therefore, the computations of particle filter can be distributed to sensor nodes. 3) Local estimation results by particle filter need to be compressed before

being transmitted to the fusion center in order to save the communication bandwidth. Vercauteren et al, [44] proposed a collaborative signal processing algorithm for object tracking and classification in sensor networks. In their work, only one sensor is active at any time, which does not fully utilize the power of multiple sensors. Use of distributed Kalman filters for tracking in a sensor network, such as in [45], is based on the linearity and Gaussian assumptions. In this chapter, we present a distributed particle filter (DPF) algorithm to perform sampling-based sequential target tracking over a wireless sensor network efficiently and accurately. In contrast to the centralized method, our approach is to distribute the computation burden and communication burden over the entire sensor network. Each local sensor node is assumed to have enough computing capacity to update its own estimate in parallel based only on its local observations. These partial estimates are then transmitted to the fusion center. Our method is different from [46], which sought to employ a quantization-based method to adaptively encode the local measurements before applying the particle filter. This method needs complicated learning procedure to run the algorithm. Similar to [47], we also propose a method to approximate the local estimate with the parameters of a low dimensional Gaussian Mixture Model(GMM). Instead of transmitting raw estimates of particles, parameters of the GMMs, which approximate the particles to estimate the posterior distribution of the moving object, are transmitted to the fusion center. This approximation scheme significantly reduces communication bandwidth requirement. The difference between our method and that in [47] is that in our method the number of components for each GMM are dynamically selected according to the posterior distribution approximated by the particles, and an optimal fusion method is introduced to fuse the collected GMMs with different number of components.

5.2 Distributed Target Tracking

The problem of single target tracking is considered here assuming that the data association problem has been solved. Let superscript j denote quantities pertaining to the j th sensor. It is assumed that each sensor operates and provides the estimate \mathbf{x}_k of the true target state to the fusion center. Note here the local

estimates $\hat{\mathbf{x}}$'s are in world Cartesian coordinates. The fusion center treats the local estimates as target measurements according to the equation:

$$\mathbf{z}_k^* = \begin{bmatrix} \hat{\mathbf{x}}_{k|k}^1 \\ \vdots \\ \hat{\mathbf{x}}_{k|k}^{N_s} \end{bmatrix} = \begin{bmatrix} I \\ \vdots \\ I \end{bmatrix} \mathbf{x}_k + \begin{bmatrix} \hat{\mathbf{x}}_{k|k}^1 - \mathbf{x}_k \\ \vdots \\ \hat{\mathbf{x}}_{k|k}^{N_s} - \mathbf{x}_k \end{bmatrix} \quad (5.1)$$

where I is the identity matrix with appropriate dimension. This model was introduced in [48] as a universal model for standard distributed fusion. Thus, the target tracking problem at the fusion center is treated as an estimation problem for the state \mathbf{x}_k subject to the target motion model (5.22) based on the sequence of pseudo-measurements given by (5.1). This is a nonlinear estimation problem and its complete solution is given by the posterior PDF $p(\mathbf{x}_k|\mathbf{z}_k^*)$.

5.3 Tracking Based on Particle Filters

Given the process and measurement models (5.22) and (5.25), the recursive Bayesian filtering paradigm provides the *a posteriori* PDF $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ via the prediction and update recursions. $\mathbf{z}_{1:k}$ represents all the available measurement information until and including time k .

Prediction:

$$p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})d\mathbf{x}_{k-1} \quad (5.2)$$

Update:

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) = \frac{p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{z}_{1:k-1})}{p(\mathbf{z}_k|\mathbf{z}_{1:k-1})} \quad (5.3)$$

where the state \mathbf{x}_k evolution is described in terms of the transition probability:

$$p(\mathbf{x}_k|\mathbf{x}_{k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{v}_{k-1})p(\mathbf{v}_{k-1})d\mathbf{v}_{k-1} \quad (5.4)$$

And how the given \mathbf{x}_k fits the available measurement is described as:

$$p(\mathbf{z}_k|\mathbf{x}_k) = \int \delta(\mathbf{z}_k - h_k(\mathbf{x}_k, \mathbf{w}_k))p(\mathbf{w}_k)d\mathbf{w}_k \quad (5.5)$$

Particle filters represent the state PDFs approximately by a set of samples and implement Bayesian recursion directly on the samples instead of dealing

with the exact analytical functional representations of the distributions. Our tracking framework is based on sequential importance sampling (SIS) for particle filtering described in [29][15]. There are two main corresponding steps in particle filters: prediction and update. Resampling is also needed to avoid the degeneracy problem.

We can envisage a general distributed particle filter (GDPF) algorithm, where local sensor nodes draw samples, calculate the importance weights and send them to the fusion center. In this case, the importance weight normalization and resampling are performed at the fusion center. The resampled particles are sent back to each sensor node. In this GDPF approach, not only is there a heavy computation at the fusion center, but also this method requires transmitting large amounts of data from the sensor node to the fusion center. This provides the motivation for the bandwidth efficient distributed particle filter based scheme.

5.4 Gaussian Mixture Approximation to Particle Filter Estimates

In order to reduce the communication cost, we approximate the locally resampled particles by a Gaussian mixture model, and only the parameters of the GMMs are transmitted to the fusion center. The parameters of GMM are learned using an iterative EM algorithm [49].

5.4.1 Expectation-Maximization Algorithm

The Expectation-Maximization (EM) algorithm is a general algorithm for finding the maximum-likelihood estimate (MLE) of the parameters of an underlying distribution where the data are incomplete, have missing values or the likelihood function involves latent variables [50, 51, 52].

EM is an iterative method which alternates between performing an expectation (E) step, which computes the expectation of the log-likelihood evaluated using the current estimate for the latent variables, and a maximization (M) step, which computes parameters maximizing the expected log-likelihood found on the E step. These parameter-estimates are then used to determine the distribution

5.4 Gaussian Mixture Approximation to Particle Filter Estimates

of the latent variables in the next E step. Given a likelihood function $L(\Theta|Z)$, where Θ is the parameter vector, $Z = \{x, y\}$ represents the complete data set, and x represents the incomplete data, y represents the unobserved latent data or missing values. The joint density function is

$$p(Z|\Theta) = p(x, y|\Theta) = p(y|x, \Theta)p(x|\Theta) \quad (5.6)$$

So the complete likelihood function can be written as

$$\mathcal{L}(\Theta|Z) = \mathcal{L}(\Theta|x, y) = p(x, y|\Theta), \quad (5.7)$$

The EM algorithm seeks to find the MLE of the marginal likelihood by iteratively applying the following two steps

Expectation step: Find the expected value of the complete-data log-likelihood $\log p(x, y|\Theta)$ with respect to the unknown data y given the observed data x under the current parameter estimates $\Theta^{(t)}$.

$$Q(\Theta|\Theta^{(t)}) = E_{p(y|x, \Theta^{(t)})} \log \mathcal{L}(\Theta|Z) \quad (5.8)$$

where the superscript t represents the iteration step. In the above equation, $\Theta^{(t)}$ is considered to be a constant, and Θ corresponds to the parameters that ultimately will be estimated in an attempt to maximize the likelihood.

Maximization step: Find the parameter that maximizes this quantity:

$$\Theta^{(t+1)} = \arg \max_{\Theta} Q(\Theta|\Theta^{(t)}) \quad (5.9)$$

Each iteration of the above two steps is guaranteed to increase the likelihood and the EM algorithm is guaranteed to converge to a local maximum of the likelihood function [53].

5.4.2 MLE of Gaussian Mixture Densities Parameters via EM

The finite Gaussian Mixture Model (GMM) is a probabilistic model for density estimation using a mixture of several Gaussian distributions with associated

5.4 Gaussian Mixture Approximation to Particle Filter Estimates

weights. The weights for Gaussian distributions are constrained to have a unit sum [54][55].

$$p(x|\Theta) = \sum_{i=1}^M \alpha_i p_i(x|\theta) \quad (5.10)$$

and

$$\sum_{i=1}^M \alpha_i = 1 \quad (5.11)$$

where the parameters are $\Theta = \alpha_1, \dots, \alpha_M, \theta_1, \dots, \theta_M$, each p_i is a density function parameterized by θ_i , M is the number of components, and $\sum_{i=1}^M \alpha_i = 1$. If we have N observations assumed to be independent, the log-likelihood of the above GMM is given by

$$\log p(x|\Theta) = \sum_{i=1}^N \log \left(\sum_{j=1}^M \alpha_j p_j(x_i|\theta_j) \right) \quad (5.12)$$

The above likelihood function is difficult to optimize because of the summation within the \log function. In order to apply EM to find the MLE of GMM, we could assume that the data set x is incomplete and unobserved data y could inform us which component density generates the data. And if the value of y is known, the likelihood with complete data becomes

$$\log p(x, y|\Theta) = \sum_{i=1}^N \log(\alpha_{y_i} p_{y_i}(x_i|\theta_{y_i})) \quad (5.13)$$

where we assume that $y_i \in 1, \dots, M$ for each i , and $y_i = k$ if i^{th} sample is generated by the k^{th} mixture component.

Given the definition of the hidden variable y_i , the Expectation step in Equation (5.8) can be rewritten as [49]

$$Q(\Theta, \Theta^{(t)}) = \sum_{l=1}^M \sum_{i=1}^N \log(\alpha_l) p(l|x_i, \Theta^{(t)}) + \sum_{l=1}^M \sum_{i=1}^N \log(p_l(x_i|\theta_l)) p(l|x_i, \Theta^{(t)}) \quad (5.14)$$

where $l \in 1, \dots, M$, and $\sum_{i=1}^M p(i|x_j, \Theta^{(t)}) = 1$ for $j \in 1, \dots, N$.

In order to find the update rule for parameter $\Theta_l = \{\alpha_l, \mu_l, \Sigma_l\}$ of each component in GMM, we can take the differentiation of the above equation, and the

5.4 Gaussian Mixture Approximation to Particle Filter Estimates

estimates of the new parameters in terms of the old parameters can be derived as follows

$$\alpha_\ell^{(t+1)} = \frac{1}{N} \sum_{i=1}^N p(\ell | \mathbf{x}_i, \Theta^{(t)}) \quad (5.15)$$

$$\mu_\ell^{(t+1)} = \frac{\sum_{i=1}^N \mathbf{x}_i p(\ell | \mathbf{x}_i, \Theta^{(t)})}{\sum_{i=1}^N p(\ell | \mathbf{x}_i, \Theta^{(t)})} \quad (5.16)$$

$$\Sigma_\ell^{(t+1)} = \frac{\sum_{i=1}^N p(\ell | \mathbf{x}_i, \Theta^{(t)}) (\mathbf{x}_i - \mu_\ell^{(t+1)}) (\mathbf{x}_i - \mu_\ell^{(t+1)})^T}{\sum_{i=1}^N p(\ell | \mathbf{x}_i, \Theta^{(t)})} \quad (5.17)$$

5.4.3 Dynamic EM for GMM

Applying the EM algorithm to estimate the parameters of GMM requires a known number of GMM components. However, in our tracking problem, we usually do not have prior knowledge of the number of components. We should set a value before applying the EM algorithm. The larger the value that we assign to the number of GMM components, the more accurate the approximation we will obtain, but more bandwidth we will use for transmission.

We introduce a modified EM algorithm here to dynamically select the number of GMM components according to the posterior distribution estimated by the particles. Assume that homogeneous sensors are used, and each sensor at each time can transmit information represented by a GMM with at most N_g components because of bandwidth limitations. We utilize the Kullback-Liebler(KL) distance to merge the GMM components if the KL distance of GMM components is less than a threshold. This will further save the bandwidth.

In probability theory and information theory, the KL distance [56] (also known as information divergence, information gain, relative entropy, etc.) is a non-symmetric measure of the difference between two probability distributions P and Q , which is defined by

$$KL(p||q) = \int p \log \frac{p}{q} \quad (5.18)$$

KL distance measures the expected number of extra bits required to code samples from P when using a code based on Q , rather than using a code based on P . The Kullback-Leibler divergence is well-defined for both discrete and continuous

5.4 Gaussian Mixture Approximation to Particle Filter Estimates

distributions, and furthermore is invariant under parameter transformations. One of its important properties is that KL distance is always non-negative. So in our work, we calculate pairwise KL distances for all the Gaussian components, and it has a closed form expression

$$KL(N(\mu_1, \Sigma_1) || N(\mu_2, \Sigma_2)) = \frac{1}{2} \left\{ \log \frac{|\Sigma_2|}{|\Sigma_1|} + Tr(\Sigma_1 (\Sigma_2^{-1} - \Sigma_1^{-1})) + Tr(\Sigma_2^{-1} (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T) \right\} \quad (5.19)$$

The algorithm starts with N_g number of components. If the distance is smaller than a threshold, we decrease N_g by 1, and re-run the EM algorithm until $N_g = 1$, or the KL distances of all the component pairs are greater than the threshold. The threshold is chosen empirically or according to the bandwidth requirement in the real application. Note that at different tracking instants, we may use a different number of GMM components for each local sensor node.

5.4.4 Best Linear Unbiased Estimators for Centralized Estimation

Each local sensor will send the parameters of the GMM to the fusion center, and the fusion center's task is to fuse the local estimates, which is known as estimation fusion. The important thing here is that we consider the local sensor GMM estimation as a kind of measurements of the true target state \mathbf{x}_k . Faced with the difficulties to determine the optimal estimation given the GMMs from the local sensor, we utilized the best linear unbiased estimation (BLUE) [57] as a suboptimal estimator to fuse the GMM components to perform centralized estimation at the fusion center.

BLUE is a linear estimator which is unbiased and has minimum variance among all other linear estimators. To employ BLUE, we assume that the fused estimator is unbiased and is a linear combination of the GMMs. Now we can determine the target state with the knowledge of only the first and second moments of the PDF.

We rewrite (5.1) as:

$$\mathbf{z}_k^* = \begin{bmatrix} \mu_{k|k}^{1,1} \\ \mu_{k|k}^{1,2} \\ \vdots \\ \mu_{k|k}^{1,q_1} \\ \vdots \\ \mu_{k|k}^{N_s, q_{N_s}} \end{bmatrix} = M\mathbf{x}_k + \begin{bmatrix} \tau_k^{1,1} \\ \tau_k^{1,2} \\ \vdots \\ \tau_k^{1,q_1} \\ \vdots \\ \tau_k^{N_s, q_{N_s}} \end{bmatrix} \quad (5.20)$$

where M is the concatenation of the identity matrices, N_s is the total number of sensors participating in tracking, and q_{N_s} is the number of GMM components at sensor N_s . $\mu_{k|k}^{i,j}$ denotes the mean from the j th Gaussian component sent by sensor i at time k , and $\tau_k^{i,j}$ is the corresponding white noise with zero mean and weight covariance $\frac{1}{\alpha_k^{i,j}}\Sigma^{i,j}$. $\alpha_k^{i,j}$ represents the weight of the j th GMM component from sensor i . Then the BLUE is:

$$\hat{\mathbf{x}}_k = (M^T C^{-1} M)^{-1} M^T C^{-1} \mathbf{z}_k^* \quad (5.21)$$

where C is the covariance matrix of $[\hat{\tau}_k^{1,1}, \dots, \hat{\tau}_k^{N_s, q_{N_s}}]^T$.

5.5 Simulation Results

A typical scenario is considered for tracking a moving target based on position measurements from multiple distributed sensors.

5.5.1 Target Motion Model

The target motion is described by the discrete-time nonlinear nearly constant turn(CT) model [58]

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}) + G\mathbf{v}_k \quad k=1,2,\dots \quad (5.22)$$

where the target state vector $\mathbf{x}_k = [x, \dot{x}, y, \dot{y}, \omega]^T$ consists of the position, velocity and the constant turn rate ω ; $\mathbf{v}_k \sim N(0, Q_k)$ is white process noise; and

$$f(\mathbf{x}) = \begin{bmatrix} 1 & \frac{\sin \omega T}{\omega} & 0 & -\frac{1-\cos \omega T}{\omega} & 0 \\ 0 & \cos \omega T & 0 & -\sin \omega T & 0 \\ 0 & \frac{1-\cos \omega T}{\omega} & 1 & \frac{\sin \omega T}{\omega} & 0 \\ 0 & \sin \omega T & 0 & \cos \omega T & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \\ \omega \end{bmatrix} \quad (5.23)$$

$$G = \begin{bmatrix} \frac{1}{2}T^2 & 0 & 0 \\ T & 0 & 0 \\ 0 & \frac{1}{2}T^2 & 0 \\ 0 & T & 0 \\ 0 & 0 & T \end{bmatrix} \quad (5.24)$$

5.5.2 Measurement Model

Although the target state is expressed in Cartesian coordinates, the measurements are usually expressed in polar coordinates of local sensors [59]. Measurements of range and bearing are given by

$$\mathbf{z}_k = h(\mathbf{x}_k) + w_k^i \quad i=1,2,\dots,N \quad (5.25)$$

with

$$h(\mathbf{x}_k) = \begin{bmatrix} r^i \\ b^i \end{bmatrix} = \begin{bmatrix} \sqrt{x_k^2 + y_k^2} \\ \tan^{-1} \frac{y_k}{x_k} \end{bmatrix} \quad (5.26)$$

and white measurement noise $w_k^i \sim N(0, R_k^i)$.

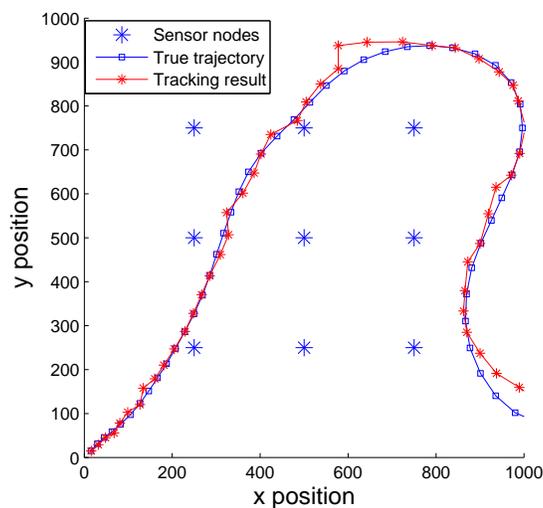


Figure 5.1: Distributed target tracking results

A sensor network scenario as shown in Figure 5.1 is considered to evaluate the performance of the proposed algorithm. Nine sensor nodes are uniformly placed in the 1000×1000 area. The target motion model is described by the nearly

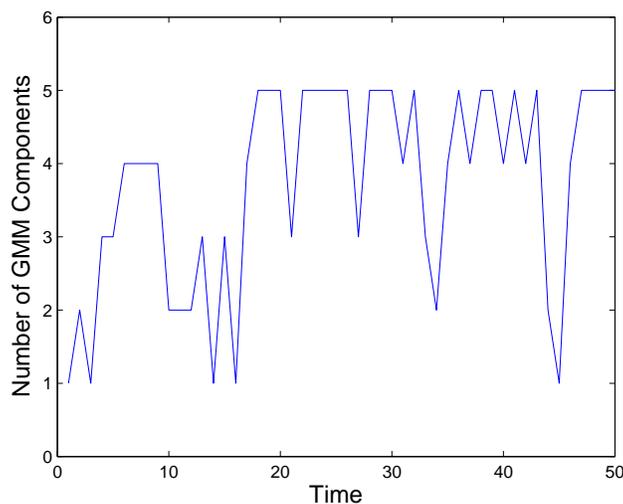


Figure 5.2: Number of GMM components

constant turn model given in Equations (5.22) and (5.23). In the simulation, we use $Q_k = [1, 1, 0.001]$, $T = 1s$, and $R_k = \text{diag}(10, 0.5)$. The importance density for the particle filter is chosen to be the prior distribution $p(\mathbf{x}_k | \mathbf{x}_{k-1})$. Resampling was performed at each iteration to make sure that the particles are independent and identically distributed before running the EM algorithm. 100 Monte Carlo runs were carried out and the position root mean square error (RMSE) was used for comparing the tracking performance.

Figure 5.2 shows the number of components used at different tracking instants at one sensor node. Here we assume that at most $N_g = 5$ components can be transmitted each time by the local sensor to the fusion center due to bandwidth limitations. It is evident that the introduction of the KL distance based method reduces the number of GMM components to be transmitted, thus saving the communication bandwidth of the network.

To evaluate the performance, we increase the number of particles used for estimation by each sensor node from 300 to 1000. Figure 5.3 shows the corresponding average number of bits transmitted to the fusion center with different number of particles. For the general distributed particle filter(GDPF), the number of bits transmitted is proportional to the number of particles. Instead of transmitting particles, sending the parameters of the GMMs incurs much less communication

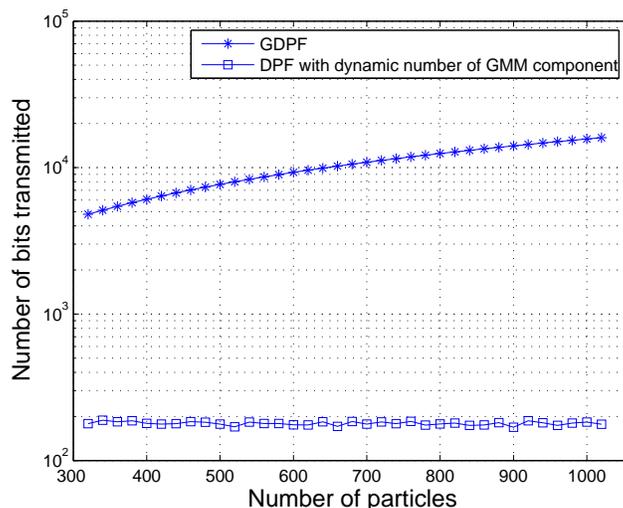


Figure 5.3: Plot of number of particles vs number of bits transmitted

cost. But the accuracy still remains almost the same as the GDPF approach especially when using a larger number of particles as shown in Figure 5.4.

If the number of GMM components at each tracking instant is fixed to be N_g , we can see the substantial saving in bandwidth from Figure 5.5. However, the RMSEs of these two methods are almost the same as shown in Figure 5.6.

5.6 Discussion

In this chapter, we have proposed a distributed target tracking algorithm based on particle filters for sensor networks. Three main contributions of this method are: first, instead of transmitting the raw particles, we use a Gaussian Mixture Model (GMM) to approximate the *a posteriori* distribution obtained from the local particle filters, and only transmit the parameters of the GMM to the fusion center. Second, in order to further save the bandwidth, the number of components of the GMM is dynamically updated according to the posterior distribution at each local sensor. Finally, an optimal rule based on the best linear unbiased estimation (BLUE) method is introduced to fuse the GMM parameters collected from local sensors. Simulation results demonstrate that our approach is accurate

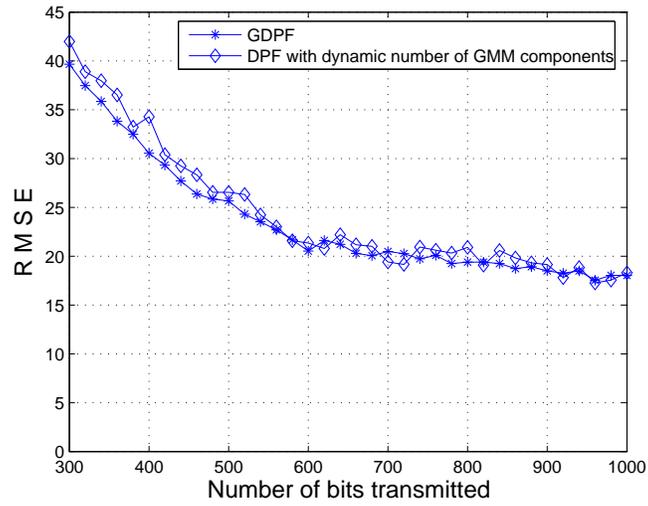


Figure 5.4: Plot of number of bits transmitted vs RMSE

and more bandwidth efficient. There is no estimation accuracy degradation when we dynamically select the number of GMM components.

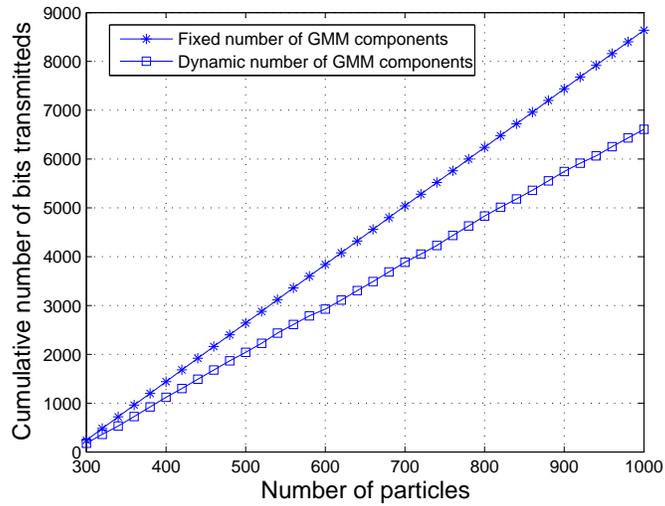


Figure 5.5: Plot of cumulative number of bits transmitted vs number of particles

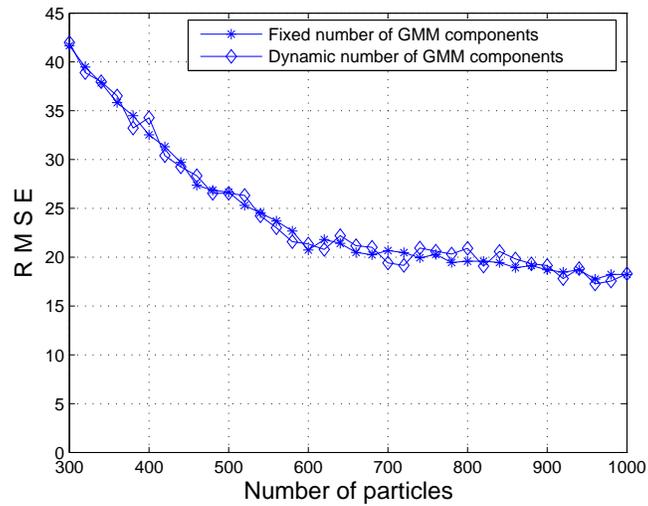


Figure 5.6: Fixed number of GMM components vs dynamic number of GMM components

Chapter 6

Target Tracking in Heterogeneous Sensor Networks

6.1 Motivation

Heterogeneous sensor networks with multiple sensing modalities are gaining popularity in diverse fields because they can provide diverse sensor data for multiple applications with different purposes [60]. Multiple sensing modalities provide flexibility and robustness, however, different sensors may have different resource requirements in terms of processing, memory, or bandwidth. And heterogeneous sensor networks can have nodes with various capabilities for supporting several sensing tasks.

Combining the information from multiple heterogeneous sensors can lead to more accurate tracking results than using a single sensor. To fuse these heterogeneous and non-linear measurements, there are many tracking algorithms, of which the most commonly used is the classical method called the extended Kalman filter (EKF) [13], where the non-linear measurement model and/or nonlinear motion model are linearized via Taylor series expansion, and the noises are approximately to be Gaussian. On the other hand, a Monte-Carlo simulation based recursive estimation algorithm, the particle filtering (PF) algorithm [5] [15] has emerged as a very promising technique to solve the non-linear and non-Gaussian filtering problem. It has been shown that using highly nonlinear measurements, such as bearing-only measurements, the PF outperforms the EKF [29].

In this chapter, we compare the tracking performance of the EKF and the PF under various situations, where different combinations of sensor measurements are available for data fusion. Different types of sensors are considered, including range-only sensors, bearing-only sensors, and radars that provide both range and bearing measurements. Besides the non-linearity in the measurements, non-Gaussian measurement noise, the glint noise modeled as a Gaussian mixture, has been used in the experiments. In addition to the relatively easy case where the target moves at nearly a constant velocity, we investigate the difficult case where the target maneuvers and an Interacting Multiple Model (IMM) algorithm has to be used. Through simulation experiments, we demonstrate that particle filter has superior performance during the first several steps after initialization. In steady state, when the data are highly nonlinear bearing-only measurements, the PF still outperforms the EKF. However, whenever radar data are available, the PF has very similar steady-state performance as the EKF in terms of MSE.

6.2 Sensor Network Setup

Since multiple heterogeneous sensors are connected to form a sensor network, it is very important to take advantage of the information from multiple sources. Here we adopt one of the most common data fusion schemes, namely the centralized fusion scheme. In a centralized fusion process, all the sensors transmit their raw measurements, such as range and bearing to the fusion center, as described in Fig. 6.1. After collecting all these measurements, the fusion center fuses them to form a new and more accurate estimate of the target state. The fusion is accomplished by the tracker in a very natural way. Namely all the raw measurements and their associated accuracies are used to update the target state. The tracker only needs to adjust its measurement equation to reflect that measurements are from multiple heterogeneous sensors. The centralized fusion scheme is optimal in the sense that no information is lost during the fusion process, since the unprocessed raw measurements are transmitted to the fusion center.

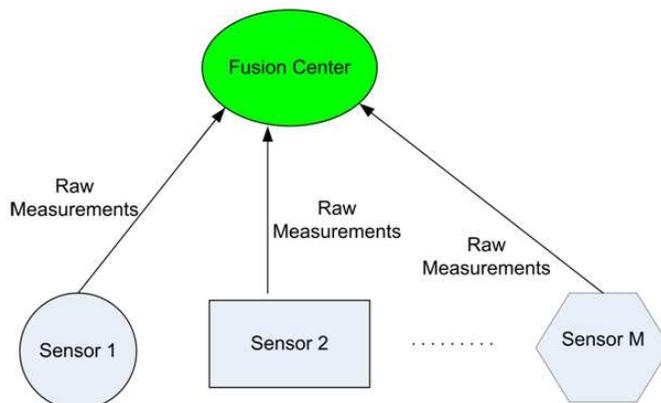


Figure 6.1: Centralized fusion process

6.3 System Models

6.3.1 Target Motion Model

The maneuvering target motion is modeled by three switching dynamics models whose evolution follows a Markov chain, also called a Jump Markov System (JMS) [58][61]. We assume that at any time, the target moves according to one of $s = 3$ dynamic behavior models: (a) Constant Velocity (CV) motion model, (b) clockwise Coordinated Turn (CT) model, and (c) anticlockwise CT model. Let $S = \{1, 2, 3\}$ denotes the set of three models for the dynamic motion. Then, the target dynamics can be written as

$$\mathbf{x}_k = f^{r_k}(\mathbf{x}_{k-1}) + \mathbf{v}_k \quad (6.1)$$

where \mathbf{x}_k is the state vector defined by $\mathbf{x}_k = [x \ \dot{x} \ y \ \dot{y}]$, k denotes the discrete time index, and $r_k \in S$ is the regime variable taking effect in the time interval $(k-1, k]$, with transition probabilities $\pi_{ij} \triangleq Pr\{r_{k+1} = j | r_k = i\}$, $(i, j \in S)$, such that $\pi_{ij} \geq 0$, $\sum_j \pi_{ij} = 1$. \mathbf{v}_k denotes the white Gaussian noise with covariance

matrix Q ,

$$Q = q \begin{bmatrix} 1/3T^3 & 1/2T^2 & 0 & 0 \\ 1/2T^2 & T & 0 & 0 \\ 0 & 0 & 1/3T^3 & 1/2T^2 \\ 0 & 0 & 1/2T^2 & T \end{bmatrix} \quad (6.2)$$

where q is a scalar, and T is the sampling time. For the CV motion model, the $f^{r_k}(\cdot)$ function can be replaced by the transition matrix $F^{r_k}(\cdot)$. When $r_k = 1$, $F^{r_k}(\cdot)$ corresponds to the standard CV model

$$F^{r_k}(\mathbf{x}_k) = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & T \end{bmatrix} \quad (6.3)$$

And $r_k = 2, 3$ correspond to clockwise and anticlockwise CT motions, respectively,

$$F^{(j)}(\mathbf{x}_k) = \begin{bmatrix} 1 & \frac{\sin(\omega_k^{(j)})T}{\omega_k^{(j)}} & 0 & -\frac{1-\cos(\omega_k^{(j)})T}{\omega_k^{(j)}} \\ 0 & \cos(\omega_k^{(j)}) & 0 & \sin(\omega_k^{(j)}) \\ 0 & \frac{1-\cos(\omega_k^{(j)})T}{\omega_k^{(j)}} & 1 & \frac{\sin(\omega_k^{(j)})T}{\omega_k^{(j)}} \\ 0 & \sin(\omega_k^{(j)}) & 0 & \cos(\omega_k^{(j)}) \end{bmatrix}, \quad j = 2, 3 \quad (6.4)$$

Here the mode-conditioned turning rates are given by

$$\begin{aligned} \omega_k^{(2)} &= \frac{a_m}{\sqrt{\dot{x}^2 + \dot{y}^2}} \\ \omega_k^{(3)} &= -\frac{a_m}{\sqrt{\dot{x}^2 + \dot{y}^2}} \end{aligned}$$

where a_m is the constant maneuver acceleration parameter.

6.3.2 Sensor Measurement Model

Three types of sensors are used in our work. These are 1) ESM sensor that reports bearing-only measurements, 2) range sensor that reports range measurements and 3) 2D RADAR sensor that reports range-bearing measurements [59]. The measurement model can be mathematically written as

$$\mathbf{z}_k^j = h^{(j)}(\mathbf{x}_k) + \mathbf{w}_k \quad (6.5)$$

where \mathbf{z}_k^j is the measurement from sensor j . $h^{(i)}(\cdot)$ corresponds to three types of sensor measurement models, $i = 1, 2, 3$.

$$h^{(1)}(\mathbf{x}_k) = \tan^{-1} \left(\frac{y_k - y^{s_j}}{x_k - x^{s_j}} \right) \quad (6.6)$$

$$h^{(2)}(\mathbf{x}_k) = \sqrt{(y_k - y^{s_j})^2 + (x_k - x^{s_j})^2} \quad (6.7)$$

$$h^{(3)}(\mathbf{x}_k) = \left[\frac{\tan^{-1} \left(\frac{y_k - y^{s_j}}{x_k - x^{s_j}} \right)}{\sqrt{(y_k - y^{s_j})^2 + (x_k - x^{s_j})^2}} \right] \quad (6.8)$$

where (x_k, y_k) is the target position at time k , (x^{s_j}, y^{s_j}) is the position of sensor j . And \mathbf{w}_k denotes the measurement noise. In our work, we examine the standard Gaussian noise as well as the glint noise [62].

Changes in the aspect toward the radar can cause irregular electromagnetic wave reflections, resulting in significant variation of radar reflections. This phenomenon gives rise to outliers in angle tracking, and it is referred to as target glint. Glint noise has a non-Gaussian distribution, and a mixture approach is widely used in modeling the non-Gaussian glint noise. In the proposed tracking algorithm, the glint noise is modeled by a Gaussian Mixture Model (GMM) with two components. This model consists of one Gaussian with high probability and small variance and another with small probability of occurrence and very high variance.

$$\mathbf{w}_k \sim (1 - a_g)N(\mathbf{0}, \Sigma_1) + a_gN(\mathbf{0}, \Sigma_2) \quad (6.9)$$

where $a_g < 0.5$ is the glint probability, and $\Sigma_1 < \Sigma_2$. Note that when $a_g = 1$, glint noise degenerates to standard Gaussian noise with zero mean and covariance matrix Σ_1 .

6.4 Target Tracking Algorithms

This section describes the recursive algorithms implemented for tracking a single target using EKF or particle filter techniques. Two of the algorithms are EKF-based and the other two are PF-based schemes. The algorithms considered are

(i) EKF-IMM, (ii) PF-IMM, (iii) EKF-Glint Noise, (iv) PF-Glint Noise. All four algorithms are applicable to both single-sensor and multi-sensor scenarios.

6.4.1 Extended Kalman Filter

Extended Kalman filter is a minimum mean square error (MMSE) estimator based on the Taylor series expansion as shown in Chapter 2. Given the target motion model and measurement model, we have

$$F_{k-1} \triangleq \left. \frac{\partial f(\mathbf{x}_k)}{\partial \mathbf{x}_k} \right|_{(\mathbf{x}_k = \hat{\mathbf{x}}_{k-1|k-1})} \quad (6.10)$$

$$H_k \triangleq \left. \frac{\partial h(\mathbf{x}_k)}{\partial \mathbf{x}_k} \right|_{(\mathbf{x}_k = \hat{\mathbf{x}}_{k|k-1})} = \begin{bmatrix} -\frac{y_k - y^s}{(y_k - y^s)^2 + (x_k - x^s)^2} & 0 & -\frac{x_k - x^s}{(y_k - y^s)^2 + (x_k - x^s)^2} & 0 \\ \frac{x_k - x^s}{\sqrt{(y_k - y^s)^2 + (x_k - x^s)^2}} & 0 & \frac{y_k - y^s}{\sqrt{(y_k - y^s)^2 + (x_k - x^s)^2}} & 0 \end{bmatrix} \quad (6.11)$$

For centralized measurement EKF fusion

$$H_k = [H_{k,S_1}^T, H_{k,S_2}^T, \dots, H_{k,S_n}^T]^T \quad (6.12)$$

$$R_k = \begin{bmatrix} R_{k,S_1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & R_{k,S_n} \end{bmatrix} \quad (6.13)$$

where H_{k,S_i} is the Jacobian of the measurement model for each sensor, and R_{k,S_i} is the covariance of the measurements model.

For the maneuvering target tracking problem, the IMM algorithm has been shown to be one of the most cost effective and simple approaches. At each calculation cycle, the IMM consists of three major steps: interaction (mixing), filtering and combination. At each time, the initial condition for the filter matched to a certain mode is obtained by mixing the state estimates of all the filters at the previous time under the assumption that this particular mode is in effect at the current time. This is followed by a regular filtering step, performed in parallel for each mode. Then a combination of the updated state estimates of all the filters yields the state estimate.

6.4.2 Particle Filtering

Particle filters represent the state probability density function approximately through a set of samples and implement Bayesian recursion directly on the samples instead of dealing with the exact analytical functional representations of the distributions [29][15]. Tracking framework based on particle filtering will show better performance on nonlinear/non-Gaussian problems than EKF. The recursive Bayesian filtering paradigm provides the a *posteriori* PDF $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ via the prediction and update recursions. **Prediction:**

$$p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})d(\mathbf{x}_{k-1}) \quad (6.14)$$

Updating:

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) = \frac{p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{z}_{1:k-1})}{p(\mathbf{z}_k|\mathbf{z}_{1:k-1})} \quad (6.15)$$

where the state \mathbf{x}_k evolution is described in terms of the transition probability:

$$\begin{aligned} p(\mathbf{x}_k|\mathbf{x}_{k-1}) &= \int p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{v}_{k-1})p(\mathbf{v}_{k-1})d\mathbf{v}_{k-1} \\ &= \int \delta(\mathbf{x}_k - f_{k-1}(\mathbf{x}_{k-1}, \mathbf{v}_{k-1}))p(\mathbf{v}_{k-1})d\mathbf{v}_{k-1} \end{aligned} \quad (6.16)$$

And how the given \mathbf{x}_k fits the available measurement \mathbf{z}_k is described as:

$$p(\mathbf{z}_k|\mathbf{x}_k) = \int \delta(\mathbf{z}_k - \mathbf{h}_k(\mathbf{x}_k, \mathbf{w}_k))p(\mathbf{w}_k)d\mathbf{w}_k \quad (6.17)$$

For maneuvering target tracking, the aim of the optimal filter is to sequentially estimate the unknown hybrid hidden state $\{\mathbf{x}_k, r_k\}$ given the observations $\{\mathbf{z}_{1:k}\}$. Applying Bayes rule, the formulation of the recursion that updates $p(\mathbf{x}_{0:k}, r_{1:k-1}|\mathbf{z}_{1:k-1})$ to $p(\mathbf{x}_{0:k}, r_{1:k}|\mathbf{z}_{1:k})$ can be derived as

$$p(\mathbf{x}_{0:k}, r_{1:k}|\mathbf{z}_{1:k}) = p(\mathbf{x}_{0:k-1}, r_{1:k-1}|\mathbf{z}_{1:k-1}) \frac{p(\mathbf{z}_k|\mathbf{x}_{0:k}, r_{1:k}, \mathbf{z}_{1:k-1})f(\mathbf{x}_k|\mathbf{x}_{k-1}, r_{k-1})\pi_{r_{k-1}r_k}}{C} \quad (6.18)$$

where C is a constant. The basic idea for solving maneuvering target tracking using a particle filter is to decouple the hybrid estimation problem into a discrete part and a continuous part. We assume that sensor measurements are independent from each other. Here is the summary of the particle filter solution for the maneuvering target tracking problem in sensor networks.

1. Generate samples of $r_k^{(i)}$ from an importance proposal distribution $\tilde{r}_k^{(i)} \sim \pi(r_k | z_{1:k}, r_{1:k-1}^{(i)})$, where $z_{1:k}$ represents the measurements from all the sensors up to time k . Generate samples $\tilde{\mathbf{x}}_k^{(i)} \sim p(\mathbf{x}_k | \mathbf{x}_{k-1}, r_k^{(i)})$.

2. Evaluate the importance weights

$$\tilde{\omega}_k^{(i)} \propto \omega_{k-1}^{(i)} \frac{p(\tilde{r}_k^{(i)} | \tilde{r}_{k-1}^{(i)})}{\pi(\tilde{r}_k^{(i)} | \mathbf{z}_{1:k}, \tilde{r}_{1:k-1}^{(i)})} \prod_{j=1}^{S_n} p(\mathbf{z}_k^j | \mathbf{z}_{1:k-1}, r_{1:k}^{(i)}) \quad (6.19)$$

3. Normalize the weights

$$\tilde{\omega}_k^{(i)} = \frac{\tilde{\omega}_k^{(i)}}{\sum_j^N \tilde{\omega}_k^{(i)}} \quad (6.20)$$

4. Resampling: multiply/discard particles $\{r_k^{(i)}, i = 1, 2, \dots, N\}$ with respect to high/low normalized importance weights $\tilde{\omega}_k^{(i)}$ to obtain N samples $\{r_k^{(i)}, \mathbf{x}_k^{(i)}\}_{i=1}^N$

5. Calculate MMSE of $\hat{\mathbf{x}}_{k|k} = \sum_{i=1}^N \omega_k^{(i)} \tilde{\mathbf{x}}_k^{(i)}$

6.5 Simulation Results

We address the problem of tracking a maneuvering target in noise using multiple heterogeneous sensors. Fig 6.2 shows the tracking scenario. In our experiments, we use the dynamic state space model to generate the synthetic data, and 50 Monte Carlo computation simulations were carried out to evaluate the performance of the algorithms for each experiment. The position mean square error is defined as

$$MSE_k = \frac{1}{N} \sum_{n=1}^N [(x_k - \hat{x}_{k|k})^2 + (y_k - \hat{y}_{k|k})^2] \quad (6.21)$$

We set sampling rate $T = 1$, and $q = 20$, then

$$Q = \begin{bmatrix} 6.6667 & 10 & 0 & 0 \\ 10 & 20 & 0 & 0 \\ 0 & 0 & 6.6667 & 10 \\ 0 & 0 & 10 & 20 \end{bmatrix} \quad (6.22)$$

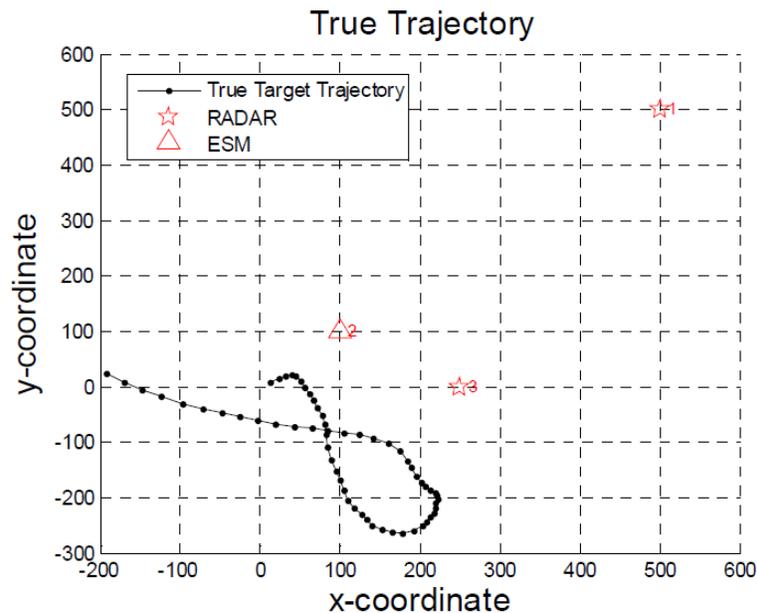


Figure 6.2: Simulated trajectory in a multi-sensor environment

The typical maneuver acceleration parameter is set to $a_m = 1 \text{ m/s}^2$. And the mode probability transition matrix used in the IMM is

$$\mathbf{\Pi} = \begin{bmatrix} 0.9 & 0.05 & 0.05 \\ 0.6 & 0.3 & 0.1 \\ 0.6 & 0.1 & 0.3 \end{bmatrix} \quad (6.23)$$

The glint probability is set to $a_g = 0.9$, the measurement noise covariances $\Sigma_2 = 10\Sigma_1$, and

$$\Sigma_1 = \begin{bmatrix} 3.0462 \times 10^{-4} & 0 \\ 0 & 5 \end{bmatrix}$$

6.5.1 Target Tracking Using Two Bearing-only Sensors

Bearing-only sensors are located $(x^{S_1}, y^{S_1}) = (100\text{m}, 100\text{m})$ and $(x^{S_2}, y^{S_2}) = (0\text{m}, 250\text{m})$ respectively. Figures 6.3 and 6.4 show the tracking results for the CV model with glint noise. Figures 6.5 and 6.6 show the tracking results for a maneuvering target. The PF shows a better tracking performance than EKF. As

we can see later, using only two bearing-only sensors is the most difficult case to track the target, and the tracking results are much worse than those of using two range sensors. For the difficult case where the target is maneuvering, we can see that the MSEs for both the EKF and the PF are higher than those in the case where the target motion follows a CV model.

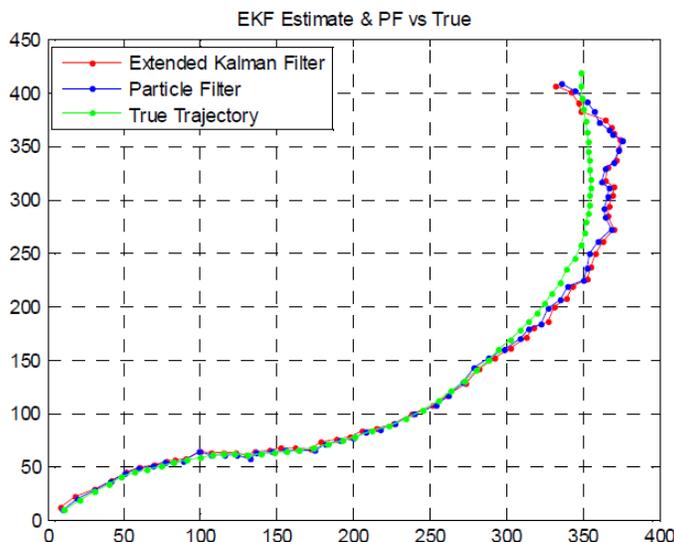


Figure 6.3: 2 bearing sensors - CV model with glint measurement noise : tracking results

6.5.2 Target Tracking Using One Radar Sensor

We assume that a radar is located at $(x^S, y^S) = (500\text{m}, 500\text{m})$. From the experimental results Figures 6.7 and 6.8, we can see that except for the first few steps, EKF and PF achieve almost the same MSE. But the computation time is much shorter for the EKF. For the difficult case where the target is maneuvering, we can see in Figures 6.9 and 6.10 that the MSEs for both the EKF and the PF are higher than those in the case where the target motion follows a CV model.

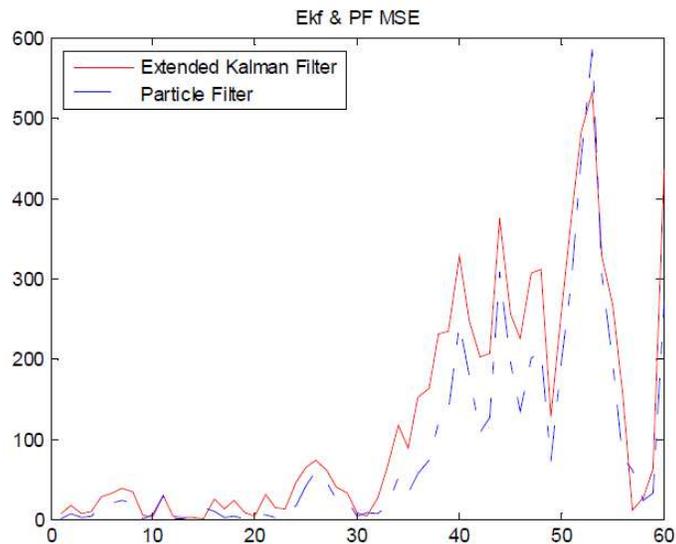


Figure 6.4: 2 bearing sensors - CV model with glint measurement noise: MSE

6.5.3 Target Tracking Using One Range Sensor and One Bearing-only Sensor

The range sensor position is set to $(x^{S_1}, y^{S_1}) = (0\text{m}, 250\text{m})$, and bearing-only sensor position is set to $(x^{S_2}, y^{S_2}) = (100\text{m}, 100\text{m})$. The function of bearing-only sensor plus range sensor is almost the same as a single radar sensor, except that the bearing-only sensor and range sensor are located at different locations, so we have similar experimental results as in Section 6.5.2 shown in Figures 6.11, 6.12, 6.13 and 6.14.

6.5.4 Target Tracking Using Two Range Sensors

Two range sensors positions are set to $(x^{S_1}, y^{S_1}) = (100\text{m}, 250\text{m})$ and $(x^{S_2}, y^{S_2}) = (300\text{m}, 0\text{m})$, respectively. For this sensor configuration, the PF still has similar steady-state performance as that of the EKF. Results are shown in Figures 6.15, 6.16, 6.17 and 6.18.

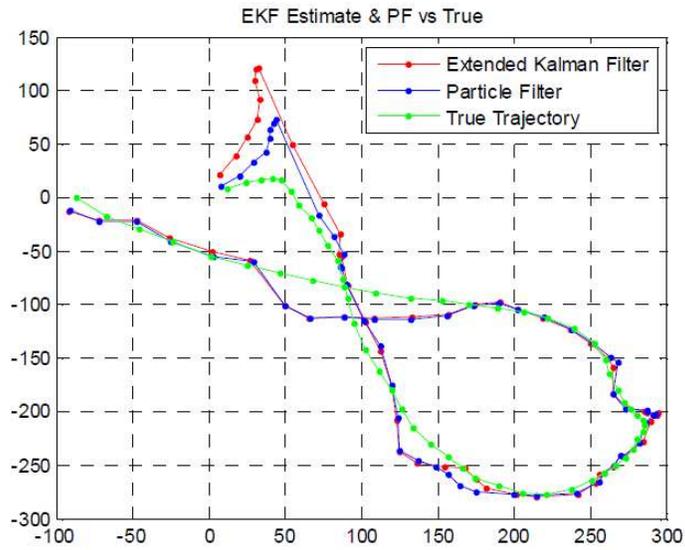


Figure 6.5: 2 bearing sensors - Maneuvering target: tracking results

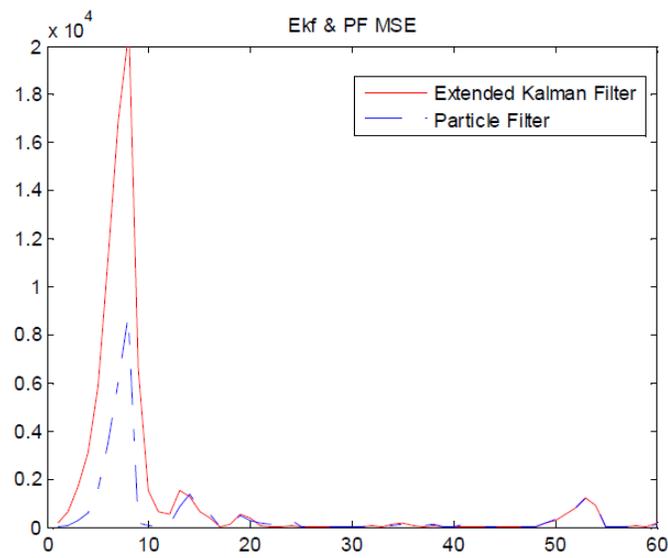


Figure 6.6: 2 bearing sensors - Maneuvering target: MSE

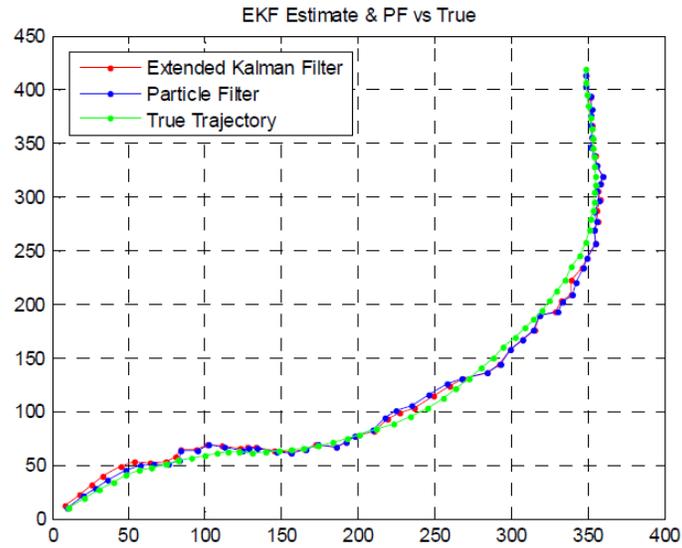


Figure 6.7: One radar sensor - CV model with glint measurement noise: tracking results

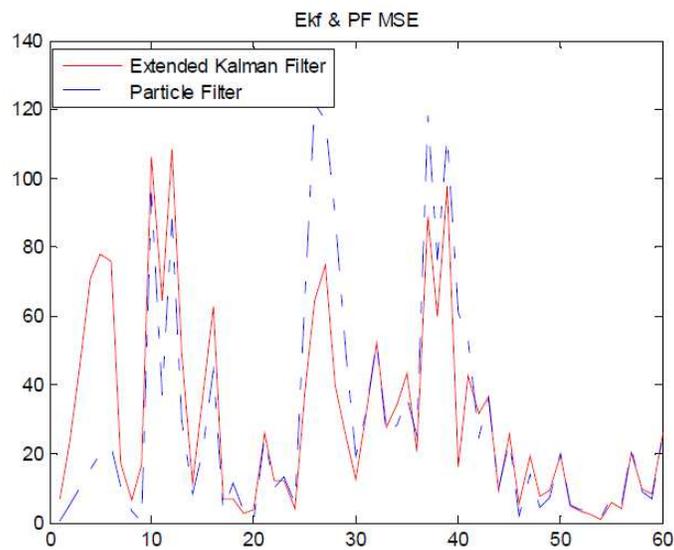


Figure 6.8: One radar sensor - CV model with glint measurement noise: MSE

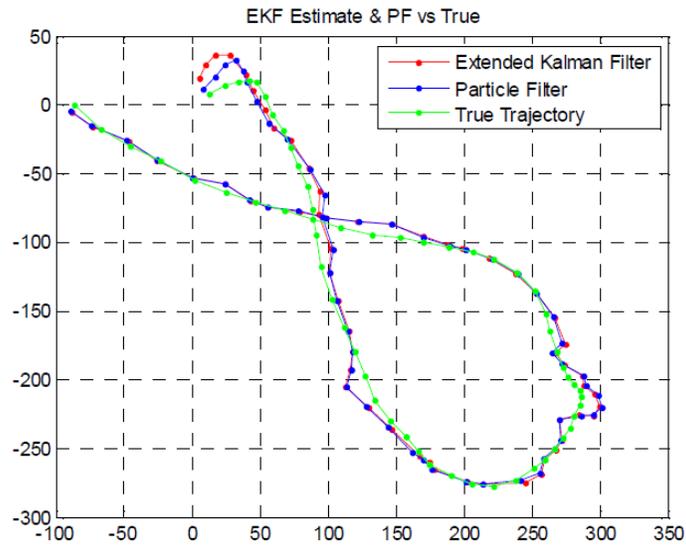


Figure 6.9: One radar sensor - Maneuvering target: tracking results

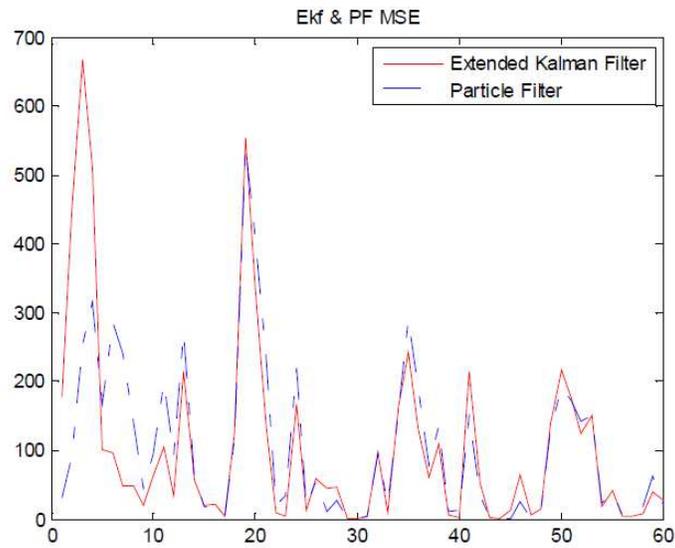


Figure 6.10: One radar sensor - Maneuvering target: MSE

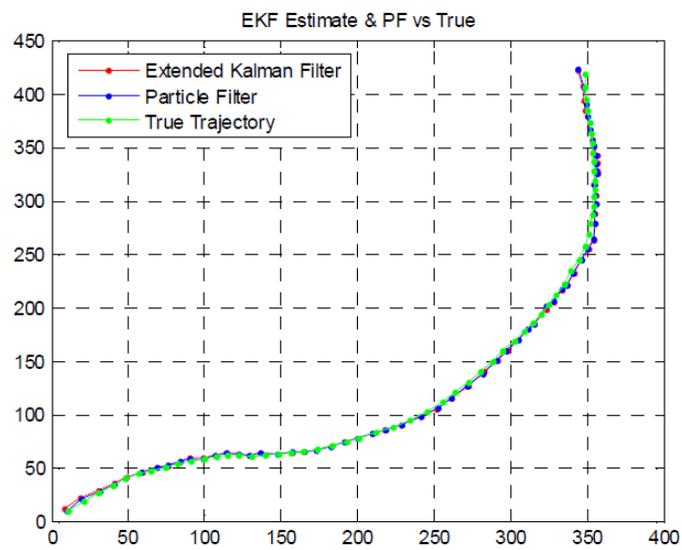


Figure 6.11: One range and One bearing - CV model with glint measurement noise: tracking results

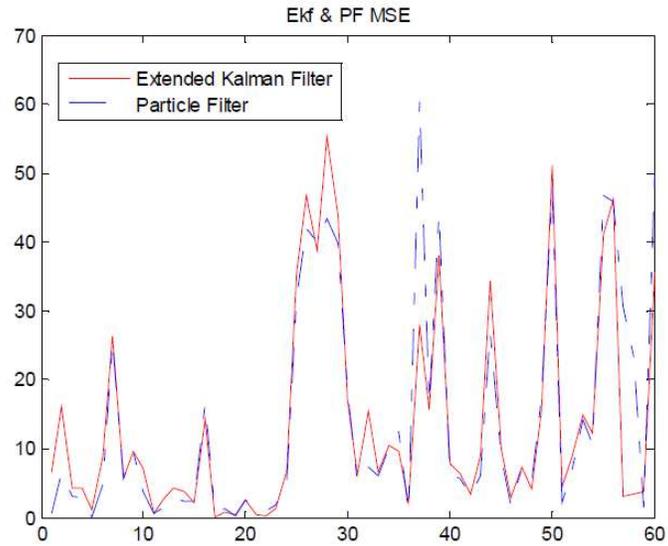


Figure 6.12: One range and One bearing - CV model with glint measurement noise: MSE

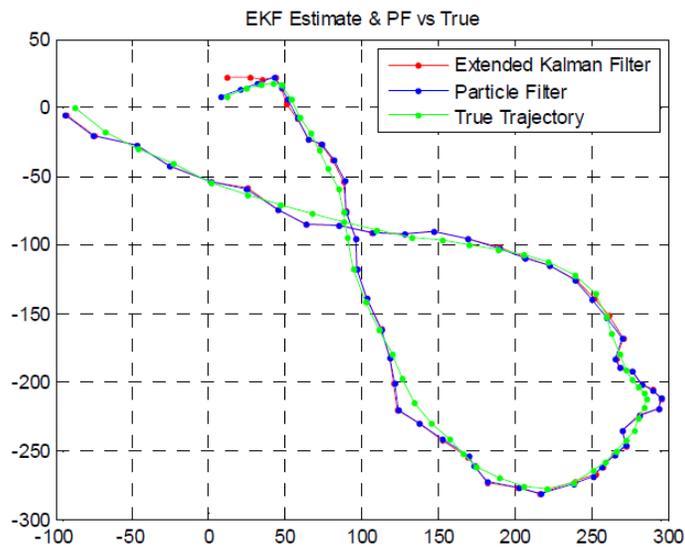


Figure 6.13: One range and One bearing - Maneuvering target: tracking results

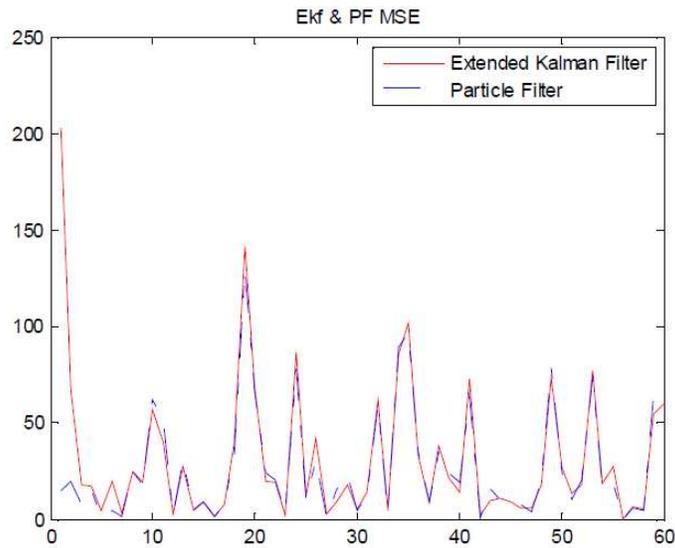


Figure 6.14: One range and One bearing - Maneuvering target: MSE

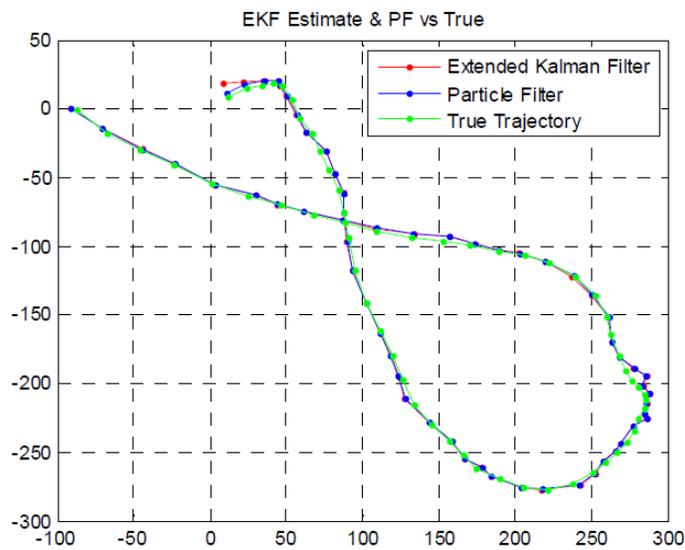


Figure 6.15: Two range sensors - CV model with glint measurement noise: tracking results

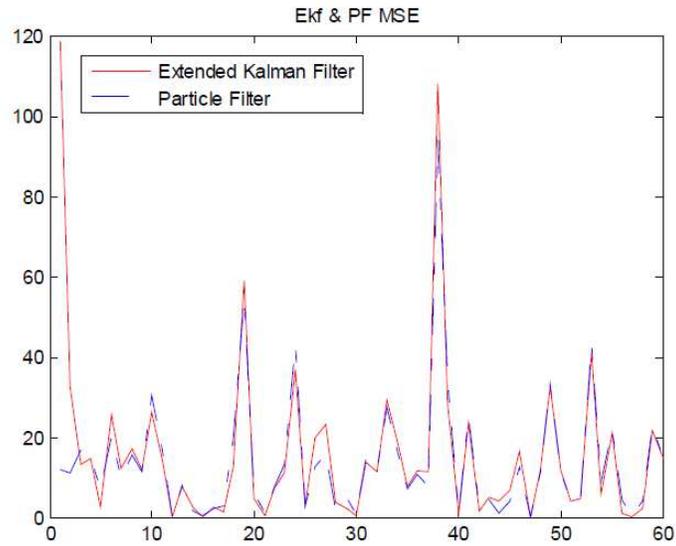


Figure 6.16: Two range sensors - CV model with glint measurement noise: MSE

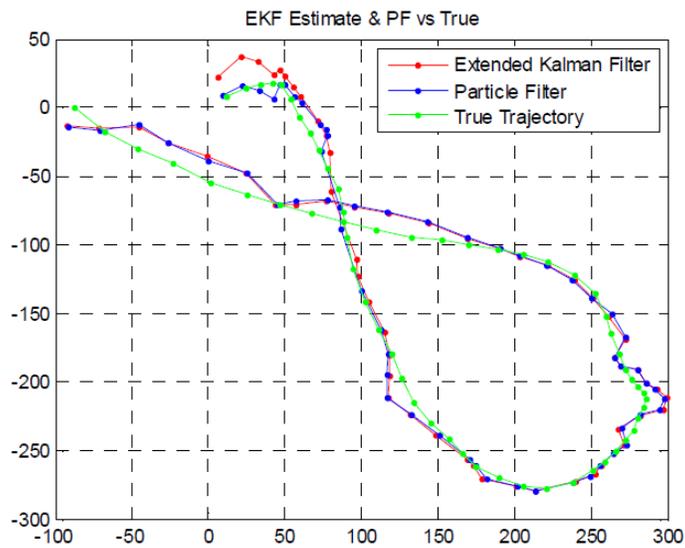


Figure 6.17: Two range sensors - Maneuvering target: tracking results

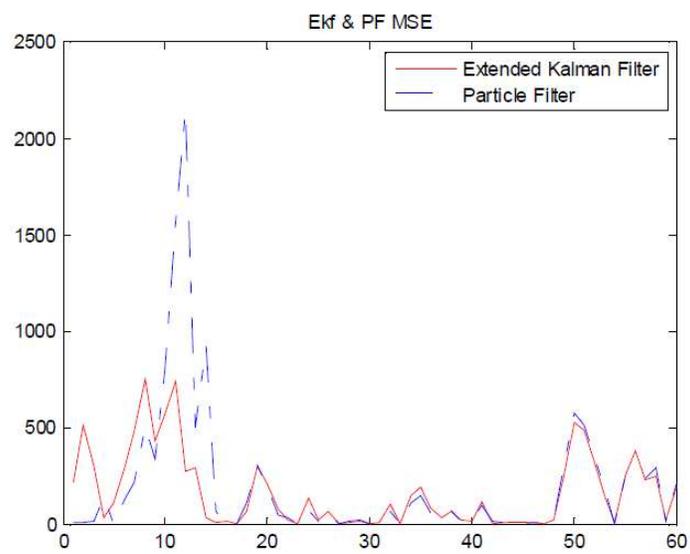


Figure 6.18: Two range sensors - Maneuvering target: MSE

6.5.5 Target Tracking Using Bearing Only, Range and Radar Sensors

In this experiment, we use three different type of sensors, and set the bearing-only sensor at location $(x^{S_1}, y^{S_1}) = (250\text{m}, 0\text{m})$, range sensor at location $(x^{S_2}, y^{S_2}) = (0\text{m}, 250\text{m})$ and radar sensor at location $(x^{S_3}, y^{S_3}) = (500\text{m}, 500\text{m})$. With more sensors providing information, the MSE is much smaller than the previous cases. As expected, more accurate tracking results are achieved, since we are fusing data from more sources. Again, the PF and the EKF have very close steady-state performance. Simulation results are shown in Figures 6.19, 6.20, 6.21 and 6.22.

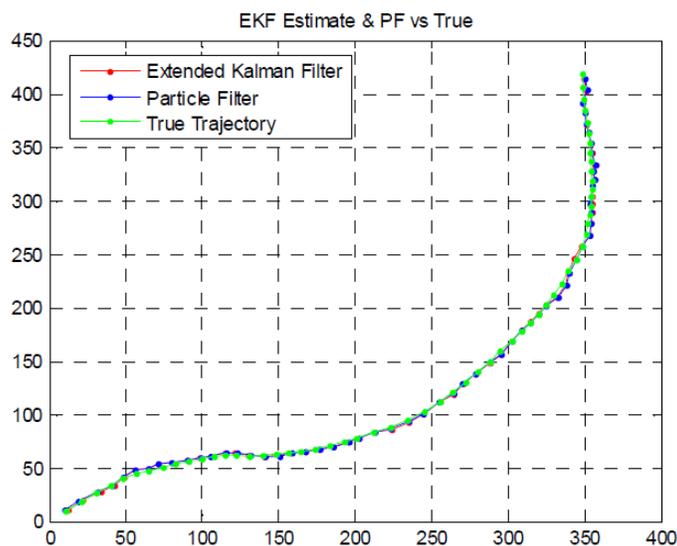


Figure 6.19: Bearing + range + radar - CV model with glint measurement noise: tracking results

From the above experiments, we can observe that even when the measurement model is nonlinear for all types of sensors, using particle filtering does not always achieve a better steady-state performance in terms of MSE. Only in the cases where bearing-only sensors are used, the particle filter shows better performance than the EKF. We also found that even when the initial condition for both EKF

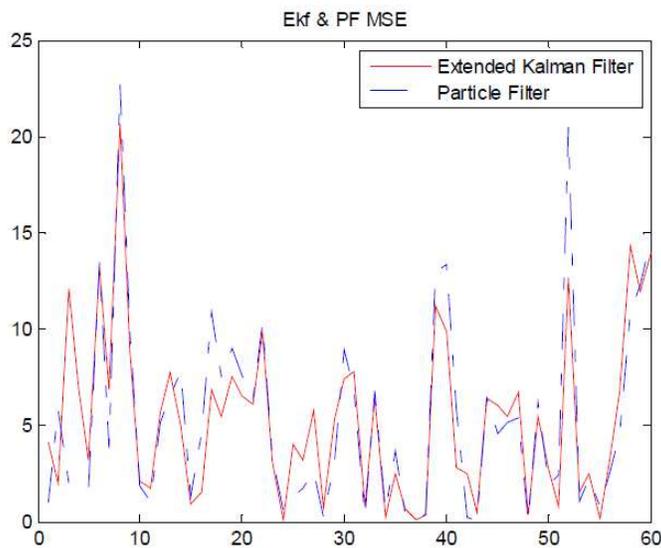


Figure 6.20: Bearing + range + radar - CV model with glint measurement noise: MSE

and PF is the same, in the first few tracking steps, PF tracking results are more accurate than EKF. This is a valuable characteristic, especially when clutter and false alarms are among the measurements. When the measurements contain many false alarms, there is uncertainty as to which measurement is from the target and which is a false alarm. With such uncertainty, inaccurate estimates even at one time step could lead the filter to diverge and result in the loss of the target track. The PF has the potential to maintain the target track for a longer time in such harsh and realistic conditions. This issue needs further investigation in the future.

6.6 Summary

In this chapter, we investigated several data fusion and target tracking algorithms for a surveillance system that consists of multiple heterogeneous sensors. Algorithms based on the classical extended Kalman filter (EKF) and on the emerging non-Gaussian and nonlinear particle filtering (PF) techniques were implemented.

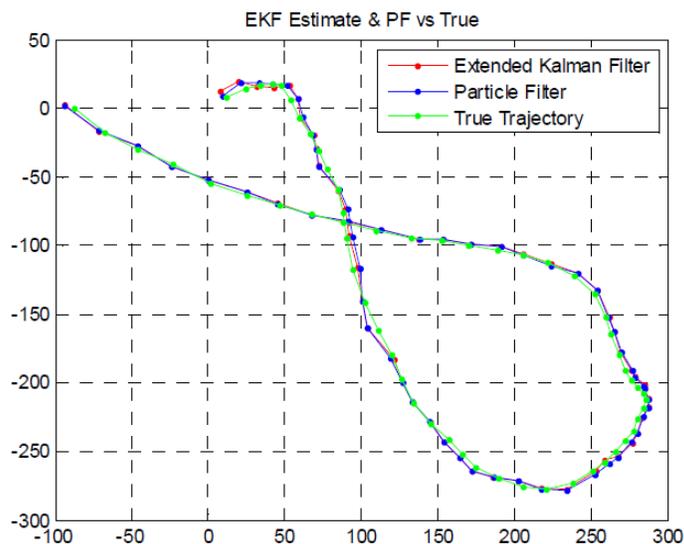


Figure 6.21: Bearing + range + radar - Maneuvering target: tracking results

These algorithms were tested in the practical case where a target maneuvers from time to time and an Interacting Multiple Model (IMM) framework was used. We also tested them in the presence of radar glint noise.

The performances of the EKF and the particle filter were compared through extensive simulation experiments. The results show that for highly non-linear measurements, such as those from multiple bearing-only sensors, particle filter exhibits a superior data fusion and tracking performance than the EKF. However, if the system receives measurements from a radar (both bearing and range measurements), the EKF and the PF have very similar tracking accuracy, and the EKF is a more desirable choice, considering that it requires much less computation than the PF, and has a much easier real-time implementation.

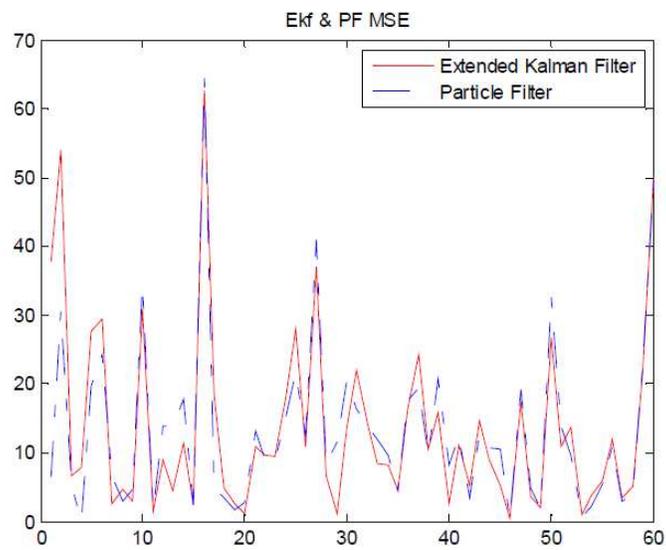


Figure 6.22: Bearing + range + radar - Maneuvering target: MSE

Chapter 7

Conclusions

The Bayesian paradigm for sequential estimation has proven to be successful for a large number of applications. Analyzing the performance of estimation results, therefore, becomes crucial to determine whether the imposed requirements are realistic or not. One of the main contributions of this dissertation is to introduce a novel conditional PCRLB for sequential Bayesian estimation problems for nonlinear/non-Gaussian dynamic systems.

In Chapter 3, we presented the exact conditional PCRLB as well as its recursive evaluation approach including an approximation. For the nonlinear/non-Gaussian systems, it is not realistic to have the analytical closed-form for the conditional PCRLB. Therefore, we proposed a general sequential Monte Carlo approximation for this bound to provide a convenient numerical evaluation solution. The sequential estimate and its performance evaluation through conditional PCRLB are calculated alternately. The current estimate results are used for computing the conditional PCRLB at the next step.

One of the most important properties of conditional PCRLB is that it is an online bound compared to the conventional PCRLB bound for the reason that it utilizes the available measurement information. As a result, it is more appropriate for evaluating the sequential estimate results dynamically. We also investigated and discussed the existing measurement dependent PCRLBs, and provided illustrative examples to compare the differences between the conditional PCRLB and the existing measurement dependent PCRLBs.

Conditional PCRLB is expected to handle broader range of problems and

result in more accurate estimation than existing approaches. One possible application area is sensor management in sensor networks. Choosing the most informative set of sensors is likely to improve the tracking performance, while at the same time reduce the requirement for communication bandwidth and the energy needed by sensors for sensing, local computation and communication. In Chapter 4, we mainly focused on applying the conditional PCRLB to the sensor selection problems for target tracking in sensor networks, and comparing its performance with existing sensor selection approaches, including information driven and those based on other existing measurement dependent PCRLBs. Simulation results for both the analog and quantized measurement data cases demonstrate the improved tracking performance by the conditional PCRLB approach compared to other existing methods in terms of tracking accuracy.

In addition, we presented a novel algorithm to save communication bandwidth for target tracking in sensor networks with distributed particle filters in Chapter 5. The transmission bandwidth from the local sensor to the fusion center is saved through an approximation of the *a posteriori* distribution obtained from the local filtering results. At the fusion center, an optimal rule was presented to fuse the information collected from the distributed sensors to make an estimate of the target state.

In Chapter 6, we presented several algorithms for tracking a maneuvering target with glint noise in heterogeneous sensor networks. Extensive simulations were carried out to compare their performance.

Future work will focus on investigating the properties of the proposed bound. Theorem 1 in Chapter 3 showed that the conditional PCRLB is not only a bound on the filtering estimator $\hat{\mathbf{x}}_{k+1}$, it also sets a bound on the smoothing estimator $\hat{\mathbf{x}}_{0:k}$, when the new measurement \mathbf{z}_{k+1} becomes available. The conditional PCRLB derived in this dissertation provides an approach to recursively predict the MSE one-step ahead. It can be extended to multi-step ahead cases in the future.

Another future challenge is to theoretically investigate the relationship between the conditional PCRLB and the unconditional PCRLBs. In Chapter 3, we have shown through simulations that the conditional PCRLB is tighter than the unconditional one. However, there is no proof yet for the general case. A rigorous mathematical proof is needed in the future.

The work in this dissertation has mainly focused on the accuracy of the single target tracking problems. It is also of considerable importance to be able to evaluate the multi-target tracking performance and therefore the conditional PCRLB for multiple target tracking in sensor works must be derived in the future.

Appendix A

Proof of Proposition 1 in Chapter 3

The following conditions are assumed to exist:

1. If $\frac{\partial p_{k+1}^c}{\partial x_i}$ and $\frac{\partial^2 p_{k+1}^c}{\partial x_i^2}$ exist and both are absolutely integrable with respect to $\mathbf{x}_{0:k+1}$ and \mathbf{z}_{k+1} . Then for any statistic T , where T is a function of $\mathbf{z}_{1:k+1}$, but not that of $\mathbf{x}_{0:k+1}$, such that $E_{p_{k+1}^c}(|T|) < \infty$, the operation of integration and differentiation by x_i can be interchanged in $\int T p_{k+1}^c$. That is

$$\frac{\partial}{\partial x_i} \left[\int T p_{k+1}^c d\mathbf{x}_{0:k+1} d\mathbf{z}_{k+1} \right] = \int T \frac{\partial p_{k+1}^c}{\partial x_i} d\mathbf{x}_{0:k+1} d\mathbf{z}_{k+1} \quad (\text{A.1})$$

2. x_i is defined over the compact interval $[a_i, b_i]$, where $-\infty \leq a_i < b_i \leq \infty$, and for $i = 1, \dots, (k+2)n_x$

$$\lim_{x_i \rightarrow a_i} p(\mathbf{x}_{0:k+1}) = \lim_{x_i \rightarrow b_i} p(\mathbf{x}_{0:k+1}) = 0 \quad (\text{A.2})$$

$$\lim_{x_i \rightarrow a_i} a_i p(\mathbf{x}_{0:k+1}) = \lim_{x_i \rightarrow b_i} b_i p(\mathbf{x}_{0:k+1}) = 0 \quad (\text{A.3})$$

Let \hat{x}_i stand for the estimate of x_i . Since \hat{x}_i is a function of $\mathbf{z}_{1:k+1}$, we have

$$\frac{\partial}{\partial x_i} \int \hat{x}_i p_{k+1}^c d\mathbf{x}_{0:k+1} d\mathbf{z}_{k+1} = 0 \quad (\text{A.4})$$

With Assumption 1, (A.4) implies that

$$\int \widehat{x}_i \frac{\partial p_{k+1}^c}{\partial x_i} d\mathbf{x}_{0:k+1} d\mathbf{z}_{k+1} = 0 \quad (\text{A.5})$$

Applying integration by parts and Assumption 2, it is easy to show that

$$\int_{a_i}^{b_i} x_i \frac{\partial p_{k+1}^c}{\partial x_i} dx_i = - \int_{a_i}^{b_i} p_{k+1}^c dx_i \quad (\text{A.6})$$

By integrating the above quantity with respect to $\mathbf{x}_{0:k+1 \setminus i}$ and \mathbf{z}_{k+1} , where $\mathbf{x}_{0:k+1 \setminus i}$ stands for the state vector up to $k+1$ excluding x_i , we have

$$\int x_i \frac{\partial p_{k+1}^c}{\partial x_i} d\mathbf{x}_{0:k+1} d\mathbf{z}_{k+1} = -1 \quad (\text{A.7})$$

Then subtracting (A.7) from (A.5), it yields

$$\begin{aligned} & \int (\widehat{x}_i - x_i) \frac{\partial p_{k+1}^c}{\partial x_i} d\mathbf{x}_{0:k+1} d\mathbf{z}_{k+1} \\ &= \int (\widehat{x}_i - x_i) \frac{\partial \log p_{k+1}^c}{\partial x_i} p_{k+1}^c d\mathbf{x}_{0:k+1} d\mathbf{z}_{k+1} = 1 \end{aligned} \quad (\text{A.8})$$

Similarly, for $i \neq j$, we have

$$\int (\widehat{x}_i - x_i) \frac{\partial \log p_{k+1}^c}{\partial x_j} p_{k+1}^c d\mathbf{x}_{0:k+1} d\mathbf{z}_{k+1} = 0 \quad (\text{A.9})$$

Combining (A.8) and (A.9) into matrix form, we have

$$\begin{aligned} & \int (\widehat{\mathbf{x}}_{0:k+1} - \mathbf{x}_{0:k+1}) \left[\nabla_{\mathbf{x}_{0:k+1}}^T \log p_{k+1}^c \right] \times \\ & p_{k+1}^c d\mathbf{x}_{0:k+1} d\mathbf{z}_{k+1} = I_{k+2} \end{aligned} \quad (\text{A.10})$$

where I_{k+2} is an identity matrix with dimension $(k+2)n_x$. Now pre-multiply by \mathbf{a}^T and postmultiply by \mathbf{b} , where \mathbf{a} and \mathbf{b} are arbitrary column vectors with dimension $(k+2)n_x$, we have

$$\begin{aligned} & \int \mathbf{a}^T (\widehat{\mathbf{x}}_{0:k+1} - \mathbf{x}_{0:k+1}) \left[\nabla_{\mathbf{x}_{0:k+1}}^T \log p_{k+1}^c \right] \\ & \times p_{k+1}^c \mathbf{b} d\mathbf{x}_{0:k+1} d\mathbf{z}_{k+1} = \mathbf{a}^T \mathbf{b} \end{aligned}$$

Applying the Cauchy-Schwarz inequality, we have

$$(\mathbf{a}^T \mathbf{b})^2 \leq \mathbf{a}^T \text{MSE}(\widehat{\mathbf{x}}_{0:k+1} | \mathbf{z}_{1:k}) \mathbf{a} \times \mathbf{b}^T I(\mathbf{x}_{0:k+1} | \mathbf{z}_{1:k}) \mathbf{b} \quad (\text{A.11})$$

Since \mathbf{b} is arbitrary, letting

$$\mathbf{b} = I^{-1}(\mathbf{x}_{0:k+1}|\mathbf{z}_{1:k})\mathbf{a} \quad (\text{A.12})$$

we can show that

$$\mathbf{a}^T \left(MSE(\hat{\mathbf{x}}_{0:k+1}|\mathbf{z}_{1:k}) - I^{-1}(\mathbf{x}_{0:k+1}|\mathbf{z}_{1:k}) \right) \mathbf{a} \geq 0 \quad (\text{A.13})$$

Since vector \mathbf{a} is arbitrary, $MSE(\hat{\mathbf{x}}_{0:k+1}|\mathbf{z}_{1:k}) - I^{-1}(\mathbf{x}_{0:k+1}|\mathbf{z}_{1:k})$ is positive semidefinite. Q.E.D.

Appendix B

Proof of Proposition 2 in Chapter 3

First, the PDF $p(\mathbf{x}_{0:k+1}, \mathbf{z}_{k+1} | \mathbf{z}_{1:k})$ can be factorized as

$$p(\mathbf{x}_{0:k+1}, \mathbf{z}_{k+1} | \mathbf{z}_{1:k}) = p(\mathbf{x}_{0:k+1} | \mathbf{z}_{1:k+1})p(\mathbf{z}_{k+1} | \mathbf{z}_{1:k}) \quad (\text{B.1})$$

At time k , the prediction or prior $p(\mathbf{x}_{k+1} | \mathbf{z}_{1:k})$ can be approximated as follows. First, a re-sampling procedure is performed, after which each particle has an identical weight, and the posterior PDF is approximated by

$$p(\mathbf{x}_k | \mathbf{z}_{1:k}) \approx \frac{1}{N} \sum_{l=1}^N \delta(\mathbf{x}_k - \mathbf{x}_k^l) \quad (\text{B.2})$$

The prediction $p(\mathbf{x}_{k+1} | \mathbf{z}_{1:k})$ is derived by propagating the particle set $\{\mathbf{x}_k^l, \omega_k^l\}$ from time k to time $k + 1$ according to the system model (3.4)

$$p(\mathbf{x}_{k+1} | \mathbf{z}_{1:k}) \approx \frac{1}{N} \sum_{l=1}^N \delta(\mathbf{x}_{k+1} - \mathbf{x}_{k+1}^l) \quad (\text{B.3})$$

If the transition density of the state $p(\mathbf{x}_{k+1} | \mathbf{x}_k)$ is chosen as the importance density function[29], then the weights at time $k + 1$ are given by

$$\omega_{k+1}^l \propto \omega_k^l p(\mathbf{z}_{k+1} | \mathbf{x}_{k+1}^l) \quad (\text{B.4})$$

Since re-sampling has been taken at time k , we have $\omega_k^l = 1/N, \forall l$. This yields

$$\omega_{k+1}^l \propto p(\mathbf{z}_{k+1} | \mathbf{x}_{k+1}^l) \quad (\text{B.5})$$

More specifically, the normalized weights are

$$\omega_{k+1}^l = \frac{p(\mathbf{z}_{k+1} | \mathbf{x}_{k+1}^l)}{\sum_{l=1}^N p(\mathbf{z}_{k+1} | \mathbf{x}_{k+1}^l)} \quad (\text{B.6})$$

Then the posterior PDF at time $k+1$ can be approximated by

$$p(\mathbf{x}_{0:k+1} | \mathbf{z}_{1:k+1}) \approx \sum_{l=1}^N \omega_{k+1}^l \delta(\mathbf{x}_{0:k+1} - \mathbf{x}_{0:k+1}^l) \quad (\text{B.7})$$

The second PDF in (B.1) involves an integral

$$p(\mathbf{z}_{k+1} | \mathbf{z}_{1:k}) = \int p(\mathbf{z}_{k+1} | \mathbf{x}_{k+1}) p(\mathbf{x}_{k+1} | \mathbf{z}_{1:k}) d\mathbf{x}_{k+1} \quad (\text{B.8})$$

Substitution of (B.3) into (B.8) yields

$$p(\mathbf{z}_{k+1} | \mathbf{z}_{1:k}) \approx \frac{1}{N} \sum_{l=1}^N p(\mathbf{z}_{k+1} | \mathbf{x}_{k+1}^l) \quad (\text{B.9})$$

Further substituting (B.6), (B.7), and (B.9) into (B.1) yields

$$\begin{aligned} & p(\mathbf{x}_{0:k+1}, \mathbf{z}_{k+1} | \mathbf{z}_{1:k}) \\ & \approx \frac{1}{N} \sum_{l=1}^N \delta(\mathbf{x}_{0:k+1} - \mathbf{x}_{0:k+1}^l) p(\mathbf{z}_{k+1} | \mathbf{x}_{k+1}^l) \end{aligned} \quad (\text{B.10})$$

Note that the choice of transition density as the importance function is only used as a tool to derive the particle-filter version of the conditional PCRLB. For the purpose of state estimation, any appropriate importance density function can be chosen for the particle filter.

Appendix C

Approximation of $B_k^{22,b}$ by Particle Filters in Chapter 3 Section 3.4.2

If the measurement noise is additive Gaussian noise, the likelihood function can be written as follows

$$p(\mathbf{z}_k|\mathbf{x}_k) = \frac{1}{(2\pi)^{\frac{n_z}{2}}|R_k|^{\frac{1}{2}}} \times \exp\left\{-\frac{1}{2}[\mathbf{z}_k - h_k(\mathbf{x}_k)]^T R_k^{-1}[\mathbf{z}_k - h_k(\mathbf{x}_k)]\right\} \quad (\text{C.1})$$

where n_z is the dimension of the measurement vector \mathbf{z}_k . Taking the logarithm of the likelihood function, we have

$$-\log p(\mathbf{z}_k|\mathbf{x}_k) = c_0 + \frac{1}{2}(\mathbf{z}_k - h_k(\mathbf{x}_k))^T R_k^{-1}(\mathbf{z}_k - h_k(\mathbf{x}_k))$$

where c_0 denotes a constant independent of \mathbf{x}_k and \mathbf{z}_k . Then the first and second-order partial derivatives of $\log p(\mathbf{z}_k|\mathbf{x}_k)$ can be derived respectively as follows

$$\nabla_{\mathbf{x}_k} \log p(\mathbf{z}_k|\mathbf{x}_k) = [\nabla_{\mathbf{x}_k} h_k(\mathbf{x}_k)] R_k^{-1}(\mathbf{z}_k - h_k(\mathbf{x}_k)) \quad (\text{C.2})$$

$$\begin{aligned}
& -\Delta_{\mathbf{x}_k}^{\mathbf{x}_k} \log p(\mathbf{z}_k | \mathbf{x}_k) \\
& = [\nabla_{\mathbf{x}_k} h_k(\mathbf{x}_k)] R_k^{-1} [\nabla_{\mathbf{x}_k}^T h_k(\mathbf{x}_k)] - \Delta_{\mathbf{x}_k}^{\mathbf{x}_k} h_k(\mathbf{x}_k) \tilde{\Sigma}_{\mathbf{v}_k}^{-1} \Upsilon_k^{22,b}
\end{aligned} \tag{C.3}$$

where

$$\tilde{\Sigma}_{\mathbf{v}_k}^{-1} = \begin{bmatrix} R_k^{-1} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & R_k^{-1} & \mathbf{0} & \vdots \\ \vdots & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{0} & R_k^{-1} \end{bmatrix}_{(n_x n_z) \times (n_x n_z)} \tag{C.4}$$

and

$$\Upsilon_k^{22,b} = \begin{bmatrix} \mathbf{z}_k - h_k(\mathbf{x}_k) & \dots & \mathbf{0} \\ \vdots & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{z}_k - h_k(\mathbf{x}_k) \end{bmatrix}_{(n_x n_z) \times n_x} \tag{C.5}$$

Now with (B.10) and (C.3), we have

$$\begin{aligned}
B_k^{22,b} & = E_{p_{k+1}^c} \{-\Delta_{\mathbf{x}_{k+1}}^{\mathbf{x}_{k+1}} \log p(\mathbf{z}_{k+1} | \mathbf{x}_{k+1})\} \\
& \approx \frac{1}{N} \sum_{l=1}^N \nabla_{\mathbf{x}_{k+1}} h(\mathbf{x}_{k+1}) R_{k+1}^{-1} \nabla_{\mathbf{x}_{k+1}}^T h(\mathbf{x}_{k+1}) \Big|_{\mathbf{x}_{k+1} = \mathbf{x}_{k+1}^l}
\end{aligned} \tag{C.6}$$

where the identity

$$E_{p(\mathbf{z}_{k+1} | \mathbf{x}_{k+1})} \{\Upsilon_{k+1}^{22,b}\} = \mathbf{0} \tag{C.7}$$

has been used.

References

- [1] Archana Bharathidasan and Vijay Anand Sai Ponduru, “Sensor networks: An overview,” *IEEE Infocom*, 2004. [1](#)
- [2] Ian F. Akyildiz, W. Su, Yogesh Sankarasubramaniam, and Erdal Cayirci, “Wireless sensor networks: a survey,” *Computer Networks*, vol. 38, pp. 393–422, 2002. [1](#)
- [3] Kazem Sohraby, Daniel Minoli, and Taieb Znati, *Wireless Sensor Networks: Technology, Protocols, and Applications*, Wiley-Interscience, April 2007. [2](#)
- [4] F. Zhao, J. Shin, and J. Reich, “Information-driven dynamic sensor collaboration,” *IEEE Signal Processing Magazine*, March 2002, vol. 19, pp. 61–72. [3](#), [49](#), [52](#)
- [5] P. Tichavsky, C. H. Muravchik, and A. Nehorai, “Posterior Cramér-Rao Bounds for Discrete-Time Nonlinear Filtering,” *IEEE Trans. on Sig. Proc.*, vol. 46, no. 5, pp. 1386–1396, May 1998. [4](#), [17](#), [18](#), [19](#), [21](#), [38](#), [50](#), [85](#)
- [6] A. Pole, M. West, and P. J. Harrison, *Applied Bayesian Forecasting and Time Series Analysis*, New York: Chapman-Hall, April 1994. [8](#)
- [7] M. West, “Robust sequential approximate bayesian estimation,” *J. Roy. Statist. Soc., Ser. B*, vol. 43, pp. 157166, 1981. [8](#)
- [8] S. C. Kramer and H. W. Sorenson, “Bayesian parameter estimation,” *Automatica*, vol. 24, pp. 789901, 1988. [8](#)

-
- [9] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*, Prentice-Hall, 1979. [8](#), [10](#)
- [10] R.E Kalman, “A new approach to linear filtering and prediction problems,” *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960. [9](#)
- [11] R. E. Kalman and R. S. Bucy, “New results in linear filtering and prediction theory,” *Trans. ASME, Ser. D, J. Basic Eng.*, vol. 83, pp. 95107, 1961. [9](#)
- [12] B. D. O. Anderson and J. B. Moore, “The Kalman-Bucy filter as a true time-varying Wiener filter,” *IEEE Trans. Syst., Man, Cybern.*, pp. 119–128, 1971. [10](#)
- [13] Y. Bar-Shalom and T.E. Fortmann, *Tracking and Data Association*, Academic Press, 1998. [11](#), [71](#), [85](#)
- [14] H. Chen, Q. Shao, and J. Ibrahim, *Monte Carlo Methods in Bayesian Computation*, Springer, New York, NY, 2001. [12](#)
- [15] A. Doucet, N. de Freitas, and N. Gordon Eds., *Sequential Monte Carlo Methods in Practice*, Springer, New York, 2001. [12](#), [14](#), [74](#), [85](#), [91](#)
- [16] M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking,” *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, February 2002. [12](#), [30](#), [34](#)
- [17] A. Marshall, *The use of multi-stage sampling schemes in Monte Carlo computations*, 1956. [14](#)
- [18] A. Kong, J. S. Liu, and W. H. Wong, “Sequential imputations and bayesian missing data problems,” *J. Amer. Statist. Assoc.*, vol. 89, pp. 278–288, 2010. [14](#)
- [19] H.L. Van Trees, *Detection, Estimation and Modulation Theory*, vol. 1, Wiley, New York, 1968. [17](#), [20](#), [23](#)

-
- [20] M.L. Hernandez, T. Kirubarajan, and Y. Bar-Shalom, "Multisensor Resource Deployment Using Posterior Cramér-Rao Bounds," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 40, no. 2, pp. 399–416, April 2004. [18](#), [19](#), [26](#), [50](#)
- [21] K. Punithakumar, T. Kirubarajan, and M.L. Hernandez, "Multisensor Deployment Using PCRLBS, Incorporating Sensor Deployment and Motion Uncertainties," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 42, no. 4, pp. 1474–1485, October 2006. [18](#), [50](#)
- [22] R. Tharmarasa, T. Kirubarajan, and M.L. Hernandez, "Large-Scale Optimal Sensor Array Management for Multitarget Tracking," *IEEE Trans. Syst., Man, Cybern. C: Appl. Rev.*, vol. 37, no. 5, pp. 803–814, September 2007. [18](#), [50](#)
- [23] R. Tharmarasa, T. Kirubarajan, M.L. Hernandez, and A. Sinha, "PCRLB-Based Multisensor Array Management for Multitarget Tracking," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 43, no. 2, pp. 539–555, April 2007. [18](#), [50](#)
- [24] M. Hurtado, T. Zhao, and A. Nehorai, "Adaptive Polarized Waveform Design for Target Tracking Based on Sequential Bayesian Inference," *IEEE Trans. Signal Process.*, vol. 56, no. 3, pp. 1120–1133, March 2008. [19](#), [25](#), [50](#)
- [25] S.P. Sira, A. Papandreou-Suppappola, and D. Morrell, "Dynamic Configuration of Time-Varying Waveforms for Agile Sensing and Tracking in Clutter," *IEEE Trans. Signal Process.*, vol. 55, no. 7, pp. 3207–3217, July 2007. [19](#), [50](#)
- [26] L. Zuo, R. Niu, and P. K. Varshney, "Posterior CRLB Based Sensor Selection for Target Tracking in Sensor Networks," in *Proc. Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, Honolulu, Hawaii, April 2007, vol. 2, pp. 1041–1044. [23](#)
- [27] L. Zuo, R. Niu, and P. K. Varshney, "A Sensor Selection Approach for Target Tracking in Sensor Networks with Quantized Measurements," in

-
- Proc. Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, Las Vegas, Nevada, March 2008, pp. 2521–2524. [23](#)
- [28] R.A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, New York, NY, 1985. [24](#)
- [29] N.J. Gordon, D.J. Salmond, and A.F.M. Smith, “Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation,” *IEE Proceedings F, Radar and Signal Processing*, vol. 140, no. 2, pp. 107–113, April 1993. [30](#), [39](#), [74](#), [85](#), [91](#), [114](#)
- [30] Y. Bar-Shalom, X.R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*, Wiley, New York, 2001. [34](#), [38](#), [39](#), [57](#), [71](#)
- [31] G. Kitagawa, “Non-Gaussian State-Space Modeling of Nonstationary Time Series,” *Journal of the American Statistical Association*, vol. 82, no. 400, pp. 1032–1063, December 1987. [39](#)
- [32] J.H. Kotecha and P.M. Djuric, “Gaussian Particle Filtering,” *IEEE Transactions on Signal Processing*, vol. 51, no. 10, pp. 2592–2601, October 2003. [39](#)
- [33] K.J. Hintz, “A measure of the information gain attributable to cueing,” *IEEE Trans. Systems Man Cybernet.*, vol. 21, no. 2, pp. 434–442, March/April 1991. [49](#)
- [34] Feng Zhao, Jie Liu, Juan Liu, Leonidas Guibas, and James Reich, “Collaborative signal and information processing: An information directed approach,” *Proc. IEEE*, vol. 91, no. 8, pp. 1199–1209, Aug. 2003. [49](#)
- [35] C.M. Kreucher, K.D. Kastella, and A.O. Hero, “Sensor management using an active sensing approach,” *Signal Processing*, vol. 85, no. 3, pp. 607–624, March 2005. [49](#)
- [36] C.M. Kreucher, A.O. Hero, K.D. Kastella, and M.R. Morelande, “An Information-Based Approach to Sensor Management in Large Dynamic Networks,” *Proc. IEEE*, vol. 95, no. 5, pp. 978–999, May 2007. [49](#)

-
- [37] J. L. Williams, J. W. Fisher, and A. S. Willsky, "Approximate Dynamic Programming for Communication-Constrained Sensor Network Management," *IEEE Trans. Signal Processing*, vol. 55, no. 8, pp. 4300–4311, August 2007. [49](#)
- [38] G. M. Hoffmann and C. J. Tomlin, "Mobile Sensor Network Control using Mutual Information Methods and Particle Filters," *IEEE Trans. Autom. Control*, vol. 55, no. 1, pp. 32–47, January 2010. [49](#)
- [39] E. Masazade, R. Niu, P.K. Varshney, and M. Keskinoz, "Energy Aware Iterative Source Localization for Wireless Sensor Networks," *IEEE Transactions on Signal Processing*, vol. 58, no. 9, pp. 4824–4835, Sept. 2010. [49](#)
- [40] E. Masazade, R. Niu, P.K. Varshney, and M. Keskinoz, "Channel Aware Iterative Source Localization for Wireless Sensor Networks," in *Proc. of the 13th International Conference on Information Fusion*, Edinburgh, Scotland, UK, July 2010. [49](#)
- [41] Nikhil Padhye, Long Zuo, Chilukuri K. Mohan, and Pramod K. Varshney, "Dynamic and evolutionary multi-objective optimization for sensor selection in sensor networks for target tracking," *IJCCI*, pp. 160–167, 2009. [65](#)
- [42] C. Y. Chong and S. P. Kumar, "Sensor networks: Evolution, opportunities, and challenges," *Proc. of the IEEE*, vol. 91(8), pp. 1247C1256, 2003. [71](#)
- [43] Carine Hue, Jean pierre Le Cadre, and Patrick Prez, "Sequential monte carlo methods for multiple target tracking and data fusion," *IEEE Trans. on Signal Processing*, vol. 50, no. 2, pp. 309–325, Feb. 2002. [71](#)
- [44] Tom Vercauteren, Dong Guo, and Xiaodong Wang, "Joint multiple target tracking and classification in collaborative sensor networks," *Journal on Selected Areas in Communications*, April 2005. [72](#)
- [45] Tien Pham and Haralabos C. Papadopoulos, "Distributed tracking in ad-hoc sensor network," *IEEE Workshop on Statistical Signal Processing*, July 2005. [72](#)

-
- [46] Mark Coates, “Distributed particle filters for sensor networks,” *Proc. of 3rd Int’l symposium on Information processing in sensor networks*, 2004. 72
- [47] Xiaohong Sheng and Yu-Hen Hu, “Distributed particle filters for wireless sensor network target tracking,” *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, 2005. 72
- [48] XiaoRong Li, Yunmin Zhu, and Chongzhao Han, “Unified optimal linear estimation fusion. i. unified models and fusion rules,” *Information Fusion, Proceedings of the Third International Conference on*, vol. 1, July 2000. 73
- [49] J. Bilmes, “A gentle tutorial of the *EM* algorithm and its application to parameter estimation for gaussian mixture and hidden markov models,” *UC Berkeley TR-97-021*, April 1998. 74, 76
- [50] A.P. Dempster, N.M. Laird, and D.B. Rubin, “Maximum likelihood from incomplete data via the em algorithm,” *Journal of the Royal Statistical Society, Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977. 74
- [51] Z. Ghahramani and M. Jordan, “Learning from incomplete data,” *Technical Report AI Lab, Memo No. 1509, CBCL Paper No. 108, MIT AI Lab*. 74
- [52] M. Jordan and L. Xu, “Convergence results for the em approach to mixtures of experts architectures,” *Neural Networks*, vol. 8, pp. 14091431, 1996, *Seminare Maurey-Schwartz (1975-1976)*. 74
- [53] C. F. Jeff Wu, “On the convergence properties of the em algorithm,” *The Annals of Statistics*, vol. 11, no. 1, pp. 95–103, 1983. 75
- [54] D. Titterton, A. Smith, and U. Makov, *Statistical Analysis of Finite Mixture Distributions*, John Wiley & Sons, Inc., 1985. 76
- [55] J.M. Marin, K. Mengersen, and C.P. Robert, *Bayesian modelling and inference on mixtures of distributions*, *Handbook of Statistics*, Elsevier-Sciences, 2005. 76
- [56] T.M. Cover and J.A. Thomas, *Elements of Information Theory*, Wiley, New York, 1991. 77

- [57] S. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*, Prentice Hall, Englewood Cliffs, NJ 07632, 1993. 78
- [58] X. Rong Li and V.P. Jilkov, “Survey of maneuvering target tracking part i: Dynamic models,” *IEEE Trans. on Aerospace and Electronic Systems*, vol. 39, 2003. 79, 87
- [59] X. Rong Li and V.P. Jilkov, “A survey of maneuvering target tracking-part iii: Measurement models,” *Proc. 2001 SPIE Conf. Signal and Data Processing of Small Targets*, vol. 4473, July-August. 80, 88
- [60] M. Yarvis, N. Kushalnagar, H. Singh, A. Rangarajan, Y. Liu, and S. Singh, “Infocom’05: Proceedings of the 24th annual joint conference of the iee computer and communications societies,” *Acta. Math.*, pp. 878–890, 2005. 85
- [61] Arnaud Doucet, N. Gordon, and Vikram Krishnamurthy, “Particle filters for state estimation of jump markov linear systems,” *IEEE Trans. on Single Processing*, vol. 49, no. 3, 2001. 87
- [62] I. Bilik and J. Tabrikian, “Target tracking in glint noise environment using nonlinear non-gaussian kalman filter,” *Proc. of IEEE International Radar Conference*, April 2006. 89

VITA

NAME OF AUTHOR: Long Zuo

PLACE OF BIRTH: Shijiazhuang, Hebei Province, P.R. China

DATE OF BIRTH: February 11, 1976

GRADUATE AND UNDERGRADUATE SCHOOLS ATTENDED:

Xi'an Jiaotong University, Xi'an, P.R. China

Institute of Automation, Chinese Academy of Science, Beijing, P.R. China

DEGREES AWARDED:

Bachelor of Science in Electrical Engineering, 1999, Xi'an Jiaotong University

Master of Science in Electrical Engineering, 2002, Institute of Automation, Chinese Academy of Science

PROFESSIONAL EXPERIENCE:

Research Assistant, Department of Electrical Engineering and Computer Science, Syracuse, 2002 - 2010