Dissertation

zur Erlangung des akademischen Grades

des Doktors der Naturwissenschaften

(Dr. rer. nat)

Adaptive Particle Filters for Wireless Indoor Target Tracking

eingereicht

am Institut für Informatik

des Fachbereichs Mathematik und Informatik

der Freien Universität Berlin

von

Yubin Zhao

Berlin, 2014

Gutachter:

Prof. Dr. Marcel Kyas
Department of Computer Science
Freie Universität Berlin

Prof. Dr. Vincent Lau
Department of Electronic and Computer Engineering
Hong Kong University of Science and Technology

Tag der Disputation: 12. November 2014

Eidesstattliche Erklärung

Ich versichere, dass ich die Doktorarbeit selbständig verfasst, und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Die Arbeit hat keiner anderen Prüfungsbehörde vorgelegen.

Mir ist bekannt, dass bei Verwendung von Inhalten aus dem Internet ich diese zu kennzeichnen und einen Ausdruck mit Angabe des Datums sowie der Internet-Adresse als Anhang der Doktorarbeit anzugeben habe.

I hereby declare to have written this thesis on my own. I have used no other literature and resources than the ones referenced. All text passages that are literal or logical copies from other publications have been marked accordingly. All figures and pictures have been created by me or their sources are referenced accordingly. This thesis has not been submitted in the same or a similar version to any other examination board.

Berlin, den November 12, 2014
Yubin Zhao

Abstract

Particle filters (PFs) are efficient tools for nonlinear state estimation especially for a wireless indoor target tracking system, who estimates the target's position using wireless facilities in the buildings or houses. Particle filters are recursive Monte-Carlo methods based on Bayes' theorem, which fuse the previous state information and the current measurement data to obtain the target position. The advantage of using PFs is that the heterogeneous information can be combined effectively within the PFs to estimate the unknown state. The purpose is to design PF algorithms with high estimation accuracy, where average absolute error approaches 0, and make PFs robust to hybrid line-of-sight (LOS) and non-line-of-sight (NLOS) errors. Particle filters should also have a low computational complexity. The major contributions are five folds:

1. The impact of instantaneous measurement error is firstly found and analyzed in this thesis. It is the major source of the estimation error of PFs. According to the analysis, a likelihood adaptation method is proposed to reduce the instantaneous measurement error. Then, adaptive PFs integrated with the likelihood adaptation method are developed.
2. Due to the NLOS and multi-path effect, the ranging error is difficult to model in indoor environment. Therefore, a dynamic Gaussian model (DGM) is proposed to describe the distribution of hybrid LOS/NLOS ranging errors. Then, adaptive PFs using DGM are extended in the high dynamic indoor environment still with the accurate estimation.
3. Design the architecture of a real world tracking system for adaptive PFs. In addition, adaptive PF fusing building layout information to improve the estimation accuracy is proposed.
4. Propose distributed PFs protocols to collaboratively estimate the target using local anchors. Selective gossip algorithms are applied in the distributed PF design.
5. Theoretical analysis of the adaptive PFs for the estimation performance based on the Cramér-Rao lower bound. It is proved that with reliable priori information, the adaptive PFs outperform conventional PFs.

The performance of the adaptive PFs are evaluated in various simulations and real world experiment. The estimation results are compared with the conventional PFs and other localization algorithms. For the conventional PFs, the average absolute errors have biases, which are about

0.5m to 1m, and the root mean square error (RMSE) is more than 2m, but the average absolute errors of proposed adaptive PFs can approach to 0 and RMSEs are only about 1.5m error, which meets the requirements of the indoor location based services. In addition, only 30 particles are required in the adaptive PFs, which significantly reduce the computational complexity. In real world experiments, adaptive PFs also outperform other localization algorithms, e.g. linear least square method, nonlinear least square method, min-max algorithm, extended Kalman filter, etc. The estimation accuracy of distributed PFs using selective gossip algorithms are guaranteed. The communication overheads are also significantly reduced compared with other distributed consensus based schemes.

Zusammenfassung

Partikel Filter (PF) sind effiziente Werkzeuge zur nichtlinearen Zustandsschätzung und eignen sich besonders für drahtlose Indoor Tracking Systeme die sich dazu eignen die Position eines Objektes innerhalb von Gebäuden zu schätzen. Partikel Filter entsprechen rekursiven Monte-Carlo Verfahren die auf dem Bayes-Theorem basieren, das die vorher bekannte Position mit neuen Messwerten fusioniert um auf die aktuelle Position zu schließen. Der große Vorteil von PFs besteht darin, das heterogene Informationen effektiv kombiniert werden können um den unbekannten Zustand zu ermitteln. Mein Ziel ist PF Algorithmen mit hoher Genauigkeit zu entwerfen, deren durchschnittlicher Fehler gegen 0 geht und die ein hohes Mass an Robustheit gegenüber Line-of-sight (LOS) als auch Non-line-of-sight (NLOS) Fehlern aufweisen. Die PF Algorithmen sollten außerdem möglichst wenig komplex und einfach berechenbar sein. Die fünf wichtigsten wissenschaftlichen Beiträge dieser Arbeit sind:

1. Der Einfluss des momentanen Messfehlers wird in dieser Arbeit analysiert. Der Messfehler ist die Hauptursache für Schätzungsfehler eines Partikelfilters. Auf Basis der Analyse wird ein Likelihood Verfahren vorgeschlagen, welches den Messfehler reduziert. Nachfolgend werden adaptive PF vorgeschlagen, die das Likelihood Verfahren anwenden.
2. Aufgrund von NLOS und multi-path Effekten ist der Entfernungsfehler einer Messung in einer Indoor-Umgebung schwer zu modellieren. Zur Lösung dieses Problems wird ein dynamisches Gauss-Model (DGM) vorgeschlagen um die Verteilung von hybriden LOS/NLOS Entfernungsfehlern zu beschreiben. Das DGM wird anschließend ebenso in einem adaptiven PF integriert um die Genauigkeit der Zustandsschätzung zu verbessern.
3. Es wird eine Real-World Architektur für ein PF basiertes Tracking-System vorgelegt. Des Weiteren wird vorgeschlagen vorhandene Karteninformationen als zusätzliche Eingabe für den adaptiven PF zu verwenden um die Genauigkeit zu steigern.
4. Es werden verteilte PF Protokolle vorgeschlagen um eine kollaborative Schätzung der Position eines Objektes zu ermöglichen. Selektive Gossip-Algorithm bilden dabei die Grundlage für das verteilte PF Verfahren.
5. Eine theoretische Betrachtung der Schätzqualität der PF anhand der Cramér-Rao-lower-bound. Es wird dabei bewiesen, das adaptive PFs mit zuverlässigen a priori Informationen

bessere Schätzungen liefern als herkömmliche PFs.

In mehreren Simulationen und real world Experimenten wird die Performance von adaptiven PFs evaluiert. Die Resultate des Schätzers werden mit den Resultaten konventioneller PF und anderen Lokalisierungsalgorithmen verglichen. Der durchschnittliche Fehler von konventionellen PFs hat einen Bias von 0,5 m bis 1,0 m und das quadratische Mittel der Fehler beträgt mehr als 2,0 m. Für den vorgeschlagene adaptive PF kann der durchschnittliche Fehler gegen 0 gehen und das quadratische Mittel der Fehler beträgt etwa 1,5 m, was den Ansprüchen von Location Based Services genügt. Weiterhin benötigt der adapative PF nur 30 Partikel was den Rechenaufwand signifikant verringert. In Real-World-Experimenten übertrifft die Genauigkeit des adaptiven PF auch die Resultate herkömmlicher Lokalisierungsalgorithmen wie Linear Least Squares, Nonlinear Least Squares, Min-Max-Algorithmus oder den erweiterten Kalman Filter und andere. Der verteilte PF mit selektiven Gossip-Algorithm hat eine garanatierbare Schätzer Genauigkeit. Des Weiteren wir der Kommunikationsaufwand im Vergleich zu anderen verteilten Konsens-Schemata erheblich reduziert.

Acknowledgement

Four years quickly flies through my hands and run away just in front of my eyes. Recall the first night I arrived in Berlin, I was amazed by the clean air, green woods and quite living place, which can not appear in Beijing. At that time, I was confused about my goal and my future. I was always wondering what I will be in the next four years study. Right now, my study time has ended. The once called future has become present. I find that I earn more than what I expect, e.g. the knowledge, the experience and the friendship. Also comes with some regrets, such as my poor German. In general, I am still grateful to the memorable days in Berlin and in Freie Universität Berlin. Countless thanks would be given to my families and friends.

When I first meet Prof. Dr. Marcel Kyas, I like his sense of humor. I have received many help from him at the beginning days when I join in FU Berlin. I thank him to invite me join in his research topic. Thanks a lot for giving me suggestions especially when I am not familiar with how to do research and how to write a technical paper.

I also appreciate Prof. Dr. -Ing. Jochen Schiller to give me an offer before Christmas eve in 2009, which makes me quite happy and exciting. Prof. Jochen Schiller is a lively person and makes me feel comfortable. A lot of thanks will be given to him for providing such a nice office for my research.

Studying in a foreign country as a minor group student is a lonely journey. When I feel lonely and confused, I can only talk to my wife. At the first two years, it seems to be difficult for both of us. We are like wondering at the cross of life, having no ideas where to go and what is next. However, we are still stick to each other, talk, argue, and encourage. One problem can be solved by two doctor candidates' minds. Gradually, we find our way to make it. Thanks to her support, I can complete so many work that is far beyond my imagination. Nowadays, we are still keeping pace with each other, encourage each other. We are facing the new challenges in the future and we are confident that the bright future is waiting for us. Endless thanks and love would be given to my wife.

Thanks to my parents. They sacrificed a lot to bring me up, educate me and support me without asking paying back. Their support help me walk further away than their initial expected. Now it is the time to pay them back and support them with my fruits. I love them for ever.

Thanks to my Chinese colleagues. I earned a solid friendship. I will give me thanks to Yuan Yang, Qiushi Wang, Yi Sun, Huaming Wu and Tianhui Meng for their help and memorable time. And I would thank to all my colleagues. In the four year research and study, I meet so many friends. Thanks to Dr. Heiko Will, Dr. Thomas Hillebrandt, Dipl.-Ing. Stephan Adler, M.Sc. Simon Schmitt for their selfless support during the research and discussions. I also would like to thank Prof. Dr. Katinka Wolter, Stephanie Bahe, Dr. Emmanuel Baccelli, Dipl.-Inform. Norman Dziengel, Dipl.-Inform. Oliver Hahm, Dipl.-Inform. Felix Shzu-Juraschek, Dipl.-Inform. Matthias Wählisch and all the members of CST for their kindly daily help, which makes me quickly get involved into the life of German research.

My study is under the financial support by China Scholarship Council (CSC) and carried on at the institute of computer science of Freie Universität Berlin. And thanks to the members of ”China Scholarship Council Stipendiaten Vereinigung”. We shared our happiness and never feel lonely. I am very grateful to the Chinese Embassy in Germany, particularly to the staffs in the Education Section. I enjoy the wonderful life in Germany with their good care and support.

Finally, wish all my friends, my families and me a pleasant life! I miss the time and life in Berlin!

Contents

Abstract i

Acknowledgement v

1 Introduction 1

1.1 Wireless Indoor Location and Tracking System 1

1.1.1 Background 1

1.1.2 Distance-based technique 2

1.2 Particle Filter for Target Tracking 3

1.2.1 Localization Algorithms 3

1.2.2 Particle Filter 4

1.3 Contribution of this Dissertation 5

1.3.1 Features of Indoor Environment 5

1.3.2 Problem Statement 5

1.3.3 Proposed Solutions 6

1.4 Publication List 8

1.5 Dissertation Structure 8

2 Particle Filter 11

2.1 Bayesian Model 11

2.2 Basic Particle Filter 12

2.2.1 Importance Sampling 13

2.2.2 Weight Calculation 13

2.2.3 Resampling 14

2.2.4 Bootstrap Particle Filter 15

2.3 The Application to Target Tracking 15

2.4 Related Particle Filtering Solutions 17

2.4.1 Bootstrap Particle Filter 17

2.4.2 Gaussian Particle Filter 18

2.4.3 Constraint Particle Filter . . . 19
2.4.4 Other Particle Filters . . . 20
2.5 The Performance of Particle Filter . . . 21
2.5.1 Estimation Accuracy . . . 21
2.5.2 Computation Complexity . . . 22
2.6 Recent Progress on Particle Filters . . . 22
2.7 Distributed Particle Filters . . . 23
2.7.1 DPF Background . . . 23
2.7.2 Related Algorithms . . . 24
2.8 Summary . . . 26

3 Likelihood Adaptation **27**
3.1 Introduction . . . 27
3.2 Related Work . . . 28
3.3 Problem Statement . . . 29
3.3.1 Particle Filter Revisited . . . 29
3.3.2 Imprecise Measurement Effect . . . 30
3.4 Likelihood Adaptation Method . . . 31
3.4.1 Predicted Measurement . . . 32
3.4.2 Belief Factor θ and Measurement Adaptation . . . 33
3.4.3 Optimal θ and Likelihood Estimation . . . 33
3.5 Adaptive Particle Filter . . . 36
3.5.1 Adaptive Bootstrap Particle Filter (A-BPF) . . . 36
3.5.2 Adaptive Gaussian Particle Filter . . . 37
3.5.3 Adaptive Constraint Particle Filter . . . 38
3.6 Simulation Evaluation . . . 39
3.6.1 Simulation Set-up . . . 39
3.6.2 Particle Filter Modeling . . . 40
3.6.3 Optimal θ for A-BPF . . . 41
3.6.4 Performance Evaluation . . . 43
3.7 Summary . . . 48

4 Dynamic Gaussian Model **49**
4.1 Introduction . . . 49
4.2 Range-based Measurement Error Modeling . . . 50
4.2.1 Problems of Ranging Error Modeling . . . 50
4.3 Error Distribution Fitting . . . 51
4.3.1 Gaussian Distribution . . . 51

4.3.2 Exponential Distribution . . . 52
4.3.3 Gamma Distribution . . . 52
4.3.4 Rayleigh Distribution . . . 53
4.3.5 Log-normal Distribution . . . 53
4.3.6 Weibull Distribution . . . 53
4.3.7 Modeling for Real Indoor Ranging Error . . . 53
4.4 Dynamic Gaussian Approximation . . . 56
4.4.1 General Gaussian Modeling . . . 56
4.4.2 Dynamic Gaussian Modeling . . . 56
4.5 Applications for Nonlinear Filters . . . 58
4.6 Summary . . . 59

5 Real System Evaluation **61**
5.1 Introduction . . . 61
5.2 Time-Of-Flight based Wireless Sensor Network . . . 62
5.3 Indoor Target Tracking System . . . 63
5.3.1 Initialization . . . 64
5.3.2 Preprocessing . . . 65
5.3.3 Anchor Selection . . . 66
5.3.4 Tracking Algorithm . . . 66
5.3.5 Performance Evaluation . . . 70
5.4 Context-Aware Particle Filter . . . 73
5.4.1 Motion Detection using Building Layout . . . 73
5.4.2 Constraint Sampling . . . 74
5.4.3 Likelihood Adaptation . . . 76
5.5 Experiment and Results . . . 76
5.5.1 Hybrid Room and Corridor . . . 78
5.5.2 The building . . . 78
5.6 Summary . . . 84

6 Distributed Particle Filter Solutions **87**
6.1 Introduction . . . 87
6.2 Network Model . . . 88
6.3 Selective Gossiping Algorithm . . . 90
6.4 PSG-DPF Algorithm . . . 91
6.4.1 Local Particle Sampling . . . 91
6.4.2 Coefficient Weights Calculation . . . 92
6.4.3 Significant Particle Selection . . . 92

6.4.4 Iteration Control . . . 93
6.4.5 Estimation Consensus . . . 93
6.5 K-selective Gossip DPF . . . 93
6.5.1 Local Sampling . . . 93
6.5.2 Coefficient Weight . . . 95
6.6 Simulation and Experiment . . . 96
6.6.1 Simulation . . . 96
6.6.2 Indoor Localization Experiment . . . 98
6.7 Summary . . . 100

7 Cramér-Rao Bound Analysis **101**
7.1 Introduction . . . 101
7.2 Cramér-Rao Lower Bound . . . 102
7.3 Fisher Information Matrix Formulation . . . 103
7.3.1 Bayesian Model Revisit . . . 103
7.3.2 Fisher Information Matrix . . . 104
7.3.3 Priori Information . . . 111
7.4 CRLB Analysis for Bayesian Model . . . 114
7.4.1 Environment . . . 114
7.4.2 Performance Comparison . . . 115
7.4.3 Unreliable Priori Information . . . 116
7.4.4 Multiple Anchors . . . 117
7.5 3D Geometric Analysis . . . 117
7.5.1 3D-ranging . . . 118
7.5.2 FIM for 3D-ranging . . . 119
7.5.3 Contours Comparison . . . 122
7.6 Summary . . . 124

8 Conclusion and Future Work **127**
8.1 Contribution . . . 127
8.2 Conclusion . . . 128
8.3 Future Work . . . 129

Bibliography **131**

List of Figures **143**

List of Tables **145**

List of Symbols **147**

Glossary **149**

About the Author **151**

Chapter 1

Introduction

1.1 Wireless Indoor Location and Tracking System

1.1.1 Background

Wireless positioning and localization technologies are essential for wireless communication systems [1]. Service providers want to offer better service based on the user's location, network management system attempts to provide efficient routing protocols, users want to find their own positions. Nowadays, many applications also require accurate location information, such as google map, facebook, weibo, weichat $etc.$. All these requirements motivate the research of localization technologies [2]. For outdoor environment, the technologies are investigated for many years and widely deployed. global positioning system (GPS) is a major technique for navigation [3]. Besides, cell-phone localization and tracking based on the base station is also widely used for cellular networks [4].

Wireless indoor localization and target tracking technologies have become popular in the recent years. It uses the existing wireless facilities to locate or track the target within an indoor environment [5]. In the past few years, the demand for systems has increased dramatically. It has entered the realms of the consumer applications, as well as medical, industrial, public safety, and transport system along with many other applications [6]. Since wireless facilities are deployed and available widely, there is a high demand for accurate positioning using wireless systems, including both indoor and outdoor environment [7].

The widely deployed GPS can not satisfy the requirements of indoor localization system since the signals can not propagate through a direct path through the building from the satellites [8]. Therefore, using existed wireless system for indoor positioning is cost effective and crucial. Locating cellphones requires the information from base station [9]. However, the signals are not reliable for position estimation in indoor environment. WiFi are widely deployed in the building for communication nowadays, which are easily implemented and suitable for indoor

localization [10]. In addition, wireless sensor network (WSN) is easily implemented in the indoor environment for sensing and it is also suitable for indoor localization. Therefore, WiFi and WSN are the main infrastructure for wireless indoor localization and tracking system.

1.1.2 Distance-based technique

In the wireless system, the device, whose position is to be estimated, is called target. Wireless system estimates the position of a mobile target according to the signals from the reference nodes, called anchors [5]. The information carried by the signal for tracking the target is called measurements. The target position is calculated according to the measurements and anchor positions. The widely used measurement techniques are classified as range based measurement and angle based measurement [11, 12]. Angle based measurement, which requires complicated circuit designed, are not suitable for all the indoor tracking applications [13]. While range based measurement technique is cost effective and feasible implemented into the existed wireless systems [14].

Range based measurement is to obtain the distances from target to anchors according to the wireless signal transmission. The distance can be attained through the methods, e.g., received signal strength (RSS), time-of-arrival (TOA), time-difference-of-arrival (TDOA) and time-of-flight (TOF) [2].

The RSS measures the distance according to the signal power [15]. The signal strength is fading during transmission, thus it is easily attained using a proper propagation model. The RSS values are obtained during the normal data communication without presenting the additional bandwidth or energy requirements. Thus, it is relative inexpensive and simple to be implemented in the hardware. Many tracking applications use RSS for localization, such as WSN or smartphone localization using WiFi, or RFID based tracking applications [16]. However, the most accurate propagation model is the free space model, which is only suitable for outdoor environment. For indoor environment, RSS suffers signal noise, multi-path effect and shadowing effect due to the non-line-of-sight (NLOS) [17, 18]. The mixture effects make RSS highly dynamic and unreliable. It is also difficult to construct a indoor propagation model for RSS due to the complicated infrastructure and materials. Therefore, the localization algorithm for RSS should be robust to the noise.

The TOA measures the distance based on the arrival time of the signal. The distance is the product of transmission time and the speed of light. No multi-path signal affects the measurement, since the receiver only consider the direct first arrived signal [1]. It provides more accurate measurement than RSS but with additional scheme, which should filter the time stamp from the data. TOA is sensitive to the NLOS environment [19]. If there is no direct transmission channel, the receiver will detect the first arrival reflected signal, which increase the measurement noise. Besides, to obtain precise TOA, target has to be synchronized with the anchors. Synchronization is also complicated and it is not suitable for memory or battery constrained equipments,

$e.g.$ smartphone or sensors. TOF is more robust to TOA, which calculates the round-trip time of signals [20]. TOF can be obtained through wireless network protocol, but also require additional scheme.

The TDOA estimate the difference of arrival time, which does not need synchronization methods [21]. Therefore, TDOA requires complicated scheme to obtain the measurements. Besides, with the increase number of anchors, the amount of measurements of TDOA increase dramatically. Thus, TDOA is used for some indoor applications which require high estimation accuracy.

1.2 Particle Filter for Target Tracking

1.2.1 Localization Algorithms

When the range measurements are available, the target position is derived according to estimation algorithms. Location and tracking algorithms can be classified into 3 categories: (1) geometric algorithms, (2) convex optimization methods, (3) filtering methods.

Geometric estimators employ range measurements to obtain the position through a geometric way. It constructs the possible regions where the target are located and derives the position. For instance, trilateration method draws circles according to the ranges with 3 anchors [22]. Theoretically, the intersections of these circles are the estimated positions. Similar methods, e.g. min-max algorithm [23], geo-n algorithm [24], centroid method [25] and iterative clustering-based localization algorithm (ICLA) [26], obtain the position all based on the geometric drawing.

Convex optimization methods formulate the range-based positioning problem as finding the optimal point to meet requirements. The commonly used method, such as linear least squares method (LLS) method or nonlinear least squares method (NLLS) method[27, 28] formulates the tracking problem as the least square problem. Both of LLS and NLLS are suitable for the Gaussian environments. For non-Gaussian environments, maximum likelihood (ML) method are employed. ML constructs the log-likelihood function based on the measurement noise probability function and derives the position with the maximum probability.

Filtering methods, such as Kalman filter or particle filter, construct prediction and measurement equations to obtain the position. In the prediction step, they use movement transition information to predict the possible movement state, then calibrate the estimation in the measurement step. Filtering methods achieve better performance for mobile target tracking [29]. Kalman filter achieves high accurate estimation for tracking applications. However, Kalman filter is only suitable for linear Gaussian models. Although extended Kalman filter (EKF) can deal with nonlinear situation, it only attain the suboptimal results. For more complex system, particle filter is required.

1.2.2 Particle Filter

The particle filter (PF) is a Monte-Carlo method based on the Bayesian theorem. It generates random samples and calculates associate weights to represent the posterior probability density function (PDF) of the estimated state [30]. In the PF, the normalized weights indicate the probability of samples. With the increasing samples, the posterior PDF is approaching the actual continuous distribution [31]. Then, the estimated state is obtained through the posterior PDF.

The system model of the PF is not restricted to the linear or Gaussian system [32]. Thus, the application area is wider than the Kalman filter and the EKF. Since the posterior PDF is represented by a set of random samples, it is suitable to describe non-Gaussian distributions, even if the distribution is without an analytical formulation. However, the PF can only obtain suboptimal results due to the sampling method [30]. It can not guarantee the optimal point is always sampled. With increasing the number of samples, the results are almost equivalent to the optimal solutions. Besides, using Bayesian theorem, PF does not need to build a complicated object function to search the optimal solutions like some convex optimization problems [33]. Thus, PF is an efficient tool for the complicated systems.

For target tracking applications, the PF generates samples according to the movement prediction function [34]. To improve the accuracy, it also may consider the speed vectors or acceleration vectors in sometime [35]. The weight is calculated based on the joint likelihood function and the previous weight. The likelihood function is the measurement noise distribution function. Thus, the PF is suitable for many measurement distributions. With previous weight and accurate likelihood function model, PF outperforms other tracking algorithms in many experiments [36, 37, 38].

Indoor tracking system involves many other information as mentioned before. Researchers attempt to use these information to improve the performance. Thus, building the objection function is more complicated using other positioning method. However, using PF to fuse other information is not difficult [39]. The other information is model as the new likelihood function and is fused into the joint likelihood function to calculate the associate weight [40]. Then the estimated posterior PDF is adapted. Thus, PF is highly adaptable to the complicated system without building complex system formulation. Nowadays, it is widely studied and deployed in the wireless positioning system [41]. In addition to the wireless signal information, PFs can also fuse other information, e.g. inertial measurement unit (IMU) [42], simultaneously localization and mapping (SLAM) [43]. The advantage using PFs in IMU and SLAM problems is that PFs convert the observations with noise into likelihoods, and they also assume the trajectories as the results of state transition. Therefore, PFs can fuse different information effectively to draw an estimation even if the problem is complicated. Various kinds of PFs are designed for different indoor scenarios. Therefore, this work mainly focus on the particle filters for the indoor target tracking applications. It attempts to find a solution to improve the performance of particle filters.

1.3 Contribution of this Dissertation

1.3.1 Features of Indoor Environment

Unlike outdoor positioning systems, the deployment of anchors for indoor localization should consider the infrastructure of the building. The randomly or uniformly deployment for outdoor is not feasible for indoor deployment. Because the signal transmission suffers interference in some part of the building. Therefore, the first thing which is concerned by the system is that the anchors can cover the whole playing field efficiently. Then, according to the different parts of the infrastructure, the features of propagation channels are not identical. Thus, the measurement performance for indoor environment varies according to the positions in the building [20].

To obtain an accurate measurement, modeling the ranging measurement is essential [19]. However, for indoor scenario, it is difficult to model the noise precisely. Firstly, modeling N-LOS transmission channel is still an issue. And then, the indoor localization contains hybrid LOS/NLOS environment. It means that the signal is transmitted through a hybrid multipath channel combining the shadowing channel and reflection channel. When considering the complicated infrastructure, it is not easy to get a general accurate error model to describe what kind of distribution the noise follows.

The mobility model for indoor localization is simple. Usually, the movement speed in the building is constant and not too fast compared with cars driving on the road [44]. Besides, the movement follows constrains. The target can not jump out of building and walk through the walls. The movement is context-awared. In most cases, the next movement can be predicted according to the trajectories in the room or corridor. Besides, the building layout is fixed, the trajectories can be calibrated using the layout information.

1.3.2 Problem Statement

This work focus on designing a robust and accurate particle filter to improve the position estimation performance. Particle filter consists of three steps: importance sampling, weight calculation and resampling. Since resampling is to recalculate the particle samples with significant weight, which is related computational performance, it is beyond the scope of this work. Therefore, the major problems for particle filter design is listed as follows:

(1) To improve the estimation accuracy, particle filter needs to generate efficient particles which can represent the possible states for estimation. Usually, the particles are sampled randomly or according to the Markov transition model. Markov transition model indicates that the position state of a target only relies on the previous state but not on the time before. In this case, the state can be predicted by one-step prediction function. It is also practical for indoor target tracking, since the current position is derived from the previous position. However, indoor buildings also provide with other information which can be used for tracking. Thus, sampling

based on the known information is a major issue. Besides, reduce the number of particles without diminishing the performance is also important for the PFs. Large amount of particles require a high computation cost which is not efficient for real time on-line application.

(2) The posterior PDF estimation of the state relies on the associate weights calculation of particles. Three major concerns involve in the weight calculation. The first one is to effective fuse all the information to obtain the joint likelihood function. Some information or range measurements are unreliable, which lead to the inaccurate estimation. Thus, PF should identify the useful reliable information for calculation. Secondly, a feasible range measurement noise model is necessary to the likelihood function. Because the wireless signal is transmitted through the hybrid LOS/NLOS channel in the indoor environment, it is difficult to obtain an accurate model. Nowadays, modeling the measurement is still an important issue in the research community. Finally, the weight calculation should be adaptable to the dynamic environment. Since the environment varies according to the different positions of target in the indoor building, adapting the weight to derive an accurate estimation is necessary.

(3) In the centralized PF, anchors forward the range measurement to the fusion center and the centralized PF estimates the position. It is not practical for an Adhoc network, where the network topology is dynamic and there is no such fixed fusion center. Then, the anchors should estimate the target's position locally and achieve consensus based on some protocols. Thus, distributed particle filter (DPF) is proposed for this scenario. Since the PF requires a number of samples for estimation. The DPF should transmit these particles throughout the whole network, which increases the communication overhead. Thus, for the DPF design, the main goal is to develop a scheme with low communication overhead and comparable estimation performance to the centralized PF. Besides, the DPF should also be robust to the network topology changes.

1.3.3 Proposed Solutions

The major contribution of this work are five folds:

(1) Likelihood adaptation method is introduced to improve the estimation performance of the PF. Firstly, the behavior of the likelihood function in the PF is analyzed. According to the analyzed results, the likelihood calculation is biased from the pre-assumed distribution by the instantaneous measurement noise dynamically. Then, an adaptive likelihood function is constructed accordingly. In this adaptive method, the concepts of predicted measurement and belief factor is defined. And the likelihood function is tuned back to the original distribution according the belief factor. With the derivation of the optimal belief factor, the estimation performance of the PF is greatly improved and is suitable for the indoor target tracking. The analysis and the likelihood adaptation is presented in [45]. Finally, two additional PFs are proposed, which are the adaptive Gaussian PF and the adaptive constraint PF. The simulation results also demonstrate the estimation performance improvement for the target tracking applications.

(2) To describe the complicated indoor environment for noise distribution, a dynamic Gaus-

sian model (DGM) method is proposed. The DGM attempts to describe the noise within a dynamic framework according to the typical ranging measurement. It uses a general Gaussian distribution model to approach different distributions. Then, it adjusts it own distribution heuristically to the multiple environment using the instantaneous measurement information. The impact of complicated infrastructure in the indoor building is ignored. Using the DGM, the scope of likelihood function in the adaptive PFs are extended, since the likelihood function is not restrict to any specific distribution any more. Then, the PFs are highly adaptable to the environment changes. This approach combines the results presented in [46].

(3) Building layout information is used for designing the context aware particle filter (CA-PF). A joint constrain sampling method is proposed according to the target behavior and range measurements. It fully considers the relationship between the movement trajectories and the building layout. Region division and detection is introduced. And layout constrain and measurement constrain are jointly constructed to guarantee generating efficient particles. This method greatly improves the estimation accuracy without imposing computation costs. Besides, for real indoor localization, this method does not need too many particles for estimation. For the real indoor target tracking application, the architecture of the tracking system is introduced. Several components within the architecture are proposed, e.g. initialization, pre-processing, anchor selection, tracking algorithm and performance evaluation. These components jointly assist the localization algorithms obtain an accurate estimation. Several algorithms are integrated within the system and are evaluated. The results for the real system evaluation are described in [47]. The CA-PF outperforms other algorithm in various indoor scenarios.

(4) For local estimation in anchors using DPFs, it is essential to implement a distributed scheme with high accurate estimation and low communication overhead. DPF based on the gossip algorithm is robust to the dynamic network topology. To reduce the communication cost, pairwise selective gossip algorithm is employed for DPF. This method calculates the coefficient weight and exchanges particles with significant weights. To achieve consensus for the whole network, a two round gossip scheme is employed in DPF design. This method is presented in [48]. The estimation performance is guaranteed and communication overhead is reduced. Besides, this scheme does not resampling step, which reduces the computation complexity.

(5) The fundamental performance limit of the position estimation is derived based on the Cramér-Rao lower bound (CRLB). It is also an important metric for the PF estimation. However, some factors should also be considered for the practical analysis. The relative height and the using of prior information also influence the estimation accuracy. The analysis of the CRLB considering the relative height is presented in [49] and the prior information for the CRLB analysis is presented in [50]. The results indicate that the relative height between the target and the anchors can influence the estimation performance significantly. Besides, the estimation accuracy of the adaptive method outperforms other Bayesian process.

1.4 Publication List

The results described in this dissertation have been previously published as:

I Yubin Zhao, Yuan Yang, and Marcel Kyas. Dynamic Searching Particle Filtering Scheme for Indoor Localization in Wireless Sensor Networks. *Proceedings of 9th Workshop on Positioning, Navigation and Communication 2012 (WPNC'13)*, IEEE, 2012.

II Yubin Zhao, Yuan Yang, and Marcel Kyas. Likelihood Adaptation of Particle Filter for Target Tracking using Wireless Sensor Networks. *Proceedings of Globecom 2013*, IEEE, 2013.

III Yubin Zhao, Yuan Yang, and Marcel Kyas. PSG-DPF: Distributed Particle Filter using Pairwise Selective Gossiping for Wireless Sensor Network. *Proceedings of Vehicular Technology Conference, 2013. VTC2013-Spring*, IEEE, 2013.

IV Yubin Zhao, Yuan Yang, and Marcel Kyas. 2D Geometrical Performance for Localization Algorithms from 3D Perspective. *Proceedings of Indoor Positioning and Indoor Navigation (IPIN), 2013 International Conference on*, IEEE, 2013.

V Yubin Zhao, Yuan Yang, and Marcel Kyas. Cramér-Rao Lower Bound Analysis for Wireless Localization Systems using Priori Information. *Proceedings of 11th Workshop on Positioning, Navigation and Communication 2014 (WPNC'14)*, IEEE, 2014.

VI Yubin Zhao, Yuan Yang, and Marcel Kyas. An Adaptive Likelihood Fusion Method using Dynamic Gaussian Model for Indoor Target Tracking. *Proceedings of Internatinal Conference on Acoustics, Speech, and Signal Processing (ICASSP 2014)*, IEEE, 2014.

The extension work and results of the published papers are only written in this dissertation.

1.5 Dissertation Structure

Chapter 2 provides the basic framework of the PF. The general procedure of PF is based on the Bayesian model, and PF approximates the posterior PDF using particle samples with associate weights. It illustrates the how the PF estimates the trajectories for indoor target tracking, and the estimation performance is depicted in the simulation. Besides, some other versions of the PFs is introduced. The recent progress of the PF in the tracking application is reviewed in this chapter. Besides, the concept of DPF and related work are also illustrated.

Chapter 3 illustrates the source of the estimation error of the PF for target tracking. We first analyze the general impact of instantaneous measurement noise which is introduced into the likelihood function and biases the particle filtering estimation. Based on the analysis, a likelihood adaptation method is proposed considering the prior information of measurement and introduce a belief factor θ, which is a tuning parameter for adaptation. The optimal θ is attained by deriving the minimum Kullback-Leibler divergence. This chapter extends the

previous work with three new versions of PFs for TOA range-based target tracking [45]. The simulation results demonstrate that the likelihood adaptation method has greatly improved the estimation performance of particle filters in a high noise environment.

The proposed likelihood adaptation method has a promising performance in the Gaussian environment. However, for the indoor target tracking problem, the ranging error does not follow the Gaussian distribution. Chapter 4 proposes a dynamic Gaussian model (DGM) to describe the indoor ranging error. A general Gaussian distribution for multiple error distribution is constructed firstly. The instantaneous LOS or NLOS error at a typical time is considered as the drift from this general distribution dynamically. Based on this modeling method, the adaptive PFs are extended to the non-Gaussian noise environment. The performance of the adaptive PFs based on the DGM are demonstrated in the simulation. This chapter extends our previous work with the distribution fitting results [46].

According to the previous ideas, the proposed PFs can be implement for the real online system. Chapter 5 firstly introduces the architecture of the indoor target tracking system. The components within the system are described. Then, a new sampling method for real indoor target tracking is proposed. The building layout information is used for PF design. The constraint sampling method is proposed which jointly consider the layout constrain and measurement constrain. The performance is evaluated through a real indoor tracking reference system. Several localization algorithms are compared within this platform, and it demonstrates that context-aware method outperforms other localization algorithms. This chapter combines our previous works with the description for the architecture of the indoor tracking system based on wireless sensor networks and more real experiments [47].

Chapter 6 introduces the design of distributed particle filter using pairwise selective algorithm. Gossip algorithm is robust to the network topology changes. However, the DPF using gossip algorithm increases the communication overhead dramatically. Thus, selective gossip algorithm is introduced in this chapter and integrated with the DPF. To guarantee the estimation performance, two pairwise coefficient weight calculation methods are proposed and a two round gossiping method is employed to guarantee the estimation performance. This chapter extends the previous work [48].

Chapter 7 uses CRLB as a fundamental tool to analyze the PFs. In additional to the basic performance analysis, the prior information is a main factor to the estimation performance of the PFs, since the PFs follow the Bayesian estimation process. The Bayesian process is classified into three types, and the CRLB for each type is derived accordingly. Based on the analysis results, the adaptive PFs with accurate prior information can achieve better performance than other Bayesian processes, just as the simulation and the experiment results indicate. Besides, the impact of the relative height different between the anchors and the target for the estimation accuracy is also analyzed. This chapter is based on previous analytical works [50, 49].

Chapter 8 concludes the whole work and points out the estimation accuracy improvements of the adaptive PFs. For the real world indoor applications, adaptive PF based on context-aware method efficiently fuse available information and outperforms other methods. Finally, we provides the outline of the future research directions.

Chapter 2

Particle Filter

2.1 Bayesian Model

The target tracking problem can be classified as a state estimation problem [32]. In order to estimate a state of a dynamic system, e.g. the location of the target in a wireless system, two common models are applied: first, a model, which describes the state-time evolution, is named system model or prediction model; second, a model, which describes the relationship between the actual state and the noisy measurement, is named observation model or measurement model [30]. The probabilistic formulation for the two models are suitable for the general framework of the Bayesian estimation approach. In the Bayesian estimation approach to the dynamic system, the posterior PDF of the state is constructed based on all the available information [51]. Since the PDF of the state contains all available information, an optimal estimate can be obtained [52].

Bayesian estimation is a recursive algorithm which involves two steps: prediction and measurement [53]. In the prediction, the sequential state $\boldsymbol{x}_t$ evolves according to the state transition model at a give discrete time t, which is given by:

$$\boldsymbol{x}_t = f_t(\boldsymbol{x}_{t-1}, \boldsymbol{q}_t) \tag{2.1}$$

where f_t is defined as the prediction function which is a possibly nonlinear function of the previous state $\boldsymbol{x}_{t-1}$ and the processing noise $\boldsymbol{q}_t$. To simply the model, $\boldsymbol{q}_t$ is assumed to be the additive noise.

In the measurement step, the measurement $\boldsymbol{z}_t$ is the observation function of $\boldsymbol{x}_t$, which is expressed as:

$$\boldsymbol{z}_t = h_t(\boldsymbol{x}_t, \boldsymbol{v}_t) \tag{2.2}$$

where h_t is defined as the observation function which is a possibly nonlinear function of state sequence $\boldsymbol{x}_t$ and measurement noise $\boldsymbol{v}_t$. For each sensor node j, it maintains a local observation

function h_t^j of $\boldsymbol{x}_t$ and local measurement noise v_t^j.

From a Bayesian perspective, the target tracking problem is to construct the posterior PDF $p(\boldsymbol{x}_t|\boldsymbol{z}_{1:t})$ of the state based on (2.1) and (2.2) [54]. The initial PDF of the initial state $\boldsymbol{x}_0$ is $p(\boldsymbol{x}_0|\boldsymbol{z}_0) = p(\boldsymbol{x}_0)$. And the posterior PDF is calculated via Bayes' rule when the measurement $\boldsymbol{z}_t$ is available:

$$p(\boldsymbol{x}_t|\boldsymbol{z}_{1:t}) = \frac{p(\boldsymbol{z}_t|\boldsymbol{x}_t)p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})}{p(\boldsymbol{z}_t|\boldsymbol{z}_{1:t-1})} \tag{2.3}$$

where $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ is the transition probability of a Markov process based on the transition model of (2.1) [55]. The Markov model is applied in (2.3) to indicate that the current state only depends on the previous state. Based on this concept, the prediction function is introduced in (2.1). For target tracking applications, linear prediction function is usually applied [56]. And $p(\boldsymbol{z}_t|\boldsymbol{x}_t)$ is the measurement likelihood function (2.2). And $p(\boldsymbol{z}_t|\boldsymbol{z}_{1:t})$ is the normalizing constant, which follows:

$$p(\boldsymbol{z}_t|\boldsymbol{z}_{1:t-1}) = \int p(\boldsymbol{z}_t|\boldsymbol{x}_t)p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})d\boldsymbol{x}_t \tag{2.4}$$

The formulation of the recursive PDF estimation based on (2.3) is a general conceptual solution, which can not be derived analytically [57]. However, some algorithms still can achieve possible optimal and sub optimal solutions. The optimal solutions include linear Kalman filter and grid-based filters [29]. The Kalman filter assumes the PDF of the state follows normal distributions and the system is linear. In this case, an analytical formulation of the optimal estimation is applied based on the least squares method. The grid-based methods can also obtain the optimal estimation if the state is discrete and finite. If the system is nonlinear and non-Gaussian, the suboptimal algorithms are employed, e.g. extended Kalman filter, approximate grid-based filter and particle filter [58]. For target tracking applications, particle filter is widely used for location estimation [59].

2.2 Basic Particle Filter

Particle filter, which is also denoted as sequential importance sampling (SIS) algorithm, is a Monte-Carlo (MC) method. If the transition model is based on the Markov chain, it is also named Markov chain Monte-Carlo method (MCMC) method. It is a recursive Bayesian filter using the MC simulations. The key method of PF is to employ a randomly generated samples (also denoted as particles) with associated weights to represent the posterior PDF of the state. Then, the state is estimated according to the samples and associated weights. As the number of samples becomes very large, the represented PDF is equivalent to the usual continuous posterior PDF. The recursive working procedure of the PF is illustrated in Fig. 2.1. The major component of PF consists of important sampling, weight calculation and resampling.

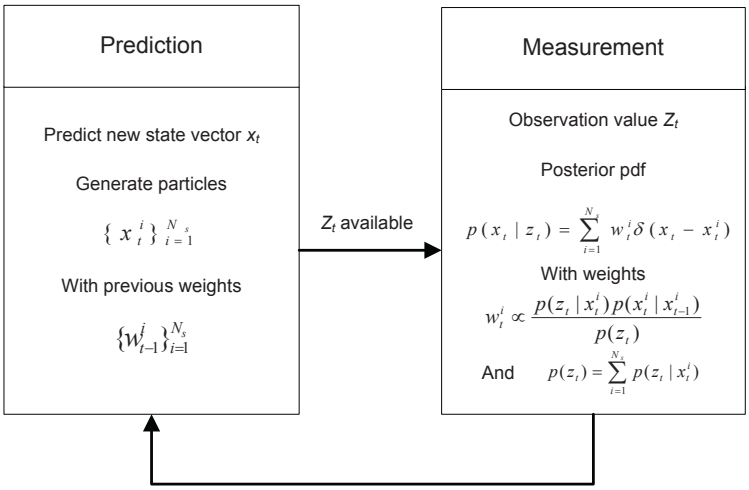


Figure 2.1: The working procedure of particle filter.

2.2.1 Importance Sampling

The PF first generates a particle set $\{\boldsymbol{x}_t^i, w_t^i\}_{i=1}^{N_s}$, where $\{\boldsymbol{x}_t^i\}_{i=1}^{N_s}$ is the sample set with associate weights $\{w_t^i\}_{i=1}^{N_s}$; $\boldsymbol{x}_t^i$ denotes a particle sample, which is a random vector, w_t^i is the associate weight and N_s is the total number of the particles. Note that, $\boldsymbol{x}_t^i$ is drawn from the pre-assumed importance density function $I_s(\boldsymbol{x}_t^i)$ [59]. The density function $I_s(\boldsymbol{x}_t^i)$ determines the previous probability of the particle set. Then, the weight, which presents the probability of particle, can be calculated using the likelihood and density of particles:

$$w_t^i \propto \frac{p(\boldsymbol{x}_t^i)}{I_s(\boldsymbol{x}_t^i)} \tag{2.5}$$

where $I_s(\boldsymbol{x}_t^i)$ can be expressed as:

$$I_s(\boldsymbol{x}_t^i) = I_s(\boldsymbol{x}_t^i | \boldsymbol{x}_{0:t-1}^i, \boldsymbol{z}_{1:t}) \tag{2.6}$$

which indicates that the sample set $\{\boldsymbol{x}_t^i\}_{i=1}^{N_s}$ evolves just like the state $\boldsymbol{x}_t$, which is an element of a sequential state vector$\{\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_t, \ldots\}$. The new samples $\{\boldsymbol{x}_t^i\}_{i=1}^{N_s}$ are also sequential samples, which are derived from the existing samples $\{\boldsymbol{x}_{t-1}^i\}_{i=1}^{N_s}$, and the state of each sample changes according to the previous state and available measurement $\boldsymbol{z}_t$ in (2.6).

2.2.2 Weight Calculation

According to Bayes theorem, the posterior probability is calculated by:

$$p(A|B) = \frac{p(B|A)p(A)}{p(B)} \tag{2.7}$$

Then, the posterior probability of each sampled particle is calculated according to the conditional measurement likelihood and previous probability [60], which is:

$$p(\boldsymbol{x}_t^i|\boldsymbol{z}_t) = \frac{p(\boldsymbol{z}_t|\boldsymbol{x}_t^i)p(\boldsymbol{x}_t^i|\boldsymbol{x}_{t-1}^i)}{p(\boldsymbol{z}_t)} \tag{2.8}$$

By substituting (2.6) and (2.8) into (2.5), the weight of each particle can be obtained:

$$w_t^i \propto w_{t-1}^i \frac{p(\boldsymbol{z}_t|\boldsymbol{x}_t^i)p(\boldsymbol{x}_t^i|\boldsymbol{x}_{t-1}^i)}{I_s(\boldsymbol{x}_t^i|\boldsymbol{x}_{0:t-1}^i, \boldsymbol{z}_t)} \tag{2.9}$$

where $p(\boldsymbol{x}_t^i|\boldsymbol{x}_{t-1}^i)$ is the transition probability for particle $\boldsymbol{x}_t^i$ and $p(\boldsymbol{z}_t|\boldsymbol{x}_t^i)$ is the measurement likelihood, which can be factorized as:

$$p(\boldsymbol{z}_t|\boldsymbol{x}_t^i) = \prod_{j=1}^{N} p(\boldsymbol{z}_t^j|\boldsymbol{x}_t^i) \tag{2.10}$$

Thus the posterior PDF $p(\boldsymbol{x}_t|\boldsymbol{z}_{1:t})$ can be approximated by using delta function:

$$p(\boldsymbol{x}_t|\boldsymbol{z}_{1:t}) \approx \sum_{i=1}^{N_s} w_t^i \delta(\boldsymbol{x}_t - \boldsymbol{x}_t^i). \tag{2.11}$$

with the normalized weight:

$$w_t^i = \frac{w_t^i}{\sum_{i=1}^{N_s} w_t^i} \tag{2.12}$$

2.2.3 Resampling

One of the common problem of the recursive PF is that: after several iterations, some particles evolve with big weights that obtain bigger weights in each iteration. After the normalizing step, all particles but one will have negligible weights. This is called the degeneracy problem for PFs [61]. The degeneracy problem significantly influences the estimation accuracy of the PFs. Therefore, resampling the particles with important weights is necessary.

Resampling is a procedure which deletes the unimportant particles and copies the remaining particles. It increases the variance of particles and improves the estimation accuracy [62]. The main idea is that according to some threshold, the PF self-identify that the quality of particles degrade with lots of negligible weights. Then, PF examines each particle. If the particle has a significant low weight, the PF will drop the particle. The remained particles are copied to compensate the vacancy of dropped particles. Finally, PF reassign equal weights to the new particle set, which is $w_t^i = \frac{1}{N_s}$, as the previous for next iteration [63].

Resampling requires large computation cost, which is far beyond the scope of this work. For

more detail, please refer to M.S. Arulampalam's tutorial [29].

2.2.4 Bootstrap Particle Filter

One of the simplified PF version is named bootstrap particle filter (BPF) [60]. It uses Markov transition model as the importance sampling function $I_s(\boldsymbol{x}_t^i) = p(\boldsymbol{x}_t^i|\boldsymbol{x}_{t-1}^i)$. Then, the weight calculation is simplified as:

$$w_t^i \propto w_{t-1}^i p(\boldsymbol{z}_t|\boldsymbol{x}_t^i) \tag{2.13}$$

Then the weight can be normalized to obtain the estimated posterior PDF:

$$w_t^i = \frac{1}{\mathcal{W}} w_{t-1}^i p(\boldsymbol{z}_t|\boldsymbol{x}_t^i) \tag{2.14}$$

where $\mathcal{W} = \sum_{i=1}^{N_s} w_{t-1}^i p(\boldsymbol{z}_t|\boldsymbol{x}_t^i)$. When considering the resampling step, the previous is assigned as equal weights, $w_t^i = \frac{1}{N_s}$. Then, the weight for each particle only depends on the measurement likelihood $w_t^i \propto p(\boldsymbol{z}_t|\boldsymbol{x}_t^i)$. Therefore, the weight is represented as:

$$w_t^i = \frac{1}{\mathcal{W}} p(\boldsymbol{z}_t|\boldsymbol{x}_t^i) \tag{2.15}$$

where $\mathcal{W} = \sum_{i=1}^{N_s} p(\boldsymbol{z}_t|\boldsymbol{x}_t^i)$.

After obtaining the posterior PDF of the state $\boldsymbol{x}_t$, the estimation can be derived:

$$\bar{\boldsymbol{x}}_t = \sum_{i=1}^{N_s} w_t^i \boldsymbol{x}_t^i. \tag{2.16}$$

2.3 The Application to Target Tracking

PFs are efficient tools for many nonlinear non-Gaussian applications. They have been widely implemented in the wireless tracking system [59]. Since wireless target tracking system is a nonlinear system, and the noise is not always Gaussian, the PF is more suitable than other tracking algorithms. The accurate estimation performance is verified in a variety of experiments.

Consider N wireless nodes deployed in a 2D plane to form a network. The nodes with known positions are denoted as anchors and the mobile device with unknown position is denoted as the target [56]. The anchors positions are assigned as $\boldsymbol{a}_j = [a_j^X, a_j^Y]^T$ where $j \in [1, ..., N]$. The target position is denoted as $\boldsymbol{x}_t = [p_t^X, p_t^Y]^T$. According to Bayes estimation model, the state $\boldsymbol{x}_t$ is evolved with the previous state $\boldsymbol{x}_{t-1}$ according to the prediction equation:

$$\boldsymbol{x}_t = f_t(\boldsymbol{x}_{t-1}) + \boldsymbol{q}_t \tag{2.17}$$

where $f_t()$ is the prediction function, $\boldsymbol{x}_{t-1}$ is the previous state and $\boldsymbol{q}_t$ is the additive prediction

noise, which follows the normal distribution $\boldsymbol{q}_t \sim \mathcal{N}(0, \boldsymbol{Q}_t)$, implying the prediction error of $\boldsymbol{x}_t$. For indoor target tracking system, the movement of target follows a first-order Markov chain [64]. Multiple distributions are used for describing the target trajectory model, e.g. the circular Gaussian PDF, the uniform ring model and radius cone model [64]. In this work, the linear transition model is applied since the movement of the target in the indoor environment is usually linear transition model with a constant speed. If the ranging measurement is obtained in a high frequency, the target is assumed as almost static between the time intervals. Then, the linear model is formulated as:

$$f_t(\boldsymbol{x}_t) = \boldsymbol{F}_t \boldsymbol{x}_t \tag{2.18}$$

where $\boldsymbol{F}_t$ is formulated as the constant linear prediction matrix:

$$\boldsymbol{F}_t = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{2.19}$$

which is an identity matrix to indicate the moving of the target is almost static if the ranging measurement is obtained highly frequently.

Anchors measure the ranging values and forward the measurements to the fusion center. At time t, the fusion center formulates a noisy measurement vector $\boldsymbol{z}_t = (z_t^1, \ldots, z_t^j, \ldots, z_t^N)^T$ with the available ranging value. The relationship between $\boldsymbol{x}_t$ and $\boldsymbol{z}_t$ follows the measurement equation of Bayesian estimation model:

$$\boldsymbol{z}_t = h_t(\boldsymbol{x}_t) + \boldsymbol{v}_t \tag{2.20}$$

where $h_t()$ is the measurement function, and $\boldsymbol{v}_t$ is the additive measurement noise at time t. The range measurement obtained from TOA, TOF or RSS for anchor j is formulated as:

$$z_t^j = h_t^j(\boldsymbol{x}_t) + v_t^j \tag{2.21}$$

where $h_t^j(\boldsymbol{x}_t)$ is the distance between the target and anchor j:

$$h_t^j(\boldsymbol{x}_t) = \sqrt{(p_t^X - a_j^X)^2 + (p_t^Y - a_j^Y)^2} \tag{2.22}$$

Then, the joint measurement function is constructed as:

$$\boldsymbol{z}_t = \begin{pmatrix} \sqrt{(p_t^X - a_1^X)^2 + (p_t^Y - a_1^Y)^2} \\ \vdots \\ \sqrt{(p_t^X - a_j^X)^2 + (p_t^Y - a_j^Y)^2} \\ \vdots \\ \sqrt{(p_t^X - a_n^X)^2 + (p_t^Y - a_n^Y)^2} \end{pmatrix} + \begin{pmatrix} v_t^1 \\ \vdots \\ v_t^j \\ \vdots \\ v_n^j \end{pmatrix} \tag{2.23}$$

where the measurement noise v_t^j of anchor j is conditional independent. Conditional independent means that the measurement only relies on the condition of the current state but the measurements are independent to each other. In the RSS based ranging measurement, the wireless power is lost due to the distance square in the outdoor scenario. However, in the indoor environment, the power loss rate is not always related to the squared distance. For TOA ranging, the transmission time highly relies on the distance. Thus, the distance based measurement is more widely used than the squared distance based measurement.

2.4 Related Particle Filtering Solutions

Various of particle filters are proposed in the literatures. The basic PF is named bootstrap particle filter (BPF), and is also called recursive sequential Monte-Carlo method. Other PFs are derived by appropriate choice of importance sampling density or modification of the resampling step, such as auxiliary particle filter or regularized particle filter. For target tracking applications, two particle filters are proposed with an improved estimation performance, which are Gaussian particle filter and constraint particle filter.

2.4.1 Bootstrap Particle Filter

Bootstrap particle filter firstly makes a prediction of $\boldsymbol{x}_t$ using (2.17). The particles are generated using Gaussian distribution function $\{\boldsymbol{x}_t^i\}_{i=1}^{N_s} \sim \mathcal{N}(\boldsymbol{x}_t, \boldsymbol{Q}_t)$, where $\boldsymbol{Q}_t$ is the covariance of prediction error and the particle $\boldsymbol{x}_t^i$ represents a possible position where the target may stand $\boldsymbol{x}_t^i = [p_t^{X,i}, p_t^{Y,i}]^T$.

The probability for each particle, which is also denoted as measurement likelihood, is calculated as [55]:

$$p(\boldsymbol{z}_t|\boldsymbol{x}_t^i) = \pi_v(\boldsymbol{z}_t - \boldsymbol{z}_t^i) \tag{2.24}$$

where $\boldsymbol{z}_t^i = h_t(\boldsymbol{x}_t^i)$ and π_v is the PDF of measurement noise $\boldsymbol{v}_t$ with variables of z_t^i. For Gaussian

distribution, the likelihood is calculated as:

$$p(z_t^j|\boldsymbol{x}_t^i) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}(z_t^j - h(\boldsymbol{x}_t^i))^2 \sigma_v^{-2}}. \tag{2.25}$$

For Gamma distribution, the particles with $z_t^j - h(\boldsymbol{x}_t^i) < 0$ will be dropped since the invalid value for the distribution function. Then, the likelihoods for the rest particles are formulated as:

$$p(z_t^j|\boldsymbol{x}_t^i) = \frac{\beta^\alpha}{\Gamma(\alpha)} (z_t^j - h(\boldsymbol{x}_t^i))^{\alpha-1} e^{-\beta(z_t^j - h(\boldsymbol{x}_t^i))} \tag{2.26}$$

After resampling the effective particles, the estimation is obtained. The whole algorithm is described in Alg. 1.

Algorithm 1 Bootstrap Particle Filter (BPF) for Tracking

//Importance Sampling
Draw: $\{\boldsymbol{x}_t^i \sim p(\boldsymbol{x}_t^i|\boldsymbol{x}_{t-1}^i)\}_{i=1}^{N_s}$;
for particle $i = 1 : N_s$ **do**
  Likelihood: $p(\boldsymbol{z}_t|\boldsymbol{x}_t^i) = \pi_v(\boldsymbol{z}_t - h_t(\boldsymbol{x}_t^i))$;
  Weight: $w_t^i = p(\boldsymbol{z}_t|\boldsymbol{x}_t^i)$;
end for
Normalizing: $w_t^i = \frac{w_t^i}{\sum_{i=1}^{N_s} w_t^i}$;
Resampling: $\{\boldsymbol{x}_t^i, w_t^i\}_{i=1}^{N_s}$;
State Estimation $\boldsymbol{x}_t = \sum_{i=1}^{N_s} w_t^i \boldsymbol{x}_t^i$;

2.4.2 Gaussian Particle Filter

Gaussian particle filter (GPF) approximates the estimated PDF by the Gaussian distributions using the PF method [57]. GPF assumes that PDF of the state follows Gaussian distribution and it samples particles according to the estimated PDF. Therefore, only mean and covariance of the estimated PDF are calculated and propagated. Due to the simplicity, the GPF is widely used in distributed particle filter applications.

Particles are drawn from Gaussian distribution functions, $\{\boldsymbol{x}_t^i\}_{i=1}^{N_s} \sim \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{Q}_t)$, where $\boldsymbol{\mu}_t$ is the mean value of estimated state and $\boldsymbol{Q}_t$ is the covariance of PDF. The Gaussian PDF evolves according to the transition model: $\boldsymbol{x}_t = f_t(\boldsymbol{x}_{t-1})$. In the two dimensional target tracking application of WSNs, the transition model is assumed as the linear function, which is $\boldsymbol{x}_t = \boldsymbol{F}_t \boldsymbol{x}_{t-1}$ and $\boldsymbol{F}_t$ is the prediction matrix. Thus, the covariance is assumed to propagate to the next time step, which is $\boldsymbol{Q}_{t|t-1} = \boldsymbol{Q}_{t-1}$. Then the initial weight for each particle is calculated as:

$$w_t^i \propto \mathcal{N}(\boldsymbol{x}_t = \boldsymbol{x}_t^i, \hat{\boldsymbol{x}}_t, \boldsymbol{Q}_{t|t-1}) \tag{2.27}$$

When the measurements are available, the weight for each particle updates:

$$w_t^i \propto p(\boldsymbol{z}_t|\boldsymbol{x}_t^i)\mathcal{N}(\boldsymbol{x}_t = \boldsymbol{x}_t^i, \hat{\boldsymbol{x}}_t, \boldsymbol{Q}_{t|t-1}) \tag{2.28}$$

where $p(\boldsymbol{z}_t|\boldsymbol{x}_t^i)$ is the likelihood.

Algorithm 2 Gaussian Particle Filter (GPF)

Prediction: $\hat{\boldsymbol{x}}_t = f_t(\boldsymbol{x}_{t-1})$
Randomly Draw: $\{\boldsymbol{x}_t^i\}_{i=1}^{N_s} \sim \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{Q}_{t|t-1})$
for particle $i = 1 : N_s$ **do**
  Likelihood: $p(\boldsymbol{z}_t|\boldsymbol{x}_t^i) = \pi_v(\boldsymbol{z}_t - h_t(\boldsymbol{x}_t^i))$;
  Weight: $w_t^i \propto p(\boldsymbol{z}_t|\boldsymbol{x}_t^i)\mathcal{N}(\boldsymbol{x}_t = \boldsymbol{x}_t^i, \boldsymbol{\mu}_t, \boldsymbol{Q}_{t|t-1})$;
  Gaussian Distribution Estimation:
  Mean: $\boldsymbol{x}_t = \boldsymbol{\mu}_t = \sum_{i=1}^{N_s} w_t^i \boldsymbol{x}_t^i$
  Covariance: $\boldsymbol{Q}_t = \sum_{i=1}^{N_s} w_t^i (\boldsymbol{\mu}_t - \boldsymbol{x}_t^i)(\boldsymbol{\mu}_t - \boldsymbol{x}_t^i)^T$
end for

2.4.3 Constraint Particle Filter

The constraint particle filter (CPF) randomly samples particles not only based on both the assumed distribution and some constraint conditions [65]. It either draws a constraint region to generate particles or eliminates the particles which do not yield the constraint conditions, $e.g.$ $\boldsymbol{x}_t^i \in c(\boldsymbol{x}_t, \boldsymbol{z}_t)$, where $c(\boldsymbol{x}_t, \boldsymbol{z}_t)$ is the constraint functions. The constraint conditions guarantee the particle generated in the target region with a very high probability.

The constraint conditions are constructed according to different applications. For indoor applications, the constraint conditions are drawn as the region according to the ranging values. Another method builds the constrains according to the building layout map and detail conditions using the geometrical information, such as the positions of doors, windows or walls. The whole algorithm are illustrated in Alg. 6.

Algorithm 3 Constraint Particle Filter (CPF)

Constraint Sampling: $\{\boldsymbol{x}_t^i \in c(\boldsymbol{x}_t, \boldsymbol{z}_t)\}_{i=1}^{N_s}$
for particle $i = 1 : N_s$ **do**
  Likelihood: $p(\boldsymbol{z}_t|\boldsymbol{x}_t^i) = \pi_v(\boldsymbol{z}_t - h_t(\boldsymbol{x}_t^i))$;
  Weight: $w_t^i \propto p(\boldsymbol{z}_t|\boldsymbol{x}_t^i)$;
  Normalizing: $w_t^i = \frac{w_t^i}{\sum_{i=1}^{N_s} w_t^i}$;
  Resampling: $\{\boldsymbol{x}_t^i, w_t^i\}_{i=1}^{N_s}$;
end for

2.4.4 Other Particle Filters

In addition to the above mentioned versions of the PFs, there are also some other kinds of PFs. These versions are derived from the BPF by using different importance sampling density or modifying the resampling step. Here, several PF algorithms are briefly introduced.

Auxiliary Particle Filter

The importance density function in the auxiliary PF is the same as the BPF. However, the auxiliary PF has two rounds of weight calculations [66]. In the first round, the weight for each particle is prone to the likelihood function. After resampling, the weights are calculated based on the posterior PDF just as the BPF. If the measurement noise is low, the auxiliary PF is better than the BPF. However, if the noise is high, the performance of BPF is not improved by using the auxiliary PF. Thus, it is not suitable for the indoor environment.

Regularized Particle Filter

The resampling step in the BPF may arise the problem of loss of diversity among the particles. If this problem is not addressed properly, it may lead to "particle collapse", which makes all the particles the same and a poor representation of the posterior density. Thus, a regularized particle filter (RPF) is proposed to solve such problem [67].

The difference between the RPF and the BPF is the resampling stage. The particles in the RPF are resampled within a kernel but not from the original particles. The kernel function is constructed to minimize the mean integrated square error between the true posterior PDF and the corresponding regularized estimated PDF. The estimation of the RPF is optimal if the density follows the Gaussian distribution.

Unscented Particle Filter

The unscented particle filter (UPF) consists of a particle filter which employs an unscented Kalman filter (UKF) to generate the samples [68]. The samples are drawn via an unscented transformation as the sigma points in the UKF. Then, the samples are updated according to the Kalman filter steps and the posterior PDF is estimated as the approximated Gaussian distribution.

The PDF estimation is approaching to the true distribution using the UPF. Combining both advantages of the PF and UKF, the UPF outperforms the BPF, the extended Kalman filter and the UKF. Besides, the UPF is not limited to the Gaussian distribution. The computation complexity is higher than BPF.

Rao-Blackwellised Particle Filter

Rao-Blackwellization is to use the Kalman filter for the part of the state model which is linear and the particle filter as the other part [69]. The linear part of the state is estimated through the

Kalman filter procedure while the rest part is obtained by particle filter with the assist of the Kalman filter.

The number of particles is required to be large if the state is with high dimension. The Rao-Blackwellised particle filter (RBPF) avoids such problem and achieves small number of particles for the state estimation. It is widely used in the positioning systems [59, 70]. However, for the range based target tracking system, the dimension of the state space is not high. The RBPF has almost the same performance of the BPF.

2.5 The Performance of Particle Filter

2.5.1 Estimation Accuracy

One major concerned problem of particle filter is the estimation accuracy. As mentioned before, with the increased number of particles, the posterior PDF estimation is equivalent to the actual distribution. Theoretical analysis has been investigated [65, 58]. The mean square error of PF is derived to be $\frac{C}{N_s}$, where C is a constant error factor, which is related to the measurement error and N_s represents the number of particles [58]. References indicate that the estimation error weakly converges to 0 with the increased number of particles [65, 58, 55]. However, the infinite number of particles can not be achieved due to the limited computing memories.

Besides, the measurement noise still affects the estimation performance. Thus, an alternative way is to employ Cramer-Rao lower bound (CRLB) to indicate the optimal performance [71]. Since PF uses MAP estimation method, which is an unbiased estimation method, it can achieve to CRLB in some typical scenarios [56].

Another evaluation method is the Kullback-Leibler divergence (KLD) estimation method, which compares the estimated PDF with the actual PDF [52]. It is a promising method to indicate the performance of the PDF estimation directly. However, the KLD evaluation requires a very high computation cost [72]. Thus, since the indoor target tracking application is a low dimensional state estimation problem, the mean square error is already enough to evaluate the performance.

For the target tracking application, several factors may influence the estimation performance of the PF [55]. They are listed below:

1. Measurement noise. According to CRLB analysis, for the unbiased estimator, the high measurement noise leads to the high estimation error. This is also applicable to the PFs. Besides, the choice of using the distribution of the measurement noise also influence the estimation accuracy. For the indoor localization, it is not easy to obtain an accurate ranging error model, which degrades the estimation performance.
2. Number of particles. Although using the unlimited particles can not achieve the perfect estimation, increasing the number of particles can lead to the improvement of the estima-

tion. However, more particles leads to more computation cost, which is essential for the real time application

3. Available information. The ranging measurement is a useful information for PFs. Every anchor forwards a measurement to the system at each time. Then, more anchors can provide more information for PFs and get more accurate position. In addition to the ranging measurement, the building map, the inertial measurement unit within the target node, and even the sound and the image through the camera are the available data, which can be transformed into the likelihood function and improve the estimation. The building map, which is easily converted into layout information, is the main important information in this thesis.

2.5.2 Computation Complexity

Thus computation cost of the PF depends on the dimension of the state model, the number of the particles and the complexity of the likelihood calculation. In the weight calculation stage, the computation complexity is $\mathcal{O}(N_s)$, where N_s is the number of the particles [30]. Since the PF should calculate the weight for each particle, the computation complexity is proportional to the grid size N_s. In the resampling stage, the PF re-orders the particles according to the weight, then, the computational complexity is $\mathcal{O}({N_s}^2)$, which increases the computation cost dramatically.

For each particle, the high dimensional state model and the large amount of information also increase the computation cost. Besides, if the state model is with high dimension, more particles are required to be generated, which also increase the complexity. Fortunately, the dimension for the indoor target tracking is not high, and only a small group of particles is needed for estimation.

2.6 Recent Progress on Particle Filters

PF can achieve promising performance for target tracking with building a proper movement model and measurement model [56]. Using a proper movement model, high quality particles can be generated based on the prediction. When consider the indoor movement, a jump Markov model is used to indicate the transition state of moving target [73]. Hypothesis testing method is also employed to examine the orientation of the target, which can provide a more accurate trajectory [74]. When the estimated PDF is largely biased from the predicted PDF, the restarting procedure is proposed to calibrate the estimation [75]. Pishdad, $et\ al.$ proposed an optimal function sampling method, which can avoid a degeneracy problem for indoor tracking [76].

Another solution to generate effective particles is to construct constrains with the floor plan or building layout. The system initially stores the map data, and sets the constrains dynamically according to the estimated position and the surround environment [77]. When the estimated state involves with multiple variables, $e.g.$ orientation, speed, accelerator $etc.$, a smart and detailed constrain condition is needed [78]. The particles can not get through a wall or jump out of

the window. With the constrain condition, the weight of unrealistic particle is assigned with a extreme low value [79, 78].

The performance is also improved with an efficient measurement model for likelihood estimation. One solution is to employ the characters of wireless techniques for estimation. A simple model is to employ a Gaussian process for modeling [80]. When considering a general model, a shadowing model is proposed to indicate the measurement noise [81]. However, the hybrid LOS/NLOS channel can varies significantly, thus, measurement noise is described with different distribution according to the situations. Nicoli $et\ al.$ developed a recursive Bayes estimator considering the NLOS and multipath effects [73]. Kushki $et\ al.$ proposed a nonlinear and non-parametric filtering method integrated with RSS fingerprints [82]. A hybrid information of TOA and RSS is exploited for likelihood calculation in [37]. Since the indoor transmission channel is complicated, interactive multi-model is proposed [17, 18], which adapts the likelihood calculation to different scenarios. Further, the combination of TOA/RSS with NLOS and multipath mitigation method is introduced in [83]. These methods rely on the understanding the features of wireless transmission techniques.

An alternative solution is to directly deal with the imprecise and unreliable measurement based on the data fusion theory [84]. The unreliable data problem is studied within several works $e.g.$ Dempster-Shafer fusion theory [85], fuzzy and possibility theory [86], transferable belief model and probability theory [87]. Adaptive PF belongs to the probability theory. Farahmand $et\ al.$ proposed an adaptive likelihood method using constrained set-membership function to mitigate the heavy tail distribution [65]. Stordal $et\ al.$ developed a method which adapt the weights of PF between the Gaussian distribution and uniform distribution [88]. These methods are heuristic ways for adaptation.

PF can also be used for indoor tracking applications with other techniques. For example, the inertial measurement unit (IMU) system use PF for pedestrian localization or dead reckoning [38, 89, 79, 42]. For RFID tag tracking, PF estimates the trajectories with the aggregate binary measurement model [90, 91]. The applications of PF for SLAM [43] and camera tracking [92] are beyond the scope.

2.7 Distributed Particle Filters

2.7.1 DPF Background

In a dynamic network, when not all the nodes can work collaboratively in a centralized way, the distributed method is required. So is the particle filter. When the wireless network is dynamic without a stable topology, it is difficult to maintain a fusion center and forward all the measurements to the fusion center. Besides, for some typical applications, the anchors of the system want to know the target position locally for convenience. Thus, it is essential to develop a efficient

DPF [93]. The DPF is attractive for large-scale distributed estimation problems in the wireless network. Nowadays, the computing capability for a single chip has increased dramatically. It is possible to maintain a large particle set for computation. Thus, the DPF is widely used in the sensor networks [94], robotic networks [95], networks of unmanned aerial vehicles [96], and the networks of cameras [97]. Possible applications include the target tracking [94], environmental monitoring [98], health-care monitoring [99], pollution source localization, chemical plume tracking and surveillance [100].

Developing DPF may face two challenges. The first challenge is to guarantee the estimation performance. To obtain an global estimation, the DPF can not access to all the local measurements. Then, the DPF may use some approximations to estimate the posterior PDF of the state. In addition, using approximations also reduce the computation and communication cost. As a result, the DPF do not perform as well as the centralized PF. Second, consensus must be reached for all the participating nodes. This requires the exchange of information, therefore, the communication between the network nodes are the major concern. Since the Bayesian estimation framework does not change eventually, the centralized PF works as the same as local PF. Thus, researchers considers what kind of information should be exchanged among nodes in DPF. Nodes are not desired to transmit the raw data to each other. Because, the other nodes should identify the reliability of the measurement. Also, for the privacy concerns, it is not wise to forward the raw data to other nodes. Therefore, the exchanged information should be either the particles or estimated states.

Implementing DPF should also consider the constrains of the energy, computation and communication within a single node. The algorithm also has to meet the application requirements, e.g. operational lifetime, latency/reaction time, robustness to the link and node failure, network mobility and scalability. Sometimes, the tradeoffs of the implementation is needed.

2.7.2 Related Algorithms

The working procedure of the DPF is almost the same as the centralized PF. The particles $\{\boldsymbol{x}_t^i\}_{i=1}^{N_s}$ are generated according to the Markov transition model, and the associate weights $\{w_t^i\}_{i=1}^{N_s}$ are calculated based on the measurement likelihood. If the measurement noise is independent to each node, the measurement function is divided into several local functions, which is indicated in (2.21). Then the measurement likelihood can be factorized into several local likelihood functions [101]:

$$p(\boldsymbol{z}_t|\boldsymbol{x}_t) = \prod_{j=1}^{N} p(z_t^j|\boldsymbol{x}_t) \tag{2.29}$$

which means that the weights can be determined locally. Then, DPF uses some protocols to achieve consensus to obtain a global estimation.

According to the data communication methods between the nodes, the DPFs are classified

into two classes: the first type is the statistics dissemination-based DPFs, which exchange the local estimated data between the nodes; the second type is the measurement dissemination-based DPFs, which only exchange raw or quantized measurements [102]. The second type is not widely used due to the privacy concerns in the network. The first type is further divided into two subsets, which are Leader-Agent-based DPF (LA-based DPF) and consensus-based DPF.

Leader-Agent-based DPF (LA-based DPF)

In the LA-based DPF, the data are aggregated through a path which is formed by several nodes [103]. The node which aggregates the data and makes the final estimation is called leader. The transmitted data can be the particles with associate weights or the approximated parameters of the estimated distribution. Then, the global estimation is obtained in the lead by using its local data and neighborhood data. There can be multiple leaders in the network according to the position of the target. When the target moves far away from the current leader, a new leader is chosen and the data are propagated to the new one.

The LA-based DPFs use message passing to propagate particles through the network. In this case, a spanning tree or a Hamiltonian cycle is required for topology management [31, 103, 51, 104]. However, maintaining a topology is not robust for a dynamic system, especially for the link or node failure. Besides, the handover of the posterior PDF from one leader to another is a communication-intensive task, especially for a high dimensional state. Thus, it is not widely deployed in the target tracking system.

Consensus-based DPF

In the consensus-based DPF, all the nodes run the local PFs simultaneously and achieve a global posterior PDF. In this case, a typical consensus algorithm is required, which establishes an agreement according to certain criterias for all the nodes. Using the consensus algorithm, each node transmits some data to a set of neighborhoods. Using the consensus algorithm offers several advantages: (1) consensus algorithm only require local communication between neighbor nodes, which needs no routing protocols or global knowledge about the network. (2) consensus algorithm is robust to the network topology changes or link failures [105, 106, 107, 108]. It is suitable to the mobile network.However, one disadvantage of the consensus-based DPF is the large communication overhead.

Sun Hwan $et\ al.$ proposed a Markov chain distributed particle filter (MCDPF) which transmitted particles through the network and calculated the associated weights based on local likelihood functions [109]. The estimation performance is similar to the centralized PF. However, this increases the communication overhead dramatically because all the particles are used for estimation.

One solution to reduce the communication cost is to compress the particle set into several parameters such as Gaussian Mixture Model (GMM) to approximate the posterior PDF. D.Gu

$et\ al.$ proposed a gossip-based expectation-maximization (EM) algorithm to estimate the parameters of a mixture approximation of the global posterior likelihood [110]. O.Hlinka $et\ al.$ employed a joint likelihood consensus algorithm to fuse estimations of a Gaussian distributed particle filter [111]. Boris $et\ al.$ used Gaussian product approximation method to estimate the global posterior PDF [112]. All of the mentioned algorithms effectively reduce the communication overhead but diminish the estimation accuracy. Another solution, which is newly develop in the recent years, is to use selective gossip, which selects the effective particles to transmit instead of all the particles [113]. If the procedure is properly handled, the estimation can achieve to the centralized PF with less communication overhead.

2.8 Summary

The procedure of particle filter and its application for target tracking is reviewed. Several popular versions of PFs are introduced. The measurement noise, error distribution, the number of particles and the amount of available anchors can influence the estimation performance of the particle filter and the computation complexity. For indoor scenario, the recent progress is illustrated, which involves several movement and measurement models for indoor environment. Distributed particle filter is also introduced in this chapter. The major concern of implementing the DPF is the communication overhead. The challenges and recent solutions are briefly reviewed. The consensus-based DPF is a promising direction for the target tracking application, because it can guarantee all the local anchors obtain the same estimation.

Chapter 3

Likelihood Adaptation

3.1 Introduction

The main source of the location estimated error is the ranging error. No matter of what localization algorithm, the ranging error can not be fully mitigated. Nonlinear filters also suffer the same problem. Although the prior information is involved in the estimation, the estimation error still exists and increases with the rise of ranging error. Conventional ways use more prior information, e.g. identifying the NLOS ranging according to the geometric information in the building, however, the estimation is still not accurate due to the highly dynamic environment. In this chapter, we discuss the main source of the estimated error, and how it influences the tracking system. Since nonlinear filters follow the Bayesian estimation framework, and PF can represent the key idea of Bayesian method, we will focus on the impact of ranging error for PF. Then, the analysis is extended to other methods in the following chapters.

Particle filter estimates the posterior PDF of the target state according to the random particle samples and associate weights. The likelihood function (LF), which relies on the measurement and noise distribution, has a large proportion in determining the particle weights. In a high measurement noise environment, the instantaneous noise is hard to predict [114]. Then LF calculation in PF based on the noisy measurement is deviated from the true value for each particle and further affects the weights of particles and the posterior PDF of the state. Therefore, the measurement noise will cause unreliable posterior PDF estimation. We investigate the conventional impact of the unreliable data for the PF. When the instantaneous noise, is inevitably introduced into the likelihood function, the likelihood calculation for each particle in the PF is biased from the original assumed distribution. Consequently, a biased estimator is attained. Therefore, our goal is to reduce the noise in LF according to our analysis. The major work in this chapter are three folds:

1) We propose a general adaptive method, which is suitable for different wireless techniques,

with combining the prior information and a tuning parameter: the prior information is the predicted measurement for each sensor based on the predicted state; and belief factor $\boldsymbol{\theta} \in [0, 1]$ is the tuning parameter, which adapts the LF to a more accurate one. By tuning $\boldsymbol{\theta}$, the impact of noise for likelihood calculation is reduced.

2) In order to obtain the optimal performance, we use Kullbeck-Leibler divergence (KLD), which is an efficient metric to compare two likelihoods, to derive the optimal $\boldsymbol{\theta}$. The optimal $\boldsymbol{\theta}$ can achieve the minimum KLD and lead to the lowest estimation error.

3) Three versions of particle filters are improved based on our adaptation method, which are bootstrap particle filter (BPF), Gaussian particle filter (GPF) and constraint particle filter (CPF). The simulations demonstrate that the proposed algorithms effectively reduce the estimation error and have robust performance in the high noisy wireless environment.

3.2 Related Work

One major problem of PF is the estimation accuracy and its robustness to high noise. Theoretical analysis has been reporte by D. Crisan $et\ al.$ [65, 58]. The mean square error of PF is derived to be C/N_s, where C is a constant value and N_s represents the number of particles [58]. When all the particles are propagated throughout the whole network, the distributed PF has the same estimation accuracy of centralized PF [109]. References indicate that the average estimation error converges to 0 with the increased number of particles [65, 58, 55]. However, the infinite particles are not available in the real application due to the computational delay and memory constrains. In addition, in the real implementation, estimation error still exist due to measurement noise.

Adaptive filtering methods are investigated to develop a robust PF. One solution is to employ the characters of wireless techniques for estimation. In this case, the precise distribution of wireless transmission model is considered and formulated. Nicoli $et\ al.$ developed a recursive Bayesian estimator considering the NLOS and multipath effects, which uses an indicator to identify the NLOS ranging measurements [73]. Kushki $et\ al.$ proposed a nonlinear and non-parametric filtering method integrated with RSS fingerprints [82]. A hybrid information of TOA and RSS is exploited for likelihood calculation in [37]. Further, the combination of TOA/RSS with NLOS and multipath mitigation method is introduced in [83], different distributions are formulated for both TOA and RSS. One of the main problem is that these methods rely on the specific wireless technique and are still not robust to the environment change. Besides, the information obtained from the wireless characteristics still contains errors, although less than the original measurement, it is still not mitigated and influences the final estimation.

An alternative solution is to directly deal with the imprecise and unreliable measurement based on the data fusion theory [84]. The unreliable data problem is studied within several works $e.g.$ Dempster-Shafer theory [85], fuzzy and possibility theory [86], transferable belief model

and probability theory [87]. These methods have different frameworks for estimation which are not easy to combine with nonlinear filters. Adaptive PF belongs to the probability theory. Farahmand $et\ al.$ proposed an adaptive likelihood method using constrained set-membership function to mitigate the heavy tail distribution [65]. The theoretical analysis indicates that the constrain based method can improve the estimation performance. Stordal $et\ al.$ developed a method which adapts the weights of PF between the Gaussian distribution and uniform distribution [88]. Similar methods can be extended to other distributions. These methods are heuristic ways for adaptation. One major problem, to our best knowledge, remains, which is the impact of instantaneous measurement noise for LF.

3.3 Problem Statement

3.3.1 Particle Filter Revisited

Lots of versions of PFs are developed during the recent years. All the PFs rely on the particle sampling, weight calculation and resampling to obtain the posterior PDF of the state. The posterior PDF estimation directly relates to the weight calculation, which depends on the likelihood function. Thus, to simplify our analysis, the bootstrap particle filter (BPF) is applied in this chapter.

BPF first generates a particle set $\{\boldsymbol{x}_t^i, w_t^i\}_{i=1}^{N_s}$ according to the importance density function $I_s(\boldsymbol{x}_t^i)$, where $\{\boldsymbol{x}_t^i\}_{i=1}^{N_s}$ is the sample set with associate weights $\{w_t^i\}_{i=1}^{N_s}$; i and N_s denote the particle number and the total number of particles respectively. The importance density function $I_s(\boldsymbol{x}_t^i)$ is assumed as the Markov transition model, which simplifies the Bayesian formulation. Thus the posterior PDF $p(\boldsymbol{x}_t|\boldsymbol{z}_t)$ of state $\boldsymbol{x}_t$ can be approximated by using delta function, which is a generalized impulse symbol:

$$p(\boldsymbol{x}_t|\boldsymbol{z}_t) \approx \sum_{i=1}^{N_s} w_t^i \delta(\boldsymbol{x}_t - \boldsymbol{x}_t^i). \tag{3.1}$$

And the weight of each particle can be calculated as follows [29]:

$$w_t^i \propto p(\boldsymbol{z}_t|\boldsymbol{x}_t^i) \tag{3.2}$$

Then the weight can be normalized to obtain the estimated posterior PDF:

$$w_t^i = \frac{1}{\mathcal{W}} p(\boldsymbol{z}_t|\boldsymbol{x}_t^i) \tag{3.3}$$

where $\mathcal{W} = \sum_{i=1}^{N_s} p(\boldsymbol{z}_t|\boldsymbol{x}_t^i)$. In the above functions, the generated particles are assumed to be of high quality, which have important weights. If some particles are not important, they are

dropped during the resampling stage.

BPF only considers the likelihood function and the prior information, e.g. $p(\boldsymbol{x}_t^i|\boldsymbol{x}_{t-1}^i)$, is not involved in the calculation. Thus, the impact of prior information has limits for the location estimation.

3.3.2 Imprecise Measurement Effect

It is clearly observed that the likelihood function calculation relies on the measurement vector and measurement noise distribution. The the measurement error influences the likelihood function and further influences the location estimations. When the measurement $\boldsymbol{z}_t$ and particle samples $\{\boldsymbol{x}_t^i\}_{i=1}^{N_s}$ are available, The measurement likelihood for each particle is calculated as [55]:

$$p(\boldsymbol{z}_t|\boldsymbol{x}_t^i) = \pi_v(\boldsymbol{z}_t - \boldsymbol{z}_t^i) \tag{3.4}$$

where $\boldsymbol{z}_t^i = h_t(\boldsymbol{x}_t^i)$ and π_v is the probability density function of measurement noise $\boldsymbol{v}_t$ with variables of $\boldsymbol{z}_t^i$. Here, researchers pre-assume that π_v represents the exact noise distribution, which is depicted as the solid curve in the example shown in Fig. 3.1. In Fig. 3.1, the noise follows zero-mean Gaussian distribution. Then, the measurement $\boldsymbol{z}_t$ is assumed as a reliable vector. If we consider noise value $\boldsymbol{v}_t$ at time t, then we obtain the following expression by substituting (2.20) into (3.4):

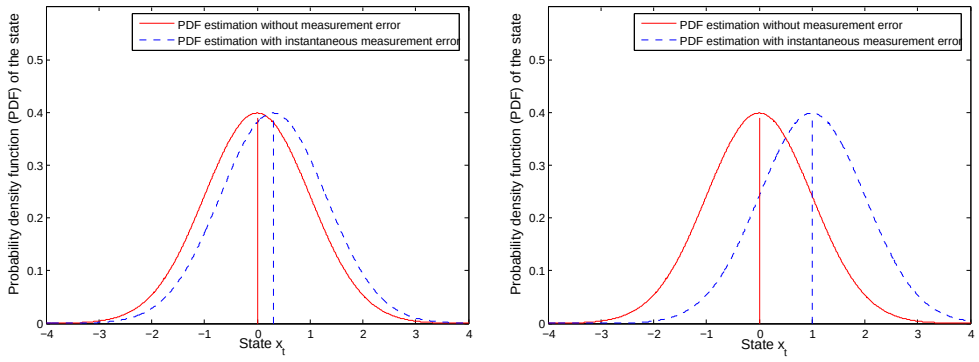
$$p(\boldsymbol{z}_t|\boldsymbol{x}_t^i) = \pi_v(h_t(\boldsymbol{x}_t) + \boldsymbol{v}_t - \boldsymbol{z}_t^i) \tag{3.5}$$

where an unpredictable instantaneous noise $\boldsymbol{v}_t$ is introduced into the likelihood function, which means that the pre-assumed the distribution is biased with a instantaneous value. Although $\boldsymbol{v}_t$ follows the zero-mean Gaussian distribution for the long term statistical results, it is hard to be estimated at a typical time unless the real measurement vector is obtained.

Equation (3.5) is illustrated as the dash curve in Fig. 3.1, which is a biased non-zero-mean Gaussian distribution. It is deviated from the original assumption due to considering the instantaneous value, $\boldsymbol{v}_t$. If $\boldsymbol{v}_t$ is 0, the likelihood function $p_A(\boldsymbol{z}_t|\boldsymbol{x}_t^i)$ is the exact distribution of measurement noise:

$$p_A(\boldsymbol{z}_t|\boldsymbol{x}_t^i) = \pi_v(h_t(\boldsymbol{x}_t) - \boldsymbol{z}_t^i) \tag{3.6}$$

where p_A represents the actual assumed distribution and (3.6) is the exact likelihood, then we would have optimal filtering with the increasing number of particles. However, in most real cases, $\boldsymbol{v}_t$ is inevitably not 0, and π_v is biased by $\boldsymbol{v}_t$ in (3.5) which is involved into the real calculation and leads to the inaccurate estimation. To distinguish the concepts of the original distribution and biased distribution, we define the impact of instantaneous measurement noise $\boldsymbol{v}_t$ on the likelihood calculation as the *imprecise measurement effect*. When $\boldsymbol{v}_t$ is becoming larger, the gap between two curves are increasing as shown in Fig. 3.1(b), which degrades the



(a) Instantaneous measurement error $\boldsymbol{v}_t = 0.3$ (b) Instantaneous measurement error $\boldsymbol{v}_t = 1$

Figure 3.1: Example: the two likelihood functions. The solid curve represents the actual likelihood function obtained without measurement error and the dash curve represents the deviated likelihood function by the instantaneous measurement error

estimation accuracy significantly. Existing methods do not solve this problem and this effect can happen in almost all the likelihood functions in PFs. Therefore, our goal is to develop an adaptation method to mitigate $\boldsymbol{v}_t$ and approach the likelihood calculation to the exact value.

3.4 Likelihood Adaptation Method

We adapt the likelihood function $p(\boldsymbol{z}_t|\boldsymbol{x}_t^i)$ towards the actual case $p_A(\boldsymbol{z}_t|\boldsymbol{x}_t^i)$ in order to reduce the instantaneous measurement effect. Besides the general Gaussian model, the prior information is required for adaptation. The adaptation is only for likelihood calculation in PF, thus, other parts of PF are not influenced. Then, the proposed method can be integrated into many other particle filtering algorithms.

Our adaptation method consists of two steps: the first step is to obtain a predicted measurement $\hat{\boldsymbol{z}}_t$ according to the previous state; the second step is to adapt the likelihood function based on $\hat{\boldsymbol{z}}_t$ and a belief factor $\boldsymbol{\theta}$ which is a tuning parameter for adaptation. The structure of adaptive PF which integrates with the predicted measurement and $\boldsymbol{\theta} = [\theta^1 \ \ldots \ \theta^N]$ is illustrated in Fig. 3.2. The blue components are the procedures of original PF, and the red components are the new methods for the adaptive PF. In the original PF, the likelihood function is determined by the measurement $\boldsymbol{z}_t$, whereas in our algorithm, the likelihood calculation is based on both of $\hat{\boldsymbol{z}}_t$ and $\boldsymbol{z}_t$. Belief factor $\boldsymbol{\theta}$ is used to adapt the value between $\hat{\boldsymbol{z}}_t$ and $\boldsymbol{z}_t$, and for each ranging measurement in $\boldsymbol{z}_t$, there is a tuning parameter according to $\boldsymbol{\theta}$. Although the likelihood calculation is different from the original PF, the structure of PF does not change too much. The proposed method can also be integrated into other kinds of PFs.

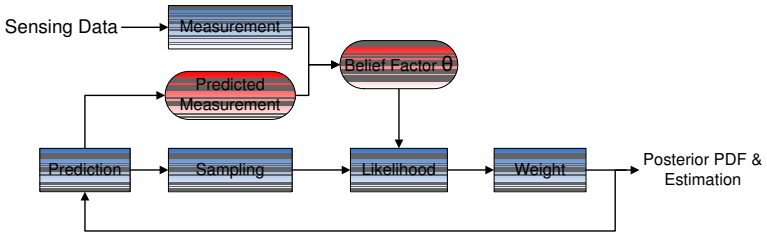


Figure 3.2: The architecture of particle filter integrated with adaptive likelihood method. The common particle filter does not contain the prior measurement and the belief factor.

3.4.1 Predicted Measurement

Predicted measurement is derived based on the prediction state. It is a virtual value and indicates what the measurement should be with a given current predicted state. It has no impact on the real measurement since systems only focus on the real measurement. However, it can be the reference for the real measurement since the real state is unknown to the system. According to the hidden Markov model, the real state is hidden in the observed measurement. Predicted measurement is to appear the hidden state on the virtual observed value. Thus, the calculated procedure can just follow the Bayesian procedure. The calculation steps are as follows: $\hat{\boldsymbol{x}}_t$ denotes the prediction value of $\boldsymbol{x}_t$:

$$\hat{\boldsymbol{x}}_t = f_t(\boldsymbol{x}_{t-1}) \tag{3.7}$$

where $\boldsymbol{x}_{t-1}$ is the estimation at previous time $t-1$. When considering the processing noise $\boldsymbol{q}_t$, we denote $\hat{\boldsymbol{x}}_t$ as:

$$\hat{\boldsymbol{x}}_t = \boldsymbol{x}_t + \boldsymbol{q}_t \tag{3.8}$$

where $\boldsymbol{q}_t$ is assumed to be the additive noise and follows normal distribution $\boldsymbol{q}_t \sim \mathcal{N}(0, \boldsymbol{Q}_t)$; $\boldsymbol{Q}_t$ is the covariance at time t. The state evolves from the previous state and contains a prediction noise. Thus, such state can lead to an observation according to the measurement function, although system may not get these values in the real case. Then we obtain a predicted measurement:

$$\hat{\boldsymbol{z}}_t = h_t(\hat{\boldsymbol{x}}_t) = h_t(\boldsymbol{x}_t + \boldsymbol{q}_t) \tag{3.9}$$

in which $\hat{\boldsymbol{z}}_t$ indicates the prediction of measurement derived from $\hat{\boldsymbol{x}}_t$. The predicted measurement $\hat{\boldsymbol{z}}_t$ is not the actual measurement but it provides the prior information which indicates that if the real state follows the prediction, the observed real measurement should be like the predicted measurement. Thus, the predicted measurement can be the reference based on the prior information.

3.4.2 Belief Factor θ and Measurement Adaptation

Belief factor $\boldsymbol{\theta}$ is the tuning parameter for the predicted measurement and it is used to adapt the measurement $\boldsymbol{z}_t$ to approach the actual measurement $h_t(\boldsymbol{x}_t)$. Note that, our goal is to reduce the instantaneous noise effect of the likelihood function $p_A(\boldsymbol{z}_t|\boldsymbol{x}_t^i)$ in (3.6). Since $p_A(\boldsymbol{z}_t|\boldsymbol{x}_t^i)$ can not be obtained directly, we use $\hat{\boldsymbol{z}}_t$ and $\boldsymbol{z}_t$ to approach $p_A(\boldsymbol{z}_t|\boldsymbol{x}_t^i)$ with $\boldsymbol{\theta}$. Then, the adaptive likelihood function $p_{AL}(\boldsymbol{z}_t|\boldsymbol{x}_t^i)$ is constructed as:

$$p_{AL}(\boldsymbol{z}_t|\boldsymbol{x}_t^i) = \pi_v(\boldsymbol{\theta}\hat{\boldsymbol{z}}_t + (1-\boldsymbol{\theta})\boldsymbol{z}_t - \boldsymbol{z}_t^i) \tag{3.10}$$

where p_{AL} indicates the adaptive likelihood. The belief factor $\boldsymbol{\theta}$ indicates how much trust $p_{AL}(\boldsymbol{z}_t|\boldsymbol{x}_t^i)$ assigns to the predicted measurement $\hat{\boldsymbol{z}}_t$. If $\boldsymbol{\theta} = 0$, $p_{AL}(\boldsymbol{z}_t|\boldsymbol{x}_t^i) = \pi_v(\boldsymbol{z}_t - \boldsymbol{z}_t^i) = p(\boldsymbol{z}_t|\boldsymbol{x}_t^i)$ which equals to the measurement likelihood; whereas when $\boldsymbol{\theta} = 1$, $p_{AL}(\boldsymbol{z}_t|\boldsymbol{x}_t^i) = \pi_v(\hat{\boldsymbol{z}}_t - \boldsymbol{z}_t^i)$ which only trusts the predicted measurement. Either case is not applicable for the real estimation, since both the predicted measurement and the measurement information should be fused together to derive a final estimation.

Here, two problems remain: (1)With introducing $\boldsymbol{\theta} < 1$, the measurement error in $\boldsymbol{z}_t$ is suppressed, but the error of $\hat{\boldsymbol{z}}_t$ is involved, which arouse a question: in what condition can our method reduce imprecise measurement effect? (2) If our method can reduce this effect, a proper $\boldsymbol{\theta}$ is required. So is there an optimal $\boldsymbol{\theta}$ that can achieve the best performance? In the real case, both the predicted measurement and noisy measurement are not reliable, thus, adapting the likelihood calculation between these values may arise new estimation error. Finding the optimal $\boldsymbol{\theta}$ can achieve the best estimation performance.

3.4.3 Optimal θ and Likelihood Estimation

Tuning $\boldsymbol{\theta}$ to find an optimal value is one possible solution, but in the real-time system, it is time consuming to find an optimal $\boldsymbol{\theta}$ with lots of samples. Since we intend to compare our adapted PDF with the actual PDF, the effective evaluation method is Kullback-Leibler divergence (KLD) [52].

KLD, which also denotes as relative entropy, quantifies the difference between two distributions for a wide range of applications. If $p_1(x)$ and $p_2(x)$ indicate two different distributions, the KLD is formulated as:

$$D_{KL}(p_1||p_2) = \int p_1(x) \log \frac{p_1(x)}{p_2(x)} dx = E_{p_1}[\log \frac{p_1(x)}{p_2(x)}] \tag{3.11}$$

where $D_{KL}(p_1||p_2)$ denotes the KLD, which is defined as the expected value of log-likelihood ration between p_1 and p_2. KLD is an non-negative distance between two different distributions, which is $D_{KL}(p_1||p_2) \geq 0$. It can be shown that $D_{KL}(p_1||p_2) = 0 \Leftrightarrow p_1(x) = p_2(x)$, which indicates that the two distributions are the same. And small $D_{KL}(p_1||p_2)$ indicates $p_1(x)$ is

similar to $p_2(x)$. KLD is not a true distance, since it is not symmetric in general and does not satisfy the triangle inequality. However, it is still appropriate as the benchmark to evaluate the closeness of a probability density to another. In addition, the KLD is a convex function, thus, it can be used for the optimization problem [115]. The KLD is an efficient method to evaluate the performance of PF [52]. It compares the estimated particles with a given known distribution. If the KLD is small, the estimated PDF is approaching to the real distribution of the state.

We employ the KLD as the object function to find optimal $\boldsymbol{\theta}$ which minimizes the distance between $p_A(\boldsymbol{z}_t|\boldsymbol{x}_t)$ and $p_{AL}(\boldsymbol{z}_t|\boldsymbol{x}_t)$. Here, we use $p_A(\boldsymbol{z}_t|\boldsymbol{x}_t)$ as the objective distribution and employ $p_{AL}(\boldsymbol{z}_t|\boldsymbol{x}_t)$ as the tuning distribution with parameter $\boldsymbol{\theta}$. Then, the KLD function is constructed as

$$\begin{aligned} D_{KL}(p_A||p_{AL}) &= \int p_A(\boldsymbol{z}_t|\boldsymbol{x}_t) \log \frac{p_A(\boldsymbol{z}_t|\boldsymbol{x}_t)}{p_{AL}(\boldsymbol{z}_t|\boldsymbol{x}_t)} d\boldsymbol{z}_t^i \\ &= \int \pi_v(h_t(\boldsymbol{x}_t) - \boldsymbol{z}_t^i) \log \frac{\pi_v(h_t(\boldsymbol{x}_t) - \boldsymbol{z}_t^i)}{\pi_v(\boldsymbol{\theta}\hat{\boldsymbol{z}}_t + (1-\boldsymbol{\theta})\boldsymbol{z}_t - \boldsymbol{z}_t^i)} d\boldsymbol{z}_t^i \end{aligned} \tag{3.12}$$

Then, optimal $\boldsymbol{\theta}$ is attainted with $D_{KL}(p_A||p_{AL})$:

$$\boldsymbol{\theta} = \arg\min D_{KL}(p_A||p_{AL}) \tag{3.13}$$

If p_A and p_{AL} are based on the same Gaussian distribution function, then (3.12) is expressed as [116]:

$$D_{KL}(p_A||p_{AL}) = \frac{||h_t(\boldsymbol{x}_t) - [\boldsymbol{\theta}\hat{\boldsymbol{z}}_t + (1-\boldsymbol{\theta})\boldsymbol{z}_t]||^2}{2\boldsymbol{R}_t} \tag{3.14}$$

where $\boldsymbol{R}_t$ is the covariance of $\boldsymbol{v}_t$. Since $\boldsymbol{R}_t$ is independent on $\boldsymbol{\theta}$, the objective function is simplified as:

$$\hat{\boldsymbol{\theta}} = \arg\min \ ||h_t(\boldsymbol{x}_t) - [\boldsymbol{\theta}\hat{\boldsymbol{z}}_t + (1-\boldsymbol{\theta})\boldsymbol{z}_t]||^2 \tag{3.15}$$

which turns to be a least-squares approximation problem [117]. Thus, it can also be denoted that finding the optimal $\boldsymbol{\theta}$ by using KLD is equivalent to find the measurement which is approaching to the real measurement. Thus, if the error is assumed to be Gaussian, finding the optimal solution is a least-squares approximation problem.

Since $\hat{\boldsymbol{z}}_t$ is the nonlinear functions of the prediction noise $\boldsymbol{q}_t$ according to (3.9), it is difficult to obtain an analytical optimal result. The optimal value can be found using Gaussian-Newton method. But it should be calculated iteratively to reach the final result. Thus, we attempt to linearize the problem and reduce the complexity which will further simplify the PF estimation. To linearize the formulation, we use first order Taylor series expansion at $\boldsymbol{x}_t$ to linearize (3.9) :

$$\hat{\boldsymbol{z}}_t \approx h_t(\boldsymbol{x}_t) + \frac{\partial h_t(\boldsymbol{x}_t)}{\partial \boldsymbol{x}_t}\boldsymbol{q}_t \tag{3.16}$$

where $\partial h_t(\boldsymbol{x}_t)/\partial \boldsymbol{x}_t = [\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^X}, \frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^Y}]^T$ is the partial differential of $h_t(\boldsymbol{x}_t)$ with respect to $\boldsymbol{x}_t$:

$$\begin{aligned} \frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^X} &= \frac{p_t^X - a_j^X}{\sqrt{(p_t^X - a_j^X)^2 + (p_t^Y - a_j^Y)^2}} \\ \frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^Y} &= \frac{p_t^Y - a_j^Y}{\sqrt{(p_t^X - a_j^X)^2 + (p_t^Y - a_j^Y)^2}} \end{aligned} \tag{3.17}$$

In this case, the problem is simplified but loss some information. And substitute (3.16) and (2.20) into (3.15), we obtain:

$$||h_t(\boldsymbol{x}_t) - [\boldsymbol{\theta}\hat{\boldsymbol{z}}_t + (1-\boldsymbol{\theta})\boldsymbol{z}_t]||^2 \approx ||\boldsymbol{\theta}\frac{\partial h_t(\boldsymbol{x}_t)}{\partial \boldsymbol{x}_t}\boldsymbol{q}_t + (1-\boldsymbol{\theta})\boldsymbol{v}_t||^2 \tag{3.18}$$

Therefore, the problem is converted into a linear optimization problem, which is solvable analytically by expressing the objective as the convex quadratic function[117]:

$$\boldsymbol{F}_t(\boldsymbol{\theta}) = \boldsymbol{\theta}\frac{\partial h_t(\boldsymbol{x}_t)}{\partial \boldsymbol{x}_t}\boldsymbol{Q}_t[\frac{\partial h_t(\boldsymbol{x}_t)}{\partial \boldsymbol{x}_t}]^T\boldsymbol{\theta}^T + [1-\boldsymbol{\theta}]\boldsymbol{R}_t[1-\boldsymbol{\theta}]^T \tag{3.19}$$

where $\boldsymbol{Q}_t$ and $\boldsymbol{R}_t$ are the covariance of $\boldsymbol{q}_t$ and $\boldsymbol{v}_t$.

Then, the optimal $\boldsymbol{\theta}$ can be obtained if and only if the first order derivation is equal to 0:

$$\frac{\partial \boldsymbol{F}_t(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = 2\boldsymbol{\theta}\frac{\partial h_t(\boldsymbol{x}_t)}{\partial \boldsymbol{x}_t}\boldsymbol{Q}_t[\frac{\partial h_t(\boldsymbol{x}_t)}{\partial \boldsymbol{x}_t}]^T - 2\boldsymbol{R}_t + 2\boldsymbol{\theta}\boldsymbol{R}_t = 0 \tag{3.20}$$

Then, the unique $\boldsymbol{\theta}$ is derived:

$$\boldsymbol{\theta} = \frac{\boldsymbol{R}_t}{\frac{\partial h_t(\boldsymbol{x}_t)}{\partial \boldsymbol{x}_t}\boldsymbol{Q}_t[\frac{\partial h_t(\boldsymbol{x}_t)}{\partial \boldsymbol{x}_t}]^T + \boldsymbol{R}_t} \tag{3.21}$$

where $\frac{\partial h_t(\boldsymbol{x}_t)}{\partial \boldsymbol{x}_t}\boldsymbol{Q}_t[\frac{\partial h_t(\boldsymbol{x}_t)}{\partial \boldsymbol{x}_t}]^T + \boldsymbol{R}_t$ is positive define since Q_t and R_t indicate the covariance of distributions.

Since $\boldsymbol{\theta}$ is the belief factor for the prior measurement, (3.21) indicates that when the measurement noise is high with a large $\boldsymbol{R}_t$, $\hat{\boldsymbol{z}}_t$ offers more contribution than the noisy measurement. In other words, when the prediction covariance $\boldsymbol{Q}_t$ is larger than $\boldsymbol{R}_t$, our method should assign more belief to $\boldsymbol{z}_t$. In this case, $\hat{\boldsymbol{z}}_t$ is useless and can introduce more estimation error. It means that our method is useful when the measurement noise is higher than the prediction error, and the optimal $\boldsymbol{\theta}$ exists. Fortunately, in the wireless tracking system, the measurement noise is always high. Besides, if the measurement noise is low, the prediction error is also small due to the Bayesian filtering estimation. Therefore, our method can effectively improve the estimation accuracy but not much in a small measurement noise environment.

Note that, if the measurements of each anchor node are conditionally independent, $p(\boldsymbol{z}_t|\boldsymbol{x}_t) = \prod p(z_t^j|\boldsymbol{x}_t)$ where z_t^j denotes the measurement value for jth anchor node, our adaptive likeli-

hood method can be implemented in the local sensor node and integrated with the DPFs. Since we focus on the estimation accuracy of PF in this chapter, the design of DPF will be discussed in chapter 6.

3.5 Adaptive Particle Filter

3.5.1 Adaptive Bootstrap Particle Filter (A-BPF)

BPF is a particle filter with low computation cost [29]. In the BPF, the random particles are sampled according to the importance density function, which is mainly based on the Markov transition model. For target tracking applications, it is easy to generate particles based on the prior distribution of the state, which simplifies the computational complexity. In the BPF, the weight for each particle is calculated according to the previous weight and the likelihood function:

$$w_t^i \propto w_{t-1}^i p(\boldsymbol{z}_t|\boldsymbol{x}_t^i) \tag{3.22}$$

which only depends on the previous weight w_{t-1}^i and the likelihood $p(\boldsymbol{z}_t|\boldsymbol{x}_t^i)$. Since the weight will be assigned equally in the resampling stage, the previous weights are equal. Then, the weight mainly depends on the likelihood function in the current time step. Thus, the estimation accuracy is influenced by the measurement noise as we analyzed.

Here we propose an adaptive bootstrap particle filter (A-BPF) to reduce the noise effect of the BPF. The algorithm of the A-BPF is presented in Alg. 4. Our adaptation method is implemented in the importance sampling part and the weight adaptation part of particle filter.

In A-BPF, the particles are sampled as the Markov process just as the same as the BPF. The particles follow the distribution according to some importance sampling function. Here, the prior Gaussian distribution function is used to indicated the potential importance sampling function, where the expectation is the predicted state $\hat{\boldsymbol{x}}_t$ and the covariance is the prediction error covariance $\boldsymbol{Q}_t$. And the prediction state $\hat{\boldsymbol{x}}_t$ is obtained through:

$$\hat{\boldsymbol{x}}_t = f_t(\boldsymbol{x}_{t-1}) \tag{3.23}$$

where the previous state $\boldsymbol{x}_{t-1}$ is calculated based on expectation of the previous particles: $\boldsymbol{x}_{t-1} = \sum_{i=1}^{N_s} w_{t-1}^i \boldsymbol{x}_{t-1}^i$. Then, we obtain the predicted measurement $\hat{\boldsymbol{z}}_t$ according to (3.9).

When the measurement $\boldsymbol{z}_t$ is available, the adapted measurement likelihood $p(\boldsymbol{z}_t|\boldsymbol{x}_t^i)$ for each particle is calculated as (3.10). And then particle weight is normalized $w_t^i \propto w_{t-1}^i p_{AL}(\boldsymbol{z}_t|\boldsymbol{x}_t^i)$, which determines the posterior PDF of the estimated state $\boldsymbol{x}_t$. Finally, $\boldsymbol{x}_t$ is attained through: $\boldsymbol{x}_t = \sum_{i=1}^{N_s} w_t^i \boldsymbol{x}_t^i$. The importance sampling and resampling parts are still the same, and the predicted measurement and optimal $\boldsymbol{\theta}$ is easy to obtain based on the analytical formulation.

Therefore, our likelihood adaptation does not introduce much computation complexity to the original PFs.

Algorithm 4 Adaptive Bootstrap Particle Filter (A-BPF)

Prediction: $\hat{\boldsymbol{x}}_t = f_t(\boldsymbol{x}_{t-1})$;
Prediction Measurement: $\hat{\boldsymbol{z}}_t = h_t(\hat{\boldsymbol{x}}_t)$;
//Importance Sampling
Draw: $\{\boldsymbol{x}_t^i \sim p(\boldsymbol{x}_t^i|\boldsymbol{x}_{t-1}^i)\}_{i=1}^{N_s}$;
//Measurement Adaptation
for Particle $i = 1 : N_s$ **do**
  Likelihood: $p_{AL}(\boldsymbol{z}_t|\boldsymbol{x}_t^i) = \pi_v(\boldsymbol{\theta}\hat{\boldsymbol{z}}_t + (1-\boldsymbol{\theta})\boldsymbol{z}_t - h_t(\boldsymbol{x}_t^i))$;
  Weight: $w_t^i = w_{t-1}^i p_{AL}(\boldsymbol{z}_t|\boldsymbol{x}_t^i)$;
end for
Normalizing: $w_t^i = \frac{w_t^i}{\sum_{i=1}^{N_s} w_t^i}$;
Resampling: $\{\boldsymbol{x}_t^i, w_t^i\}_{i=1}^{N_s}$;
State Estimation $\boldsymbol{x}_t = \sum_{i=1}^{N_s} w_t^i \boldsymbol{x}_t^i$;

3.5.2 Adaptive Gaussian Particle Filter

Gaussian particle filter (GPF) approximates the estimated PDF by Gaussian distributions using the particle filter method. GPF assumes that PDF of the state follows Gaussian distribution and it samples particles according to the estimated PDF. Therefore, only mean and covariance of the estimated PDF are calculated and propagated. Due to the simplicity, the GPF is widely used in distributed particle filter applications since only some parameters are transmitted throughout the network.

Particles are drawn from Gaussian distribution functions, $\{\boldsymbol{x}_t^i\}_{i=1}^{N_s} \sim \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{Q}_t)$, where $\boldsymbol{\mu}_t$ is the mean value of estimated state and $\boldsymbol{Q}_t$ is the covariance of the state PDF. The Gaussian PDF evolves according to the transition model: $\boldsymbol{x}_t = f_t(\boldsymbol{x}_{t-1})$. In the two dimensional target tracking application of WSNs, the transition model is assumed as the linear function, which is $\boldsymbol{x}_t = \boldsymbol{F}_t\boldsymbol{x}_{t-1}$ and $\boldsymbol{F}_t$ is the prediction matrix. Thus, the covariance is assumed to propagate to the next time step, which is $\boldsymbol{Q}_{t|t-1} = \boldsymbol{Q}_{t-1}$. Then the initial weight for each particle is calculated as:

$$w_t^i \propto \mathcal{N}(\boldsymbol{x}_t = \boldsymbol{x}_t^i, \hat{\boldsymbol{x}}_t, \boldsymbol{Q}_{t|t-1}) \tag{3.24}$$

When the measurements are available, the weight for each particle updates:

$$w_t^i \propto p(\boldsymbol{z}_t|\boldsymbol{x}_t^i)\mathcal{N}(\boldsymbol{x}_t = \boldsymbol{x}_t^i, \hat{\boldsymbol{x}}_t, \boldsymbol{Q}_{t|t-1}) \tag{3.25}$$

where $p(\boldsymbol{z}_t|\boldsymbol{x}_t^i)$ is the likelihood. Since the likelihood is also independent of the other parts of GPF, our adaptation method can be integrated into it which is named as adaptive Gaussian particle filter (A-GPF). The procedure of A-GPF is illustrated in Alg. 5:

Algorithm 5 Adaptive Gaussian Particle Filter (A-GPF)

Prediction: $\hat{\boldsymbol{x}}_t = f_t(\boldsymbol{x}_{t-1})$
Calculate $\hat{\boldsymbol{z}}_t$ based on (3.9)
Randomly Draw: $\{\boldsymbol{x}_t^i\}_{i=1}^{N_s} \sim \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{Q}_{t|t-1})$
for particle $i = 1 : N_s$ **do**
  Likelihood: $p_{AL}(\boldsymbol{z}_t|\boldsymbol{x}_t^i) = \pi_v(\boldsymbol{\theta}\hat{\boldsymbol{z}}_t + (1-\boldsymbol{\theta})\boldsymbol{z}_t - h_t(\boldsymbol{x}_t^i))$;
  Weight: $w_t^i \propto p(\boldsymbol{z}_t|\boldsymbol{x}_t^i)\mathcal{N}(\boldsymbol{x}_t = \boldsymbol{x}_t^i, \boldsymbol{\mu}_t, \boldsymbol{Q}_{t|t-1})$;
  Gaussian Distribution Estimation:
  Mean: $\boldsymbol{x}_t = \boldsymbol{\mu}_t = \sum_{i=1}^{N_s} w_t^i \boldsymbol{x}_t^i$
  Covariance: $\boldsymbol{Q}_t = \sum_{i=1}^{N_s} w_t^i (\boldsymbol{\mu}_t - \boldsymbol{x}_t^i)(\boldsymbol{\mu}_t - \boldsymbol{x}_t^i)^T$
end for

The difference between A-BPF and A-GPF is: the prediction error in A-BPF follows an arbitrary assumed distribution whereas A-GPF uses Gaussian distribution to indicate such distribution. The assumed distribution in A-BPF is obtained based on the statistical analysis and the estimation of A-BPF can be accurate if the assumed distribution is correct. The estimated Gaussian distribution of A-GPF is influenced by the measurement noise and the estimation can not be as accurate as A-BPF in a high noise environment.

3.5.3 Adaptive Constraint Particle Filter

Constraint particle filter (CPF) randomly samples particles not only based on both the assumed distribution and some constraint conditions [65]. It either draws a constraint region to generate particles or eliminates the particles which does not yield the constraint conditions, $e.g.$ $\boldsymbol{x}_t^i \in c(\boldsymbol{x}_t, \boldsymbol{z}_t)$, where $c(\boldsymbol{x}_t, \boldsymbol{z}_t)$ is the constraint functions. The constraint conditions guarantee the particle generated in the target region with a very high probability. But if the weight calculation is influenced by instantaneous noise, the estimation may have large error. Therefore, the likelihood adaptation method is essential for improving CPF.

The constraint conditions can be set up according to different applications. We will detail the conditions for the range-based target tracking in the next section. After sampling the particles, the prediction is obtained based on the prediction function. Then, our adaptation method is used and the weight calculation follows the same procedure as A-BPF, which is illustrated in Alg. 6. The major difference between BPF and CPF is the additional constraints for particles, which can improve the performance of BPF with additional information. However, when the measurements are not accurate, the constraints can mislead the estimation. The adaptive constraint particle filter (A-CPF) can avoid this problem by adapting the likelihood into an accurate value and lead CPF to obtain the precise estimation.

For range-based localization systems, the constraint region can be drawn by bounding box algorithm, which is robust to the measurement noise [118]. Using z_t^j to denote jth range measurement for node j with the position $[a_j^X, a_j^Y]$ in the two dimensional playing field, and

Algorithm 6 Adaptive Constraint Particle Filter (A-CPF)

Constraint Sampling: $\{\boldsymbol{x}_t^i \in c(\boldsymbol{x}_t, \boldsymbol{z}_t)\}_{i=1}^{N_s}$
Prediction: $\hat{\boldsymbol{x}}_t = f_t(\boldsymbol{x}_{t-1})$
Calculate $\hat{\boldsymbol{z}}_t$ based on (3.9)
for particle $i = 1 : N_s$ **do**
  Likelihood: $p_{AL}(\boldsymbol{z}_t|\boldsymbol{x}_t^i) = \pi_v(\boldsymbol{\theta}\hat{\boldsymbol{z}}_t + (1-\boldsymbol{\theta})\boldsymbol{z}_t - h_t(\boldsymbol{x}_t^i))$;
  Weight: $w_t^i \propto p(\boldsymbol{z}_t|\boldsymbol{x}_t^i)$;
  Normalizing: $w_t^i = \frac{w_t^i}{\sum_{i=1}^{N_s} w_t^i}$;
  Resampling: $\{\boldsymbol{x}_t^i, w_t^i\}_{i=1}^{N_s}$;
end for

$\boldsymbol{x}_t = [p_t^X, p_t^Y]$ denotes the target position at time t, then the geometric constraint region is:

$$\begin{cases} s_{x,t}^{min} = max\{a_j^X - z_t^j\}_{j=1}^N \\ s_{x,t}^{max} = min\{a_j^X + z_t^j\}_{j=1}^N \\ s_{y,t}^{min} = max\{a_j^Y - z_t^j\}_{j=1}^N \\ s_{y,t}^{max} = min\{a_j^Y + z_t^j\}_{j=1}^N \end{cases} \tag{3.26}$$

where N is the number of sensor nodes. Then the particles are sampled within this region. Usually, the particles are generated uniformly according to the maximum entropy principle and all particles are assigned with equal weights.

3.6 Simulation Evaluation

3.6.1 Simulation Set-up

Our scheme is evaluated in the simulation of target tracking. We randomly deploy 100 sensors in a two-dimensional square $100m \times 100m$ region. One target runs through a circle path in anticlockwise direction with a constant angular speed. All the sensors can sense the target and measurement the distance based on TOA method. The algorithms are implemented in a centralized way. Since we just evaluate the estimation performance of the proposed algorithm, the network topology or the placement of fusion center is beyond the scope. The simulations consider the problems for the real implementations, e.g. the random deployment of the wireless sensor nodes, the ranging measurement noise, the random trajectories of the target. Thus, the simulation results are comparable to the real-world experiments. The real-world experiments are presented in Chapter 5.

Sensor node j is assigned coordinations $\boldsymbol{a}_j = [a_j^X, a_j^Y]^T$ and the target position state at time t is $\boldsymbol{x}_t = [p_t^X, p_t^Y]^T$. Sensor nodes intensively measure ranging values which are the distances

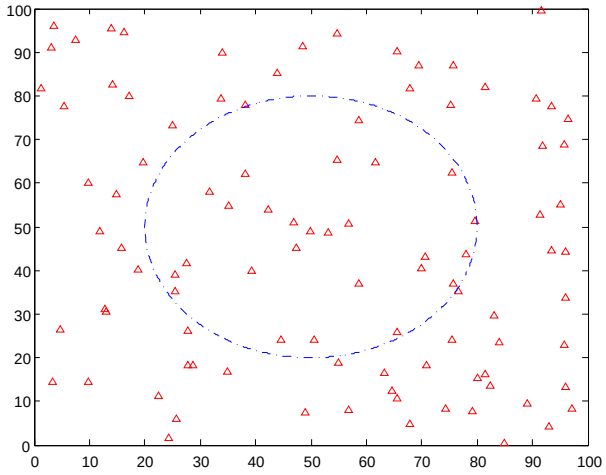


Figure 3.3: Network Deployment and Target Trajectory. The triangle marks the positions of sensor nodes and the dash line marks the target's trajectory

between the target and the nodes. The measurement noise follows Gaussian distribution. Then, the measurement for node j is denoted as:

$$z_t^j = h_t^j(\boldsymbol{x}_t) + v_t^j = \sqrt{(p_t^X - a_j^X)^2 + (p_t^Y - a_j^Y)^2} + v_t^j, \tag{3.27}$$

where $v_t^j \sim \mathcal{N}(0, \sigma_v^2)$ is the measurement noise. The measurement noise for each sensor node is identical independent distributed, so $\boldsymbol{R}_t = diag(\sigma_v^2)$. And the measurement probability for each particle in sensor node j,which is $p(z_t^j|\boldsymbol{x}_t^i)$, can be expressed as:

$$p(z_t^j|\boldsymbol{x}_t^i) = \frac{1}{\sqrt{2\pi\sigma_v^2}} e^{-\frac{1}{2}(z_t^j - h(\boldsymbol{x}_t^i))^2 \sigma_v^{-2}}. \tag{3.28}$$

3.6.2 Particle Filter Modeling

We employ linear model as the prediction function. Since the ranging values are frequently measured, the target can not move in a large distance during two measurement intervals. Thus, the prediction function (2.17) can be expressed as:

$$\hat{\boldsymbol{x}}_t = \boldsymbol{x}_t + \boldsymbol{q}_t \tag{3.29}$$

Initially, we assume that the prediction noise covariances is:

$$\boldsymbol{Q}_t = \begin{pmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{pmatrix} \tag{3.30}$$

where the prediction noise in each coordinate is assumed to be identical independent distributed, so $\sigma_x^2 = \sigma_y^2$.

In the particle sampling stage, the samples are drawn according to the Gaussian distribution $\mathcal{N}(0, \sigma_v^2)$ in BPF and A-BPF. The prior distribution is used throughout the whole simulation. Thus, the sampling procedure is same for BPF and A-BPF. The only difference is the likelihood adaptation method.

The optimal $\boldsymbol{\theta}$ for A-BPF can be obtained by substituting the measurement vector into (3.21). However, a simplified equivalent version can be used due to the sensor's independent measurement feature. The measurement function can be factorized into each independent single range measurement. The measurement noise covariance can be re-written as a single variable σ_v^2. Then, substitute $h_t^j(\boldsymbol{x}_t)$ into (3.21), the optimal $\boldsymbol{\theta}$ is calculated as:

$$\boldsymbol{\theta} = \frac{\sigma_v^2}{\sigma_v^2 + \sigma_x^2} \tag{3.31}$$

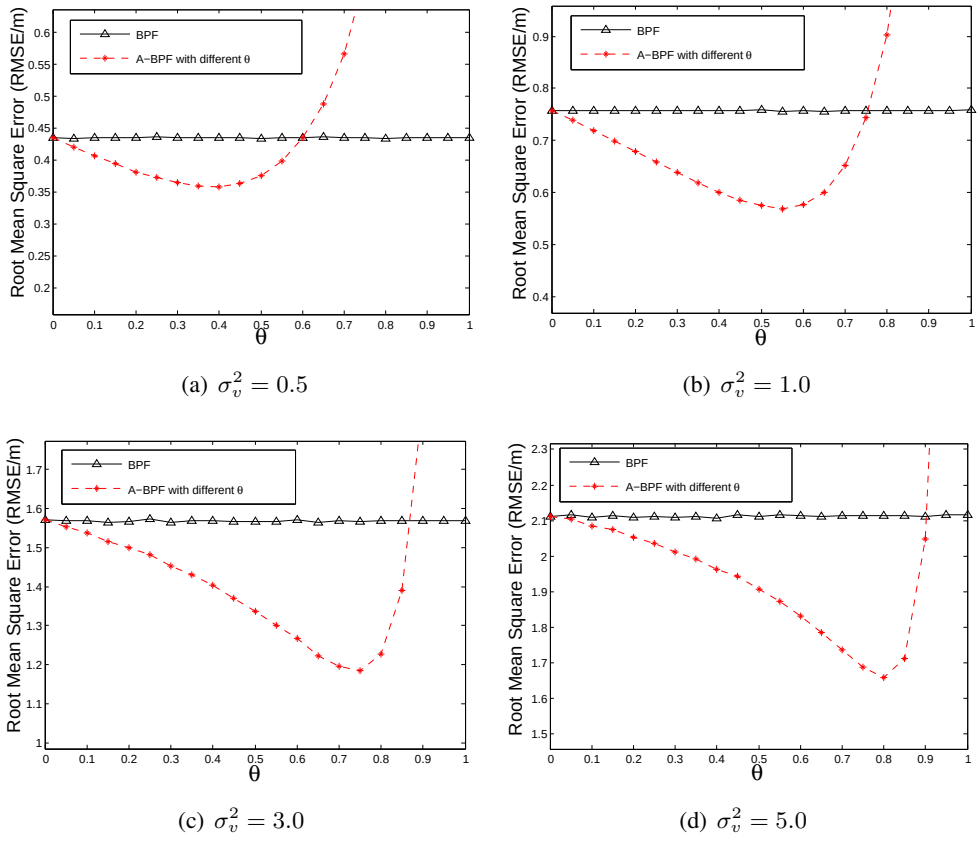
where $\boldsymbol{\theta}$ only depends on the measurement noise and prediction noise in this case.

3.6.3 Optimal θ for A-BPF

First, we compare the estimation accuracy of BPF with our algorithm, A-BPF, to analyze the relationship between $\boldsymbol{\theta}$ and the estimation accuracy. We implement A-BPF and vary $\boldsymbol{\theta}$ from 0 to 1 with a distance step 0.05 and verify whether $\boldsymbol{\theta}$ has an optimal value with minimum estimation error. The algorithms are tested in the different measurement noise scenarios, in which the noise variance σ_v^2 is tuned from 0.5 to 5. The results of our simulation are averaged by 1000 Monte-Carlo trials. In each measurement noise scenario with each $\boldsymbol{\theta}$, 1000 Monte-Carlo simulations are tested. We randomly deploy sensor nodes in every trial, and BPF and A-BPF estimates the target with the same measurement. In this simulation, all algorithms generates 1000 particles at each time step t. Fig. 3.4 illustrates the root mean square error (RMSE) of each algorithm with some different measurement noise scenarios.

In Fig. 3.4, the solid line represents RMSE of BPF and the dash line indicates RMSE of A-BPF with different $\boldsymbol{\theta}$, in which $\boldsymbol{\theta}$ changes from 0 to 1 with an interval of 0.05. When $\boldsymbol{\theta} = 0$, A-BPF totally believes the noisy measurement and has the same accuracy of BPF. RMSE of A-BPF is decreasing when $\boldsymbol{\theta}$ increases from 0, which indicates our adaptation method can improve the measurement likelihood and estimation accuracy. The estimation performance is better than BPF. As indicated in each figure in Fig. 3.4, A-BPF has a minimum RMSE with optimal $\boldsymbol{\theta}$ in different measurement noise scenarios. The minimum RMSE of the A-BPF can reach 1.6m when $\sigma = 5$. When $\boldsymbol{\theta}$ is larger than the optimal value, RMSE of A-BPF begins to increase. And when $\boldsymbol{\theta}$ approaches to 1, it is over tuned and causes high estimation error. Thus, finding a proper $\boldsymbol{\theta}$ is important to the estimation.

When the measurement noise is low, $e.g.$ in Fig. 3.4(a), A-BPF does not improve much even



(a) $\sigma_v^2 = 0.5$ (b) $\sigma_v^2 = 1.0$

(c) $\sigma_v^2 = 3.0$ (d) $\sigma_v^2 = 5.0$

Figure 3.4: Estimation Error Comparison for Different Algorithms with different measurement error variances

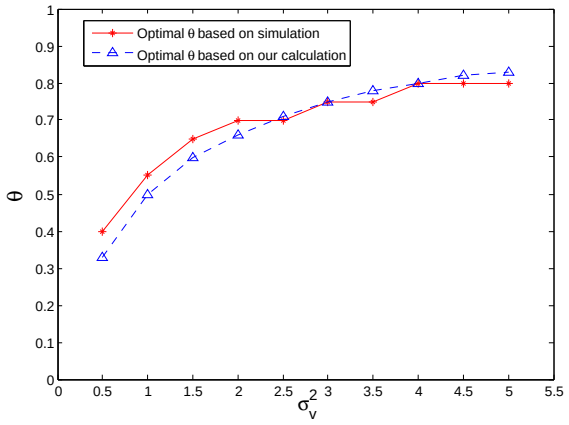


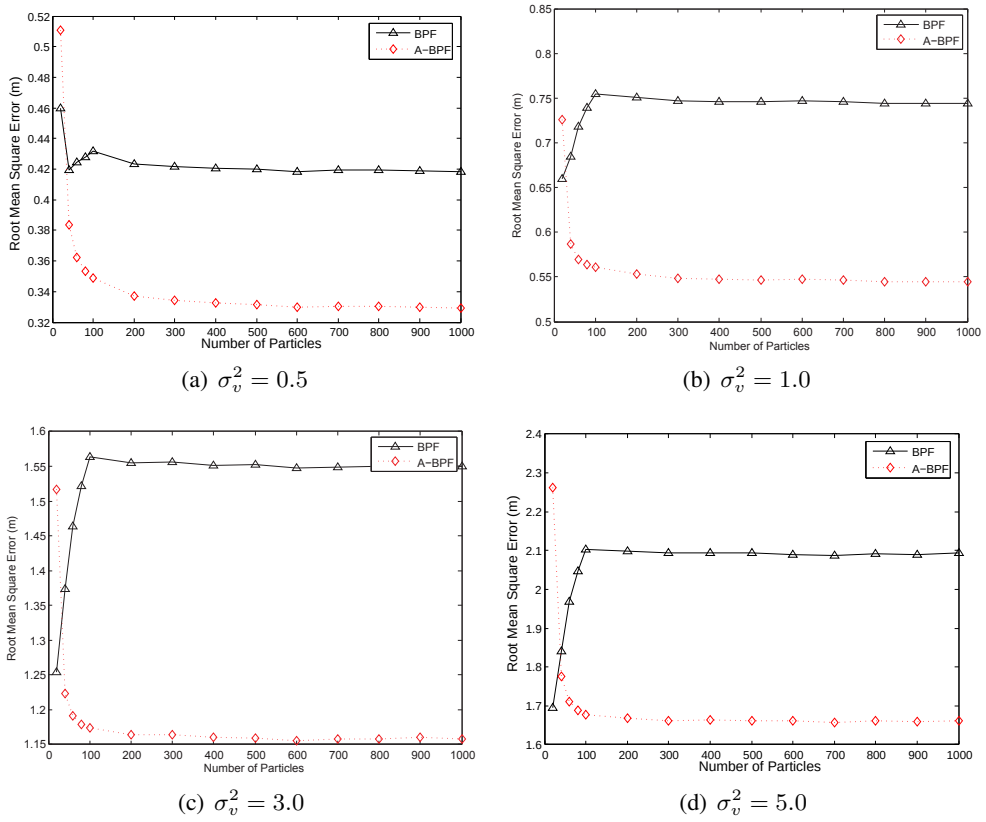
Figure 3.5: Optimal $\boldsymbol{\theta}$ comparison between the simulation results and our calculation based on (3.31)

with the optimal $\boldsymbol{\theta}$. However, when the measurement noise is quite large, $e.g.$ in Fig. 3.4(d), A-BPF can effectively reduce RMSE with the optimal $\boldsymbol{\theta}$. And the gap between A-BPF and BPF is much larger in Fig. 3.4(d) than Fig. 3.4(a). Besides, the optimal $\boldsymbol{\theta}$ increases when the measurement noise rises, $e.g.$ in Fig. 3.4(a), $\boldsymbol{\theta}$ is 0.3 and it becomes 0.8 in Fig. 3.4(d).

The comparison of $\boldsymbol{\theta}$ obtained in simulation and derived in A-BPF is depicted in Fig. 3.5. The solid curve is the optimal $\boldsymbol{\theta}$ in the simulations and the dash curve is the optimal $\boldsymbol{\theta}$ based on (3.31) with different measurement noise. As shown in Fig. 3.5, the optimal $\boldsymbol{\theta}$ based on our calculation in (3.31) is slightly different from the simulation results, because the optimal $\boldsymbol{\theta}$ is derived according to the approximation in (3.16). Since the optimal $\boldsymbol{\theta}$ is approaching the simulation results, it is still suitable for implementation. The estimations based on the optimal calculation and Monte-Carlo simulation are not too different. Thus, using the derived optimal $\boldsymbol{\theta}$ is feasible for A-BPF with high accuracy. Fig. 3.5 also depicts the optimal $\boldsymbol{\theta}$ increases with the rise of measurement noise. It testifies (3.31) that $\boldsymbol{\theta}$ rises when the measurement noise becomes larger. In this case, A-BPF assigns more belief for the predicted measurement for adaptation.

3.6.4 Performance Evaluation

We also compare the estimation performance of PFs with varying the particle numbers. As Crisan $et\ al.$ indicate, the estimation error should converge to 0 with the increasing number of particles [58]. However, this only happens if the error is small, as the solid curve illustrated in Fig. 3.6(a). When the measurement noise begin to rise, increasing the particle number of the BPF can not improve the estimation accuracy. In Fig. 3.6(d), RMSEs of the BPF begin to rise when the number of particles exceed 100. It implies that high measurement error lead significant biased likelihood calculation which degrades the estimation. However, our adaptive method can improve the estimation accuracy, and makes PFs converge to a low RMSE which is shown by



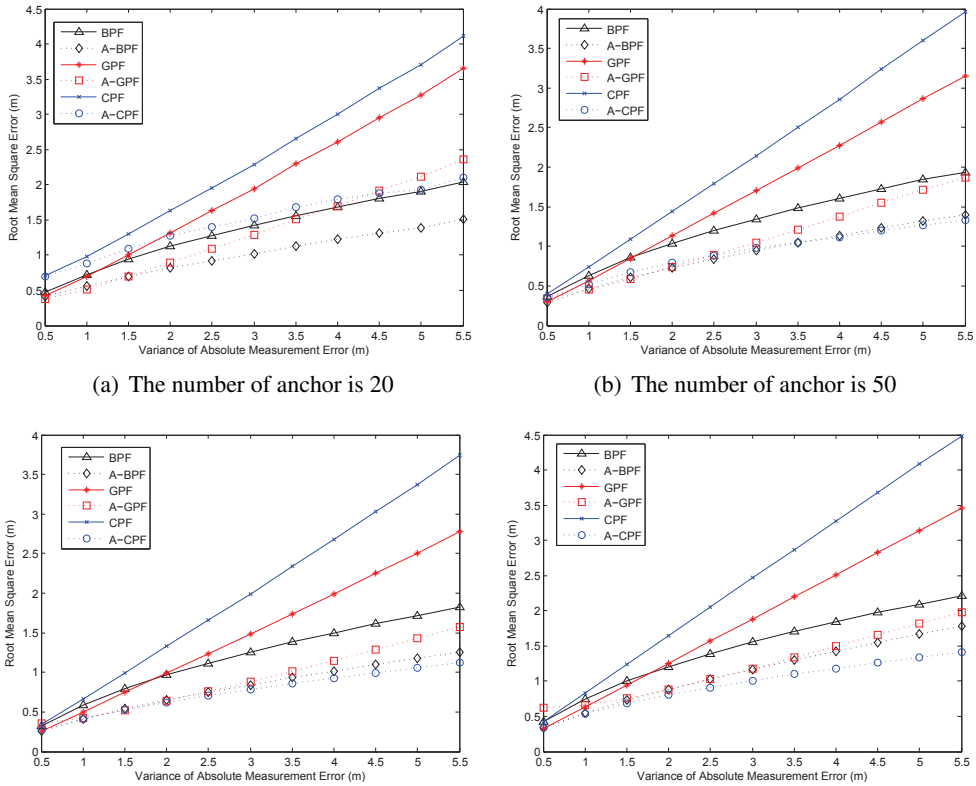
(a) $\sigma_v^2 = 0.5$ (b) $\sigma_v^2 = 1.0$

(c) $\sigma_v^2 = 3.0$ (d) $\sigma_v^2 = 5.0$

Figure 3.6: Root mean square error (RMSE) comparison for different algorithms with different number of particles

the dash curve in Fig. 3.6.

Fig. 3.7 illustrates the RMSE comparison for BPF, A-BPF, GPF, A-GPF, CPF and A-CPF. The measurement noise covariance varies from 0.5 to 5.5, and we use the optimal $\boldsymbol{\theta}$ based on (3.31) as the belief factor in the adaptive PFs. When the measurement error is small, the performance of PFs are similar. When the error increases, the estimation errors of BPF, GPF and CPF rise accordingly. Especially, CPF has the highest RMSE due to the imprecise likelihood calculation, although it has the constraints. It indicates that the measurement noise is the major impact which influences the estimation performance of PFs. The adaptive PFs have better performance. A-BPF and A-GPF have the similar estimation accuracy. RMSE of A-CPF is the lowest. In this simulation, different number of anchors are deployed in each simulation, which shows slight performance changes from less anchors to more anchors. In Fig. 3.7(a), when less anchors are deployed in the playing field, A-BPF and A-CPF have almost the same performance even with the increase of measurement error. When more anchors are deployed, A-CPF outperform A-BPF in which the constraint function plays an important role in the estimation. However, the performance of the algorithms do not change much with multiple anchors. Thus, it is not



(a) The number of anchor is 20

(b) The number of anchor is 50

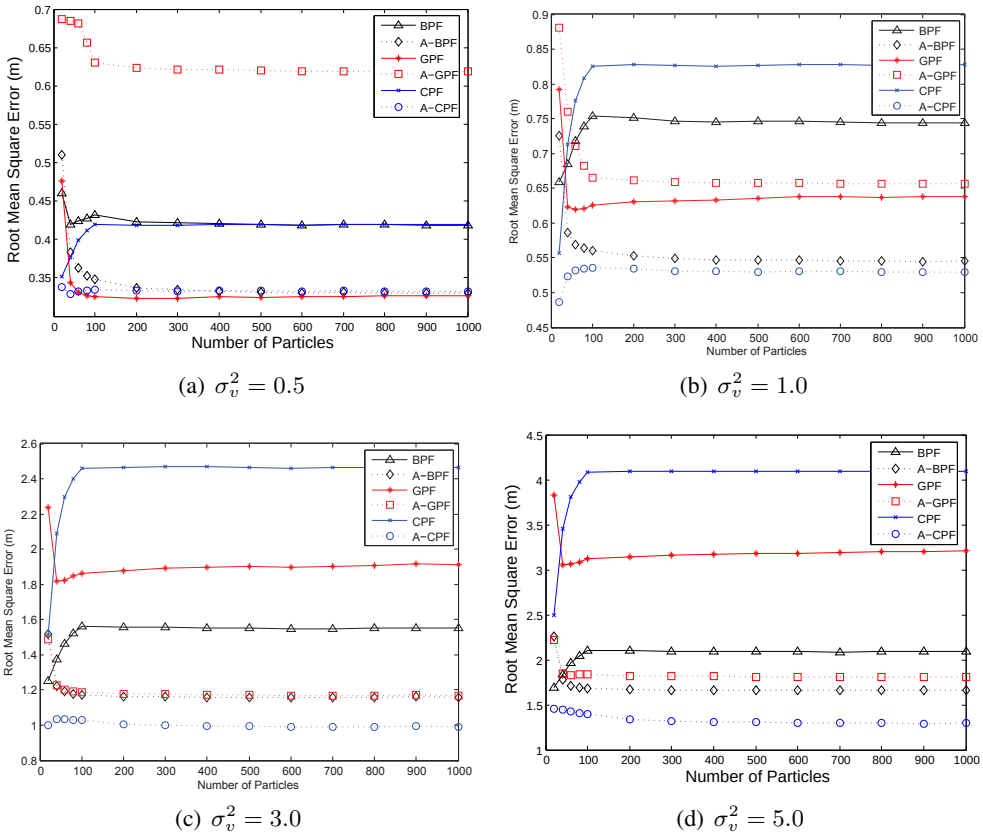
(c) The number of anchor is 80

(d) The number of anchor is 100

Figure 3.7: Root mean square error (RMSE) comparison for different algorithms with different ranging error variances

necessary to deploy anchors densely to obtain a more accurate location.

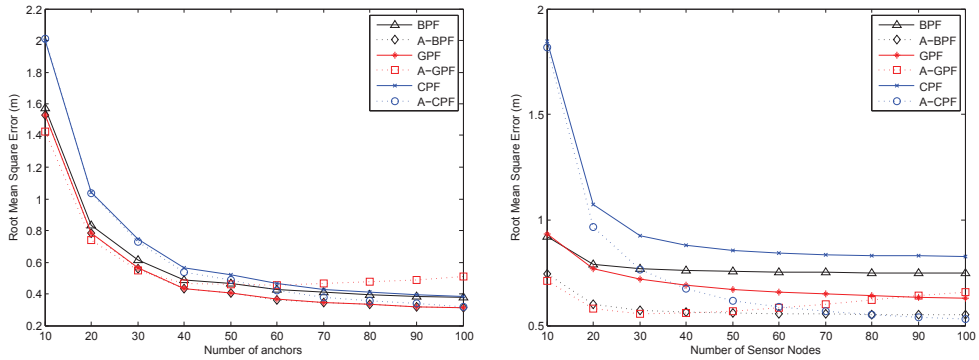
We also compare the estimation performance of PFs with varying the particle numbers. As Crisan *et al.* indicates, the estimation error should converge to 0 with the increasing number of particles [58]. However, simulation results show that the estimation accuracy is corrupt with high measurement noise. When the error is small, as illustrated in Fig. 3.8(a), RMSEs of the PFs can converge to a very low value with the increased particle number except the BPF and the CPF, which illustrates that the measurement noise can influence the convergence of PFs although not much. In this case, our adaptive method does not improve much and A-GPF even has a higher RMSE than the GPF. When the measurement noise begin to rise, increasing the particle number of the original PFs can not improve the estimation accuracy. In Fig. 3.8(d), RMSEs of the BPF and the CPF begin to rise when the number of particles exceed 100 and the GPF does not improve either. It implies that high measurement error leads significant inaccurate likelihood calculation which degrades the estimation. Besides, the CPF outperforms the BPF which is mentioned by Farahmand *et al.* [65] only occurs in the low noise case. However, our



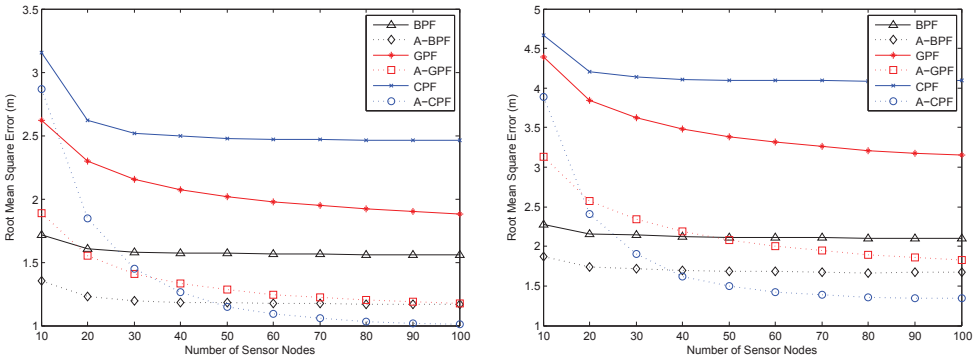
(a) $\sigma_v^2 = 0.5$ (b) $\sigma_v^2 = 1.0$

(c) $\sigma_v^2 = 3.0$ (d) $\sigma_v^2 = 5.0$

Figure 3.8: Root mean square error (RMSE) comparison for different algorithms with different number of particles



(a) $\sigma_v^2 = 0.5$ (b) $\sigma_v^2 = 1.0$



(c) $\sigma_v^2 = 3.0$ (d) $\sigma_v^2 = 5.0$

Figure 3.9: Root mean square error (RMSE) comparison for different algorithms with different number of anchors

adaptive method can improve the estimation accuracy, and make PFs converge to a low RMSE. In Fig. 3.8, RMSEs of 3 adaptive PFs decrease with the rising particle number. RMSEs can converge to a very low value even the measurement error is high. Therefore, our method can reduce the imprecise measurement effect and achieve a better performance.

To obtain a more accurate estimation, anchors are deployed as many as possible, which can provide more information. In Fig. 3.9, the number of anchors is adapted and the RMSEs are depicted accordingly. When the number of anchors is small, the RMSEs of the algorithms are quite high. With the increasing anchors, the RMSEs drop accordingly. In Fig. 3.9(a), the performance of these algorithms are quite similar. However, in Fig. 3.9(d), the performance difference between each algorithm is quite large. For the original PFs, the RMSEs do not change too much with more anchors, due to the imprecise measurement effect. Since all the measurements contain error, original PFs can not obtain a more accurate estimation. However, with our adaptation method, the measurements are more reliable and the estimation performance is improved with increasing the number of anchors.

3.7 Summary

In this chapter, the measurement error impact for likelihood calculation in PF is analyzed and it is found that the measurement likelihoods for particles are influenced by the large instantaneous errors at a given time, which is the major source of estimation error. Therefore, the estimation accuracy of PF in wireless tracking systems can be further improved by reducing the measurement noise. Based on the analysis, an adaptation method is proposed by introducing the predicted measurement and its belief factor $\boldsymbol{\theta}$. The optimal $\boldsymbol{\theta}$ is derived and implemented into our proposed adaptive algorithms, the A-BPF, the A-GPF and the A-CPF. In the simulation, it is observed that some analytical conclusion for PF is not suitable to the high measurement scenarios, and we verify that the adaptive PFs improve the estimation accuracy with different measurement noise environments and the optimal $\boldsymbol{\theta}$ derived in our method approaches to the actual value.

The proposed adaptation method is a general solution for PFs based on the general analysis. Thus, it can be implemented in many applications. Since this work is for the indoor localization, the proposed adaptive PFs are examined in the indoor target tracking systems. The simulation results indicate that, comparing with the original PFs, our algorithms can effectively reduce the estimation error which can achieve nearly 1m better than the original PFs. Besides, the A-CPF is more accurate for target tracking than other filters, which is below 1.5m.

The adaptive PF is a general solution which is suitable for many wireless ranging techniques. The optimal $\boldsymbol{\theta}$ is derived based on the Gaussian assumption, in which the measurement error follows normal distribution. However, the error for indoor localization systems does not always follow Gaussian distribution. Thus, the adaptive PFs are not always suitable for other environments. Therefore, in the next chapter, how to model the indoor ranging error and extend our adaptive PFs to the indoor environment are discussed.

Chapter 4

Dynamic Gaussian Model

4.1 Introduction

The major challenge of nonlinear filters for indoor localization is the measurement error. Due to the complicated infrastructure and hybrid NLOS/LOS transmission channel, the measurement error is quite high and makes the estimation unreliable [1]. For conventional outdoor environment, the error is modeled as a static normal or Gaussian distributions. However, experimental results and analysis indicate that Gaussian distributions are not sufficient to describe the error model in the indoor environment. Thus Kalman filter or PF based on Gaussian distributions can not estimate the position accurately. Some references model the noise as Exponential, Rayleigh, Weibull or Gamma distribution [71, 119, 120]. However, these models are suitable for some typical static position. The parameters and type of distributions can be quite different in different positions due to the complicated infrastructure of indoor environment.

In the previous chapter, the adaptive PFs are proposed and work well in the Gaussian distribution. However, the non-Gaussian cases are not mentioned. The ranging error in the indoor localization system does not always follow Gaussian distribution. Therefore, in this chapter, a dynamic Gaussian modeling (DGM) method is proposed to describe the measurement error for indoor environment. First, a general Gaussian distribution model is constructed to describe the ranging error based on histograms. For a typical scenario, the instantaneous ranging error from LOS or NLOS is considered as the drift from the general Gaussian distribution. Based on DGM model, the hybrid LOS/NLOS indoor ranging error, can be model dynamically in a uniform framework. Although it is not an accurate model, it is suitable and adaptable for non-linear filters based on Gaussian assumption.

Since DGM involves instantaneous error at each time, the likelihood adaptation method is applied in the non-Gaussian case based on the Gaussian model. The adaptive PFs integrated with the DGM are evaluated in the simulation with multiple distribution cases. The results illustrate

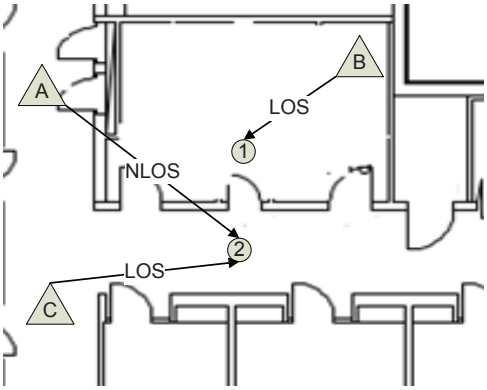


Figure 4.1: The Hybrid LOS/NLOS Indoor Environment.

that the adaptive PFs outperform the conventional PFs based on a typical error distribution and the DGM is suitable for the complex environment.

4.2 Range-based Measurement Error Modeling

4.2.1 Problems of Ranging Error Modeling

Besides the asynchronous clock, range-based indoor positioning system is mainly influenced by the multipath transmission effect and NLOS effect. Due to the complicated environment with hybrid LOS and NLOS transmission channel, using the current general distribution models to describe the ranging error is not accurate. And the traditional Gaussian assumption is not suitable for indoor environment.

Some references model it as Exponential, Rayleigh, Weibull or Gamma distribution [71, 119, 120]. The problem of modeling NLOS error is that the modeling assumes that the wireless propagation channel is static with a fixed position. However, the measurement for the moving target has various scenarios. As illustrated in Fig. 4.1, the target is in the room, which is position 1, with an AP, C, deployed in the corridor, the transmission is NLOS, but when the target moves out, at position 2, the transmission turns into LOS for C. In this case, using a unified exact distribution can not model both LOS and NLOS scenarios, which leads to the inaccurate estimation for filtering methods. Even the change of infrastructure can influence the propagation channel, e.g. when a new obstacle arrives and affects the wireless propagation, the distribution model is probably changed.

NLOS identification and mitigation is another solution. One of identification is based on the building layout, but it requires the prior mapping information of the infrastructure [121]. Another method employs statistic results of the NLOS ranging measurements [122] or compare with LOS general model [123]. The positioning algorithms either set typical constraints or just miti-

gate NLOS measurements to derive the location [119, 124]. NLOS identification methods rely on the statistics or serials of samples, which is not suitable for the instantaneous measurement with a single value. Besides, when all measurements are NLOS and mitigated, it is hard for positioning algorithms to obtain an accurate estimation.

Hybrid model describe LOS and NLOS separately by using Gaussian distribution and Gamma distribution. And the error distribution is the hybrid Gaussian and Gamma distribution with a tuning probability [125]. This method fully considers multiple situations for indoor positioning. Similar method such as fingerprinting calculates almost all the possible positions to get full picture of error distribution [126]. Both of the modeling methods are complicated and not robust to environment changes.

4.3 Error Distribution Fitting

Since there is no common distribution for the TOA ranging error based on theoretical analysis, multiple distributions are applied for the error fitting to check which one is most suitable for the real indoor environment. Here, Gaussian distribution, exponential distribution, Gamma distribution, Rayleigh distribution, log-normal distribution and Weibull distribution are used for fitting.

4.3.1 Gaussian Distribution

Gaussian distribution, which is also denoted as normal distribution, is a very common continuous probability distribution. The two parameters of the Gaussian distribution, the expectation and the covariance, are also the important parameters of the statistic modeling. It is often used when the distribution is unknown.

The PDF of the Gaussian distribution, $\mathcal{N}(\mu, \sigma^2)$, is expressed as

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp(\frac{(x-\mu)^2}{2\sigma^2}) \tag{4.1}$$

where μ indicates the expectation and σ^2 is the variance of x.

The Gaussian distribution is suitable for modeling the ranging error for TOA or RSS in outdoor scenarios. In outdoor localization systems, the wireless transmission channel suffers less multi-path effect and less interferences, thus, the error is mainly from the inaccurate measurement of the system. Then the Gaussian distribution is applied precisely. However, the environment of the indoor localization system is much more complicated. The Gaussian distribution can not model the ranging error well in such scenario. Some researchers use the Gaussian distribution with a large positive expectation to indicate the TOA ranging model, however, the histogram indicates that the positive bias of the ranging error is not so large and such model still can not describe the error well.

Although the Gaussian distribution is not well suited for the indoor ranging error, the optimization based on the Gaussian assumption can make the calculation simple. Thus, the Gaussian assumption is widely used in the optimization problems, such as NLLS. In this chapter, the Gaussian assumption is employed to approximate other distributions.

4.3.2 Exponential Distribution

The exponential distribution is a probability distribution which describes the time between events in a Poisson process. It has the key property of being memoryless. The PDF of the exponential distribution is formulated as:

$$p(x) = \begin{cases} \lambda \exp(-\lambda x) & x > 0 \\ 0 & otherwise \end{cases} \tag{4.2}$$

where λ is the parameter of the exponential distribution, which is often called rate parameter.

For the TOA ranging, if the clocks in the anchors and the target are highly synchronized, the arrived data is formulated as the Poisson process. In this case, the ranging error is modeled as the exponential distribution. However, the exponential distribution does not consider the transmission behavior in the indoor environment, such as multi-path effect. Besides, the clocks are not synchronized and the error suffers a positive bias due to the NLOS ranging.

4.3.3 Gamma Distribution

The Gamma distribution, $\Gamma(\alpha, \beta)(x)$, is a two parameters probability distribution, which is frequently applied to model the waiting time. In addition, it is also used for the Bayesian statistics. The PDF of the Gamma distribution is formulated as:

$$p(x) = \begin{cases} \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} \exp(-\beta x) & x > 0 \\ 0 & others \end{cases} \tag{4.3}$$

where $\Gamma(\alpha)$ indicates the Gamma function; α is the shape parameter and β is the rate parameter. Both of the two parameters are positive. Besides, if α is an integer, then it turns to be Erlang distribution:

$$\Gamma(\alpha) = (\alpha - 1)! \tag{4.4}$$

And if $\alpha = 1$, the Gamma distribution turns into the exponential distribution.

The Gamma distribution is well suitable for the indoor error modeling [71, 119, 120]. It has a small positive bias with a long tail to indicate the NLOS ranging measurement. It can also model the LOS ranging. Thus, in a hybrid environment, the Gamma distribution is a promising model for the ranging error. However, it still can not model the negative error caused by the system.

4.3.4 Rayleigh Distribution

The Rayleigh distribution is also an one parameter continuous distribution. It is formulated as:

$$p(x) = \begin{cases} \frac{x}{\lambda^2} \exp(-\frac{x^2}{2\lambda^2}) & x > 0 \\ 0 & others \end{cases} \tag{4.5}$$

where λ is the scale parameter.

The Rayleigh distribution is commonly used for modeling the wireless transmission channels, where the multi-path effect is the major impact. The ranging error can be modelled when the TOA signal is propagated through the Rayleigh channel. However, the indoor environment is more complicated than the Rayleigh channel, where the shadowing effect is also an important factor.

In addition to the common used models which are mentioned above, another two distributions are also involved in the error modeling, which are the log-normal distribution and the Weibull distribution.

4.3.5 Log-normal Distribution

The log-normal distribution is the continuous distribution of a random variable whose logarithm is normally distributed. A random variable of such distribution is the positive real value:

$$p(x) = \begin{cases} \frac{1}{\sqrt{2\pi\sigma^2 x^2}} \exp(-\frac{\ln x - \mu}{2\sigma^2}) & x > 0 \\ 0 & others \end{cases} \tag{4.6}$$

4.3.6 Weibull Distribution

The PDF of the Weibull distribution is formulated as:

$$p(x) = \begin{cases} \frac{\beta}{\eta}(\frac{x}{\eta})^{\beta-1} \exp(-(\frac{x}{\eta})^{\beta}) & x > 0 \\ 0 & others \end{cases} \tag{4.7}$$

where $\beta > 0$ is the shape parameter while $\eta > 0$ is the scale parameter of the distribution. The Weibull distribution interpolates between the exponential distribution where $\beta = 1$ and the Rayleigh distribution where $\beta = 2$.

4.3.7 Modeling for Real Indoor Ranging Error

The TOA ranging is examined in multi scenarios via a robot moving around the room or the corridor with several deployed anchors. The robot attempts to move through every possible position in the playing field, and the ranging measurements are tested several times for a single position, thus, the ranging error obtained in this experiment can represents all the possible random values.

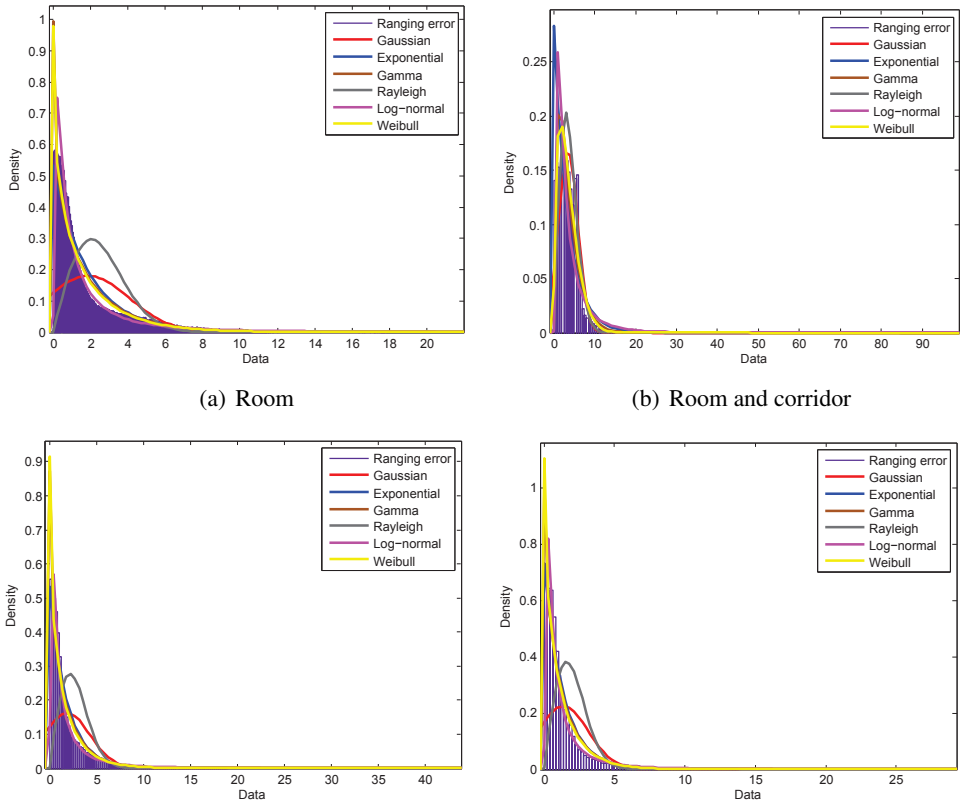
Then, the ranging error is collected by comparing the measurements with the actual ranging values. In this experiment, our goal is to model the wireless propagation channel and its impact for the TOA ranging. Therefore, the negative error which are caused by the asynchronous clock in the sensor nodes are discarded. Only the positive error are left. The error is modelled by the 6 distributions mentioned above. The parameters for the different distributions are derived based on the fitting tool in Matlab which employs EM method.

In the first scenario, 9 anchors are deployed in a room and the robot carrying the target moves through the whole room. In the second scenario, the playing field is two rooms and a corridor, which is a hybrid of LOS/NLOS measurement. In the last two scenarios, the robot moves along the same corridor. However, the experiments are evaluated on both of the left side and the right side separately to check whether the error follows the same distribution. The data are collected from the static positions and the moving trajectories. Then, it almost represents all the possible situations.

The error fitting results are depicted in Fig. 4.2. It is clearly observed that the Gaussian distribution is not suitable for modelling such error, which has different shape to the histogram. In addition, the shapes of the Gaussian distribution and the Rayleigh distribution are not quite similar to the statistic histograms. Although the Exponential distribution can also model the positive error, the trend of exponential distribution is decreasing which still can not represent the histogram well. For Gamma, log-normal and Weibull distributions, the shapes are close to the histogram for each scenario. However, the shapes are different for a single distribution in different scenarios.

The fitting parameters are listed in Table 4.1. As it indicates, the parameters for different scenarios are quite different. Even for the same corridor, the parameters for the left side are different from the ones for the right side. Thus, it is not easy to obtain a unique model to describe the error model. Even for the same distribution, the characters are not the same for different scenarios. Take the Gamma distribution for instance, the parameter α of the Gamma distribution in the room or corridor is smaller than 1, which makes the Gamma distribution approaching the exponential distribution. While, in the hybrid scenario where the robot moves through both of the room and the corridor, α is bigger than 1, which is a typical Gamma distribution.

Compare with the distribution fitting residual error, the root mean square error (RMSE) of the Gaussian distribution is 0.0469 and the mean residual error is 0.0013, which indicates that the fitting is close to the histogram. For other distributions, the RMSE and mean residual error are all higher than Gaussian distribution. Take Gamma distribution and log-normal distribution for instance, Gamma distribution and log-normal distribution suppose to best fit the indoor ranging for localization systems [121, 125]. However, the RMSE of Gamma distribution is 0.0839 and the RMSE of log-normal distribution is 0.0763 in the hybrid room and corridor scenario. Based on the residual error, Gaussian distribution should be the best fitting model. However, when



(a) Room

(b) Room and corridor

(c) Corridor (left side)

(d) Corridor (right side)

Figure 4.2: Indoor ranging error fitting

Table 4.1: Ranging error fitting

Position	Room	Room and Corridor	Corridor (L)	Corridor (R)
Gaussian	$\mu = 1.86777$ $\sigma = 2.20838$	$\mu = 3.49696$ $\sigma = 2.3588$	$\mu = 1.85887$ $\sigma = 2.47477$	$\mu = 1.35467$ $\sigma = 1.77633$
Exponential	$\lambda = 1.86777$	$\lambda = 3.49696$	$\lambda = 1.85887$	$\lambda = 1.35467$
Gamma	$\alpha = 0.844764$ $\beta = 2.21099$	$\alpha = 1.657$ $\beta = 2.11042$	$\alpha = 0.852322$ $\beta = 2.18095$	$\alpha = 0.908704$ $\beta = 1.49077$
Rayleigh	$\lambda = 2.04517$	$\lambda = 2.98266$	$\lambda = 2.18859$	$\lambda = 1.57962$
Log-normal	$\mu = -0.0728542$ $\sigma = 1.34746$	$\mu = 0.920752$ $\sigma = 1.01179$	$\mu = -0.0706542$ $\sigma = 1.31882$	$\mu = -0.339022$ $\sigma = 1.2679$
Weibull	$\eta = 1.74622$ $\beta = 0.879836$	$\eta = 3.83069$ $\beta = 1.44507$	$\eta = 1.728$ $\beta = 0.875843$	$\eta = 1.28658$ $\beta = 0.907503$

consider the real experiment performance, Gamma distribution is close to the real case [127]. In the following chapters, we use Gaussian distribution to indicate the outdoor environment and Gamma distribution to indicate the indoor environment as the general ranging error model for PFs.

4.4 Dynamic Gaussian Approximation

In this chapter, we use a dynamic Gaussian model (DGM) to approximate the ranging error, which makes the error distribution suitable for Gaussian-based non-linear filters. Therefore, firstly, a general Gaussian model is obtained to describe the ranging error. The Gaussian distribution function is a symmetrical function, however, the TOA ranging error is always positive and not symmetrical. Thus, a real Gaussian distribution function can not be used to represent the error distribution. Then, a two-stage modeling is introduced.

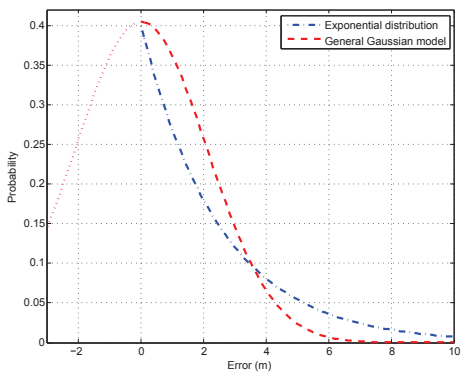
4.4.1 General Gaussian Modeling

Firstly, a general Gaussian model is proposed to describe the statistical histogram of the error distributions. The general Gaussian model is to use the part of the shape of the Gaussian distribution function to represent the distribution. The negative values of the Gaussian distribution is discarded. The expectation of the constructed Gaussian distribution is the mode of the error distribution, and the standard deviation indicates the shape of the error distribution. Unlike Gaussian distribution fitting method, which uses the mean and standard deviation of the errors to construct a Gaussian distribution, this method does not attempt to construct a real distribution but rather a function to approximate the error.

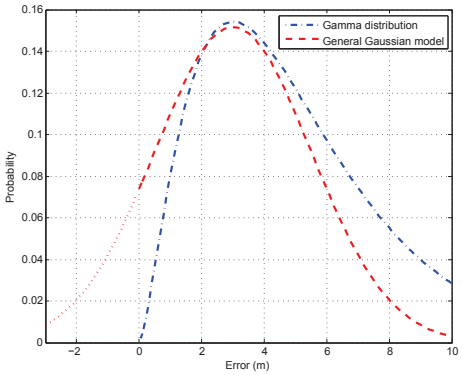
How to use general Gaussian model to describe other distributions is illustrated in Fig. 4.3. The mean value for the general Gaussian model represents the mode of the distribution function. The standard deviation attempts to tune the shape of Gaussian function to fit the distribution. For the statistical histogram, which is fitting with any typical distribution in Fig. 4.3(f), the general Gaussian model attempts to cover all the values with a proper shape of Gaussian function.

4.4.2 Dynamic Gaussian Modeling

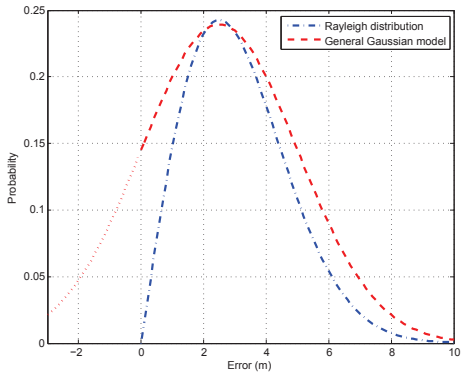
The parameters of Gaussian model are obtained from the statistical histograms of extensive ranging experiments in the indoor environment. As illustrated in Fig. 4.4, the general Gaussian model is derived from the statistical histograms. No additional information is required and the statistical histograms are not restrict to LOS or NLOS. To make it suitable to the multiple positions, the measurement error, no matter of LOS or NLOS, is considered as an instantaneous error at each time. Then the model based on the instantaneous error at this time is considered as the drift from the generic Gaussian model accordingly. As illustrated in Fig. 4.4, the dash curve represents the generic Gaussian model for ranging error. The star marks an instantaneous



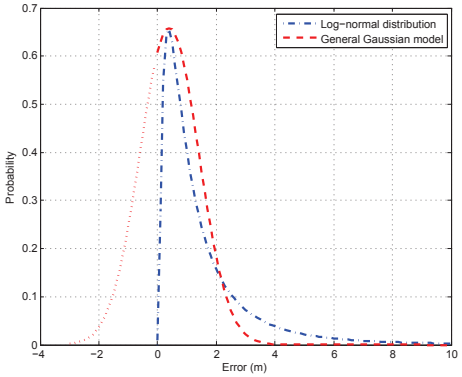
(a) Exponential distribution



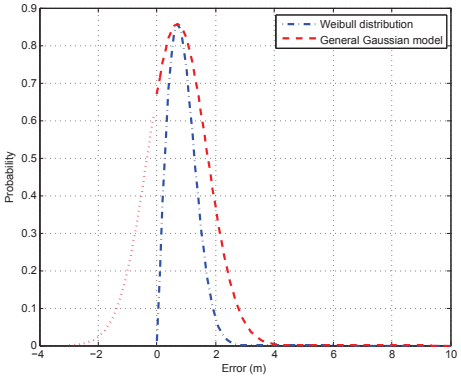
(b) Gamma distribution



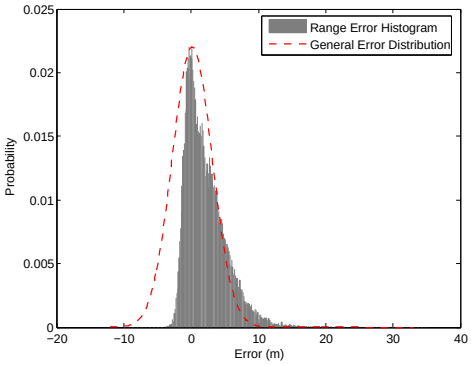
(c) Rayleigh distribution



(d) Log-normal distribution

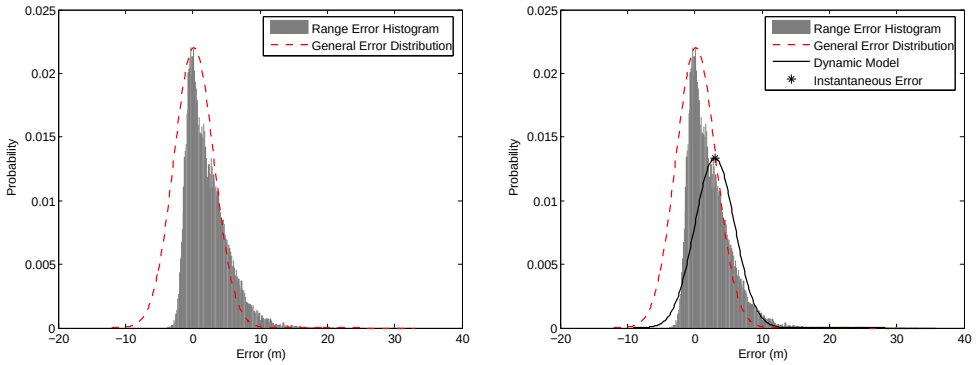


(e) Weibull distribution



(f) Statistical Error histogram

Figure 4.3: General Gaussian model fitting for other distributions and statistical histogram.



(a) General Gaussian distribution from histogram (b) Dynamic Gaussian model: drift from general Gaussian distribution

Figure 4.4: Dynamic Gaussian Modeling for Indoor Ranging Error.

error at a typical time. Then, the expectation of the error distribution moves to the position of star from the general Gaussian model with the same standard deviation, just as the solid curve depicted. Thus, when the instantaneous measurement is available, the error model is biased by the instantaneous error. In this case, the error distribution is dynamically modeled based on the Gaussian model. When the instantaneous error is small, there is no big difference between general model and biased model, just as the outdoor scenario. However, when the instantaneous error is quite large, the two models are significantly different, but we can still model it using the drifted Gaussian model. Thus, DGM is typically constructed for indoor environment.

Although the DGM is not an accurate error distribution for indoor positioning system, it is suitable for nonparametric nonlinear filters in the dynamic environment. In Gaussian nonlinear filters, the error distribution is assumed as a fixed Gaussian distribution which is not practical in indoor environment. However, DGM indicates that the instantaneous error is still within the Gaussian distribution framework. Therefore, nonlinear filters are implemented easily without considering an accurate error model. Besides, DGM illustrates how the error influences the estimation. Because the instantaneous error is involved in the distribution related calculation.

4.5 Applications for Nonlinear Filters

The likelihood adaptation method mentioned in the previous chapter is based on the real Gaussian distribution. The optimal θ is derived based on the minimum KLD, which is also based on the Gaussian assumption. Since it is a promising solution for the target tracking problems, the DGM can help the likelihood adaptation method extend to the non-Gaussian case.

As indicated before, the ranging error can be modeled as a general Gaussian model, which can be expressed as $v_t^j \sim \mathcal{N}(\mu_t^{j^*}, R_t^{j^*})$, where $\mu_t^{j^*}$ is not the mean value of the error. Then,

each instantaneous error is the drift of the Gaussian distribution. Thus, the adaptive PFs can be modified based on the DGM.

For BPF, almost all the parts of the A-BPF are still the same, except the likelihood function and the optimal θ. The likelihood function is based on the general Gaussian model, while the optimal θ is:

$$\theta_t^j = \frac{R_t^{j*}}{\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial \boldsymbol{x}_t}\boldsymbol{Q}_t[\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial \boldsymbol{x}_t}]^T + R_t^{j*}} \tag{4.8}$$

where R_t^{j*} is obtained based on the error distribution approximation. If the approximation does not fit the error well, the estimation is degraded accordingly.

The procedure of A-GPF also does not change much. However, since the Gaussian distribution approximation is also based on the DGM, the accuracy of the general Gaussian model still influence the estimation performance of A-GPF.

The major advantage of A-CPF is that the constraint conditions are robust to the measurement noise, no matter of what distribution it follows. Thus, even without an accurate general Gaussian approximation model, the performance is still good. In this case, the A-CPF based on the DGM outperforms than the other adaptive PFs.

However, DGM is still not suitable for the Kalman filtering algorithms. Because Kalman filters assume the distribution is zero-mean normal distribution, and DGM always contain a bias in both general model and dynamic model.

4.6 Summary

In this chapter, the problems of the indoor ranging error modeling is discussed. It is difficult to find a unique general probability distribution to describe the hybrid LOS/NLOS ranging error in the indoor environment according to the target moving behavior and the distribution fitting results. Thus, a dynamic Gaussian model is proposed to represent the error model. Based on that model, the adaptive PFs can be extended to the non-Gaussian cases. For the real applications, the adaptive PFs based on DGM will be test on the real experiment platforms.

Chapter 5

Real System Evaluation

5.1 Introduction

In the previous chapters, the proposed algorithms are evaluated in the simulation. In the real application, the problems are more complicated than the simulation. The real indoor target tracking system does not only consider the implementation of the tracking algorithms, but also consider the hardware, the relative information, the communication protocol and robustness of the software.

In this chapter, a real-time indoor target tracking system is introduced. The system is implemented with a robot and several sensor nodes. The communication technique is based on the time-of-flight (TOF) method. The procedure of TOF and the architecture of the system are illustrated. Based on the architecture, several components are developed to assist the localization algorithms obtain an accurate estimation. In addition to the location estimation, the metrics of evaluating the estimation performance is also illustrated.

Since the real application also consider the relative information of the surroundings. The building layout information can be used as the constraint conditions for the PFs. In this chapter, a new PF which fuses the building layout information and considers the target motion behavior is proposed, which is named context-aware particle filter (CA-PF). With more information, the performance is further improved.

The whole system is evaluated in many scenarios. Several PFs and popular used localization algorithms are examined with numerous metrics. The results indicates that CA-PF outperforms the other algorithms.

5.2 Time-Of-Flight based Wireless Sensor Network

Wireless sensor network is used for localization due to it simplicity and fast deployment. The target and anchors are all the equipments of sensor nodes. In our system, the range measurement is based on TOF range measurement [128]. It measures the time delay when a wireless packet is transmitted between the mobile node and an anchor node. The distance between the mobile node and an anchor is known by multiplying the speed of microwave with the time delay. The time delay can mainly estimated according to round trip time (RTT) of a packet. The mobile node begins to sense the TOF when it sends a data packet to an anchor. There is a propagation delay T_{td1} when the anchor receives the data packet. The time of processing the data packet within the hardware of a sensor node is called processing delay T_{pd}, which is assumed as a constant value. After processing, the anchor sends the ACK back to the mobile node with another propagation delay T_{td2}. The RTT T_{RTT} is the sum of T_{td1}, T_{pd} and T_{td2}. The whole procedure is illustrated in Fig. 5.1. Therefore, TOF can be calculated as:

$$T_{TOF} = \frac{T_{td1} + T_{td2}}{2} = \frac{T_{RTT} - T_{pd}}{2}. \tag{5.1}$$

The measurement for each anchor is formulated as:

$$z_t^j = c \times T_{TOF} = \sqrt{(p_t^X - a_j^X)^2 + (p_t^Y - a_j^Y)^2} + v_t^j \tag{5.2}$$

where z_t^j denotes the measurement for jth anchor; $\boldsymbol{x}_t = [p_t^X, p_t^Y]^T$ is the target's coordinates; $[a_j^X, a_j^Y]^T$ denotes the anchor's position; v_t^j is the measurement noise.

To make TOF more precise, the anchor will follow the same procedure as the mobile node, which sends the data packet, receives the ACK as shown in Fig. 5.1 and calculates the RTT and TOF according to equation (5.1). Then the anchor sends the result back to mobile node. Thus, the final TOF is the average value of these two TOF results.

The key point of TOF technique is the sensor's ability of estimating time accurately. The indoor wireless propagation channel influences the data packet propagation, which will further influence the propagation delay and distance measurement. Due to reflection, refraction and scattering of microwaves by numerous reflecting surfaces inside a building, the data packet often arrives at the receiver through more than one path, which is defined as multi path effects [5, 129]. Multi path effects degrade the performance of wireless sensor network, and the performance is further degraded by noise and other wireless interference.

Besides, moving objects make the wireless indoor environments not stationary even the mobile node and anchors are fixed [129]. Even a small change can disturb the indoor wireless channel's character. The variation occurs in bursts, which severely increase the propagation delay and longer measurement distance than expectation. Sometimes, it even leads to link failures.

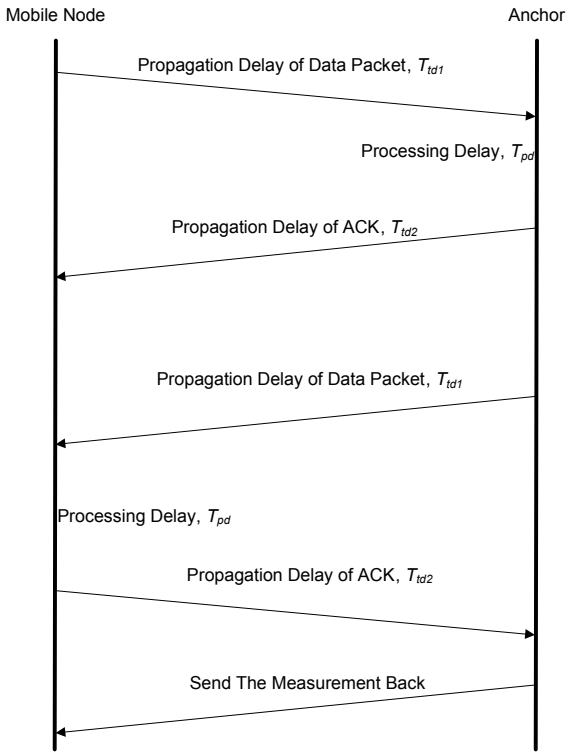


Figure 5.1: TOF working procedure

Finally, the TOF ranging method requires a robust synchronization scheme to make the time estimation reliable.

5.3 Indoor Target Tracking System

As indicated before, the topology of the tracking network system is simple, which consists of the mobile node as the target, the anchors and the fusion center. The packets are transmitted between the target and anchors. Then the ranging values are forwarded to the fusion center. The final estimation is calculated in the fusion center using some typical tracking algorithm.

Because of the high measurement noise and the dynamic environment, the tracking system often encounters some unexpected error and unreliable measurement, such as communication outage and extreme unrealistic ranging value. Then, a single tracking algorithm can not solve all the localization problem for the real world. Thus, other schemes are required to assist the tracking algorithm to obtain an accurate estimation.

The system architecture is depicted in Fig. 5.2. Besides the target tracking algorithm, there are four additional component in the system: initialization, preprocessing, anchor selection and performance evaluation. Initialization constructs the basic information of the system. Prepro-

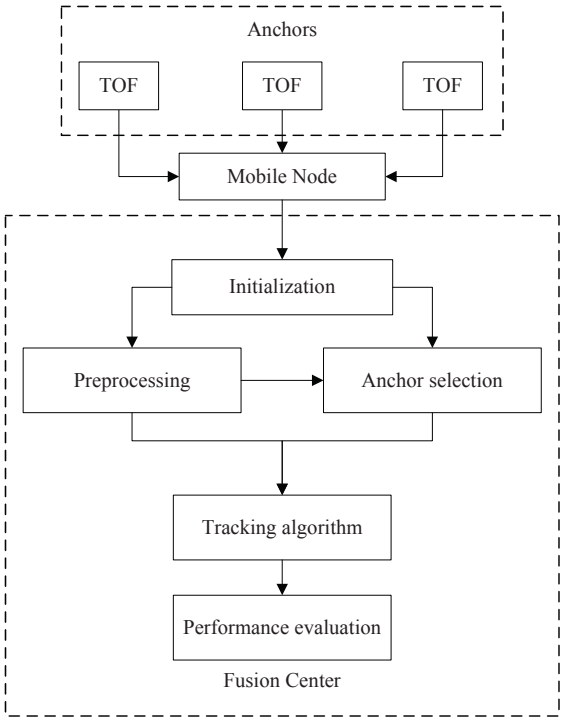


Figure 5.2: System Architecture

cessing and anchor selection reduce the ranging error before using the target tracking algorithm. The last component is to evaluate the whole estimation performance for the experiment.

5.3.1 Initialization

For the real indoor positioning applications, the location estimation is based on the typical building or infrastructure. Thus, the environment information should be initialized. A building map is imported into the system. Besides, the deployed anchors with their actual position are also marked on the map and the according coordinates of the anchors are stored in the system. For graphic interface of the tracking application, the building map with anchors is presented in the software. Then, the estimated trajectory is depicted on the map.

Besides the map and the anchor information, other information can also be applied during the initialization. The building layout, e.g., the coordinates of the rooms, the size of the hallway or the positions of the large obstacles, can influence the estimation. Besides, other information, e.g., the number of targets, walking people and the source of interference, which may interrupt the communication, is also important for the position estimation. Thus, for the tracking system, it is benefit to import all the possible information and the system can adapt the estimation based on these information

5.3.2 Preprocessing

Preprocessing attempts to reduce the ranging measurement error when the anchors obtain the TOF ranging. Due to the high noise and fluctuation of environment, the original ranging measurement contains much error which can lead to high estimation error. Thus, preprocessing calibrates the ranging value during the TOF.

Besides, there is a sampling period for each anchor. Within this period, the ranging values are the same although the system can read the memory several times. Then, the duplicate values is useless for the estimation and it is also a waste of energy. Thus, another task for preprocessing is to distinguish the duplicated value and the new ranging value.

Two algorithms are applied in the preprocessing: the first one is the median filter and the second one is the average filter. For each anchor, the ranging measurement is 1 dimensional vector. Thus, the filters are implemented in a distributed way for each anchor. The ranging data is processed by median filter and the average filter and then forwarded to the next component for further estimation.

Median filter is an effective nonlinear digital filtering technique, which is used to reduce the signal noise. It is widely implemented in the image processing systems. The main idea of the median filter is to output the signal entry by entry. For each entry of the signal, the median filter chooses the median value of the neighbors to replace the signal. The length of the neighbors is named window. The window moves from the beginning of the signal vector to the end step by step. Then, the median value within the window is outputted. When the window is large, the complexity of the algorithm is high accordingly. In addition, if the window has an odd number of signals, it is easy to obtain a median value. However, if the number of signals is even, it is possible to have two median values. Implementing the median filter for each anchor can reduce the extreme ranging value and make the sequential measurement smooth.

Average filter is a calibration method to obtain an average value of the ranging measurement. It also has an window, however, there is only one value as the output. All the signals within the window are averaged. Since the ranging measurement contains the duplicated measurement and the outage information, the length of the window largely affects the output value. If the length of the window is small, which can not cover the sampling period, the average ranging values still has copies which can not improve the estimation performance. If the length of the window is large, the target may possible move a long distance during such period. Then, the system will loss the tracking information within the sampling period. In addition, the outage information will be dropped if there are still ranging values within the window. However, if the window contains all the outage information except ranging value, the output of the average filter is the outage information, which means that the anchor can not communicate with the target.

Both of the median filter and the average filter can reduce the noise and the outage information. However, the median filter can not distinguish the state of the communication, which means

whether the outage information is real or just a fluctuation of the environment for a short period. The average filter will loss some ranging values if the window is too large. Thus, it is better to implement both of the two algorithms. The ranging data is processed by the median filter first to reduce the noise and then calibrated by the average filter to obtain the final ranging value for further estimation.

5.3.3 Anchor Selection

The ranging data are forwarded to the fusion center after preprocessing. The system fuses the ranging data together with the anchor's positions to derive the target's position. Before that, the system firstly identifies which data is reliable and then chooses the related anchors. This procedure is the component of anchor selection.

Anchor selection scheme is to choose effective anchors with their ranging measurement at a typical time to derive the target's position. The selection is based on some specific metrics. One of the method is based on the geometric effect between the positions of anchors and the target [130]. Another metric is based on the information quality, which considers whether the ranging measurement is reliable [21]. In this work, anchor selection only considers the reliability of the ranging measurement.

When the ranging data is propagated to the anchor selection, the component firstly selects the measurements without communication outage, since the anchors that can not communicate with the target is useless. Meanwhile, the system also drops the measurements with extreme large values. In this system, the effective measurement is considered as the measurement obtained from the anchor which is near to the target. If the ranging value is quite large, the anchor can not be quite close to the target. Especially, the ranging measurement can not be too large in a building. If the real distance between the anchor and the mobile target exceeds $20m$, there is no effective communication channel between them. Thus, the upper bound of the ranging value is set up as $15m$.

5.3.4 Tracking Algorithm

The tracking algorithm is to estimate the target's position with a given set of measurements and anchors. Many algorithms can be implemented in this component, which are not restricted to the PFs. In general, the tracking algorithms can be classified into three categories: geometric method, convex optimization method and nonlinear filtering method.

Geometric estimators employ range measurements to obtain the position through a geometric way, such as trilateration, min-max algorithm or geo-n algorithm [131]. Convex optimization methods, such as linear-least-squares (LLS) method or nonlinear-least-squares (NLLS) method, formulate the range-based positioning problem as least-squares problem and estimate the optimal point [27, 28]. Filtering methods, such as Kalman filter (KF) or PF, use movement transition information and achieve better performance for mobile target tracking [29]. Here, the popular

used algorithms are listed below:

Non-Linear-Least-Square (NLLS)

NLLS formulates the location estimation problem as minimizing the sum of the squared residuals between the observed ranges and the estimated distances $||\boldsymbol{x}_t - [a_j^X, a_j^Y]^T||$ [132]:

$$\tilde{\boldsymbol{x}}_t = \arg\min \sum_{j=1}^{N} (||\boldsymbol{x}_t - [a_j^X, a_j^Y]^T|| - z_t^j)^2 \tag{5.3}$$

NLLS can be solved by using Newton optimization algorithms. Firstly, an initial possible position is chosen and then start searching the optimal point through the minus gradient direction. The estimated position is improved recursively until a local minimum objection function of (5.3) is found. The estimation accuracy is quite good. However, to find a global optimal position, the searching procedure must run several times, especially when the estimated position is near the optimal position, the searching rate becomes quite slow, which is expensive in terms of computation overhead.

Linear-Least-Square (LLS)

LLS linearizes NLLS by constructing linear matrix to form the expression $\boldsymbol{A}\boldsymbol{x}_t = \boldsymbol{b}$, where $\boldsymbol{A}$ is a $(n-1) \times 2$:

$$\boldsymbol{A} = \begin{pmatrix} a_1^X - a_N^X & a_1^Y - a_N^Y \\ a_2^X - a_N^X & a_2^Y - a_N^Y \\ \vdots & \vdots \\ a_{N-1}^X - a_N^X & a_{N-1}^Y - a_N^Y \end{pmatrix} \tag{5.4}$$

and $\boldsymbol{b}$ is formulated as:

$$\boldsymbol{b} = \frac{1}{2} \begin{pmatrix} (z_t^N)^2 - (z_t^1)^2 + ||[a_1^X, a_1^Y] - \boldsymbol{x}_t||^2 \\ (z_t^N)^2 - (z_t^2)^2 + ||[a_2^X, a_2^Y] - \boldsymbol{x}_t||^2 \\ \vdots \\ (z_t^N)^2 - (z_t^{N-1})^2 + ||[a_{N-1}^X, a_{N-1}^Y] - \boldsymbol{x}_t||^2 \end{pmatrix} \tag{5.5}$$

Then, LLS can be solved analytically:

$$\tilde{\boldsymbol{x}}_t = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{b} \tag{5.6}$$

Since the estimation problem is linearized, the computing overhead is reduced significantly. LLS is widely used in location algorithms for its simplicity. However, LLS is vulnerable to the measurement noise, and the performance is not as good as NLLS. Several improved method is

proposed for LLS [133, 118].

Min-Max Algorithm

Min-max algorithm, also known as bounding-box algorithm, is a simple and straightforward estimator. The main idea is to build a square region (like a box) around the anchor. The box is drawn according to the anchor's position and the range measurement: $[a_j^X - z_t^j, a_j^Y - z_t^j] \times [a_j^X + z_t^j, a_j^Y + z_t^j]$. Then, the estimated position is in the center of the intersections of these squares. And the intersections are determined by:

$$[max(a_j^X - z_t^j), max(a_j^Y - z_t^j)] \times [min(a_j^X + z_t^j), min(a_j^Y + z_t^j)] \tag{5.7}$$

Since the final position is arbitrarily determined at the center of (5.7), several improvements are studied to provide more reasonable estimation [23, 118]. These methods do not increase the computing overhead too much. Thus, min-max is still the simplest and robust algorithm for localization.

GEO-N Algorithm

Geo-n algorithm is another geometrical estimator for range-based location systems [24]. It uses the basic idea of trilateration, in which the intersection of circles is the estimated position. Geo-n first draws a set of circles according to the range measurements. Then, it makes a selection of all pairwise intersection points between circles. If the intersection points do not contribute to the localization or are suspected to increase the position error, Geo-n removes these points by using median filter. The remaining intersection points are assigned weights according to the range measurement and the final estimation are obtained based on the normalized weighted intersections. Geo-n also has the ability to approximate the estimation when there is no intersections when the range is too short.

Geo-n has strong robust and accurate estimation performance. Although it claims to be complicated, the calculation is faster than NLLS. Besides, unlike other location algorithms, the geometric effect of anchor-target is effectively reduced. The performance is almost equally good for all the playing field.

Extended Kalman Filter

Kalman filter is a set of recursive equations which are based on the prediction and update processes. The Kalman filter is suitable for the linear system with the Gaussian noise. For nonlinear systems, extended Kalman filter (EKF) is applied. The EKF linearizes the nonlinear measurement function and estimates the position within KF framework by using the Taylor expansion [134]. However, during the estimation, the EKF still assumes that the error follows zero-mean normal distribution.

The prediction and update processes are still based on the Bayesian estimation framework in EKF. In the prediction step, EKF assumes the target movement follows a linear transition model, which is:

$$\boldsymbol{x}_t = \boldsymbol{F}_t \boldsymbol{x}_{t-1} + \boldsymbol{q}_t \tag{5.8}$$

where the target state $\boldsymbol{x}_t = [p_t^X, \dot{x}_t, p_t^Y, \dot{y}_t]$, $\dot{x}_t$ and $\dot{y}_t$ denote the speed in each coordinate and $\boldsymbol{F}_t$ is the transition matrix:

$$\boldsymbol{F}_t = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{5.9}$$

where Δt is the time interval; $\boldsymbol{Q}_t$ is the covariance of the prediction error:

$$\boldsymbol{Q}_t = \begin{pmatrix} \sigma_x^2 & 0 & 0 & 0 \\ 0 & \sigma_{\dot{x}}^2 & 0 & 0 \\ 0 & 0 & \sigma_y^2 & 0 \\ 0 & 0 & 0 & \sigma_{\dot{y}}^2 \end{pmatrix}. \tag{5.10}$$

Then, the measurement $\boldsymbol{z}_t$ follows:

$$\boldsymbol{z}_t = h_t(\boldsymbol{x}_t) + \boldsymbol{v}_t \tag{5.11}$$

where $\boldsymbol{v}_t$ is the covariance of the measurement error, which follows the normal distribution $\boldsymbol{v}_t \sim \mathcal{N}(0, \boldsymbol{R}_t)$. Then the state $\boldsymbol{x}_t$ is formulated within the Kalman filter framework:

$$\bar{\boldsymbol{x}}_t = \hat{\boldsymbol{x}}_t + \boldsymbol{K}_t(\boldsymbol{z}_t - h_t(\boldsymbol{x}_t)) \tag{5.12}$$

where K_t is the Kalman gain:

$$\boldsymbol{K}_t = \boldsymbol{P}_t^- \boldsymbol{H}_t^T (\boldsymbol{H}_t \boldsymbol{P}_t^- \boldsymbol{H}_t^T + \boldsymbol{R}_t)^{-1} \tag{5.13}$$

where P_t^- is the prior estimation covariance, which is calculated as:

$$\boldsymbol{P}_t^- = \boldsymbol{F}_t \boldsymbol{P}_{t-1} \boldsymbol{F}_t^T + \boldsymbol{Q}_t \tag{5.14}$$

where $\boldsymbol{P}_{t-1}$ is the posterior covariance in the previous estimation and $\boldsymbol{Q}_t$ is the prediction error covariance. For the recursive estimation, the posterior covariance at current time P_t is formulated

as:

$$\boldsymbol{P}_t = (\boldsymbol{I} - \boldsymbol{K}_t \boldsymbol{H}_t)\boldsymbol{P}_t^- \tag{5.15}$$

Then the projection function $\boldsymbol{H}_t$ is the Jacobian matrix of the partial derivative of $h_t(\boldsymbol{x}_t)$ over $\boldsymbol{x}_t$:

$$\begin{aligned}
\boldsymbol{H}_t &= \begin{pmatrix}
\frac{\partial z_t^1(\boldsymbol{x}_t)}{\partial p_t^X} & \frac{\partial z_t^1(\boldsymbol{x}_t)}{\partial \dot{x}_t} & \frac{\partial z_t^1(\boldsymbol{x}_t)}{\partial p_t^Y} & \frac{\partial z_t^1(\boldsymbol{x}_t)}{\partial \dot{x}_t} \\
\vdots & \vdots & \vdots & \vdots \\
\frac{\partial z_t^j(\boldsymbol{x}_t)}{\partial p_t^X} & \frac{\partial z_t^j(\boldsymbol{x}_t)}{\partial \dot{x}_t} & \frac{\partial z_t^j(\boldsymbol{x}_t)}{\partial p_t^Y} & \frac{\partial z_t^j(\boldsymbol{x}_t)}{\partial \dot{x}_t} \\
\vdots & \vdots & \vdots & \vdots \\
\frac{\partial z_t^n(\boldsymbol{x}_t)}{\partial p_t^X} & \frac{\partial z_t^n(\boldsymbol{x}_t)}{\partial \dot{x}_t} & \frac{\partial z_t^n(\boldsymbol{x}_t)}{\partial p_t^Y} & \frac{\partial z_t^n(\boldsymbol{x}_t)}{\partial \dot{x}_t}
\end{pmatrix} \\
&= \begin{pmatrix}
\frac{p_t^X - a_1^X}{\sqrt{(p_t^X - a_1^X)^2 + (p_t^Y - a_1^Y)^2}} & 0 & \frac{p_t^Y - a_1^Y}{\sqrt{(p_t^X - a_1^X)^2 + (p_t^Y - a_1^Y)^2}} & 0 \\
\vdots & \vdots & \vdots & \vdots \\
\frac{p_t^X - a_j^X}{\sqrt{(p_t^X - a_j^X)^2 + (p_t^Y - a_j^Y)^2}} & 0 & \frac{p_t^Y - a_j^Y}{\sqrt{(p_t^X - a_j^X)^2 + (p_t^Y - a_j^Y)^2}} & 0 \\
\vdots & \vdots & \vdots & \vdots \\
\frac{p_t^X - a_N^X}{\sqrt{(p_t^X - a_N^X)^2 + (p_t^Y - a_N^Y)^2}} & 0 & \frac{p_t^Y - a_N^Y}{\sqrt{(p_t^X - a_N^X)^2 + (p_t^Y - a_N^Y)^2}} & 0
\end{pmatrix}.
\end{aligned} \tag{5.16}$$

The EKF is only suitable for the Gaussian noise environment. However, the real indoor target tracking system does not always follows the Gaussian distribution. In addition, the EKF may loss information during the first order Taylor expansion. Thus, the estimation performance is not good for the real applications. But the computation complexity of the EKF is still low.

5.3.5 Performance Evaluation

To evaluate the accuracy of the tracking algorithm and the system, an addition component is required to be implemented. Performance evaluation compares the actual target position with the estimated position. The real actual target position is obtained from other system, e.g. the reference system which uses a robot with a really accurate localization system to record the position. Then, the system compares the records of the real position and the estimated position to examine the estimation accuracy.

Different applications requires different metrics for localization algorithms. In general, the estimation should be precise, which is defined as the average error approaching to 0, and accurate, which should have low variance. For the continuous tracking, the system should not loss the target and show a reasonable trajectory. Some static location based service, e.g. facebook, only require that the extreme error is not larger than $1km$. All these requirements focus on different aspects of localization algorithms. Thus, we employ several metrics to evaluate the estimation performance comprehensively, considering the performance of accuracy and precise,

and also the feature of consistency and the extreme case. The main idea is to calculate the distance from the estimated location to the real position, which is defined as the root squared error $e = \sqrt{(p_t^X - \hat{p}_t^X)^2 + (p_t^Y - \hat{p}_t^Y)^2}$. For sequential process, a sequential set of positions is compared with the real positions. Then, several metrics based on the squared error are derived.

Root Mean Square Error (RMSE)

The root mean square error (RMSE) is a general performance metric which is often applied in the tracking system. The RMSE calculates the mean square error of the estimated positions and converts to the root value as the average distance from the real positions:

$$e_{RMSE} = \sqrt{\frac{1}{n_t}\sum_{t=0}^{n_t}((p_t^X - \hat{p}_t^X)^2 + (p_t^Y - \hat{p}_t^Y)^2)} \tag{5.17}$$

where n_t is the total length of the discrete time sequence. A similar metric is named mean square error (MSE), which is the squared value of the RMSE, also represents the overall performance. Since the system is range based tracking application, the RMSE is used as the error distance.

Root Median Square Error (ME)

The RMSE is the general performance of the estimation, which contains extreme large errors. Sometimes, researchers attempt to know what is the usual error during the estimation, which is more realistic since large error does not appear very often. Thus, root median square error (RMeSE) or called median error (ME) is applied:

$$e_{ME} = median\sqrt{\{(p_t^X - \hat{p}_t^X)^2 + (p_t^Y - \hat{p}_t^Y)^2\}_{t=0}^{n_t}} \tag{5.18}$$

where only median value is chosen and the maximum and minimum error do not involved in the calculation. The RMeSE indicates the estimation error with a high probability. It is also a general performance metric for the tracking applications.

Average Error

The average error (AE) is used to indicate the bias of the estimation error. It calculates the average error value of each coordinate and then derive the error distance from the actual position:

$$e_{AE} = \sqrt{(\sum_{t=0}^{n_t}(p_t^X - \hat{p}_t^X))^2 + (\sum_{t=0}^{n_t}(p_t^Y - \hat{p}_t^Y))^2} \tag{5.19}$$

The AE can not illustrate the overall performance of the system like RMSE, because AE only represents the expectation of the error in each coordinates. Even the AE of an algorithm is 0, the variations can be large which depicts the trajectories frequently fluctuate around the actual

trajectories significantly. In this case, the estimation trajectories are not clear for the applications and users. However, it can indicate which algorithm is a biased estimator. If the AE of an algorithm approaches to 0, this algorithm is an unbiased estimator, such as PF. Some geometric localization algorithms and convex optimization algorithms are biased estimators. The overall performance of a biased estimator may not be worse than an unbiased estimator. However, the biased estimator means that the estimated position always have a distance to the actual position.

Percentage Error

Min and max errors are the lower and upper bound of the estimation error, which indicates the best and worst cases during the estimation. However, these values are the extreme case, which mislead the impression of a typical algorithm performance. Thus, a percentage error is a good metric for estimation. Initially, we sort all the estimation error. Then, we apply the max error which is lower than 5% of all the errors and the min error which is higher than 95% of all the estimation errors as the metrics. In this case, these values can indicates the boundaries of the general estimation performance for a typical algorithm.

Loss Rate

If the estimation error is too large, the system will consider it loses tracking the target in this measurement period. In addition, many localization require at least three anchors to derive the target's position. If only the number of anchors are too small or no ranging measurements are obtained from preprocessing, it is also considered as loss of tracking. For LLS, if the projection matrix is irreversible, the algorithm can not obtain an estimation, which is also feasible for some other algorithms that can not calculate the position even with the given measurements. All the cases mentioned above can lead to the loss of tracking. The loss rate is the percentage of the number of lost tracking during the whole estimation. The loss rate indicates the robustness of the algorithms for the real application with multiple situations. For filtering methods, e.g., the EKF and the PF, if no ranging measurements are available, the predicted state can represent the estimated state. Thus, no target loss even if the communication outrage happens in principle, unless the prediction error is higher than $10m$. For indoor localization systems, $10m$ estimation error indicates the system cannot obtain the target correctly, e.g., locating the target in a wrong room.

Cumulative Distribution Function

Another important metric to evaluate the estimation performance is the cumulative distribution function (CDF) of the error. The CDF curves also illustrate the major performance of the algorithm. If the curve increases rapidly, the algorithm performs quite well. Usually, the upper curve indicates the better performance. In addition, the CDF curves can clearly indicate how the algorithm performs in different estimation error scenarios.

5.4 Context-Aware Particle Filter

Initially, the system imports all the information related to the target tracking application, which contains the map of the building. The building layout is quite useful for the system, because the movement in the indoor environment can not be too large within a small area. The building layout provides constrains for the target. For indoor localization, the target can not move fast and widely as in the outdoor case. Target in the building always moves with a low and constant speed. And its movement behavior are limited to the building layout. Integrating with building information, such as map matching, the localization system can avoid wired motion tracking estimation such as walking through a wall or jumping out of the building [2], and also reduce the estimation error [71].

Using building information still has some drawbacks in the real system. First, the methods which use building information, such as map matching, a large database should be built and the model is quite complicated. Secondly, some constraint methods have prior knowledge of target movement which is not feasible for the real scenario. For instance, particle elimination methods know the target moves along the hallway, thus they will eliminate the particles which are not in the hallway. However, in the real world case, targets can move anywhere they want, if the prior information is wrong, the tracking path is limited in the constraint region. Finally, even if the prior information is correct, the measurement noise still influence the estimation significantly.

In this section, a new PF combined with the previous work is proposed. A low complexity PF scheme which integrates target's motion context and building information is introduced. First, the indoor building is divided into several possible regions during the initialization. The region information is stored in the system database. Then, the system predicts the mobile target's motion behavior. The region where the target belongs to is predicted based on the linear prediction equation in particle filter. With the detected region, a joint constraint conditions are constructed based on the region information and measurement information. The joint constraint conditions are robust to the ranging noise, no matter of LOS or NLOS effect. Besides, it does not need to concern whether the target is locked in the predicted region. The particle samples are generated within this constraint conditions. Finally, the estimation is attained using proposed likelihood adaptation method, which is presented in the previous chapters.

5.4.1 Motion Detection using Building Layout

The building consists of rooms and hallways. The target has different motion behaviors in rooms and hallways. Thus the system divides the building layout into several region according to the motion behavior in the building. It records the coordinates of each region as the constraint condition. If the target is predicted to move in this region, one constraint conditions are the coordinates of this region. No additional information is recorded in our system, such as NLOS conditions. Thus the complexity is quite low.

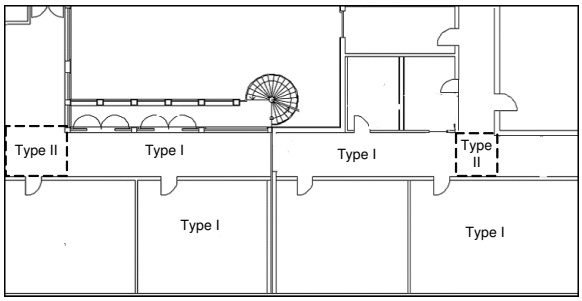


Figure 5.3: Region partition: Type I: room or corridor without cross; Type II: corridor on the cross.

It is easy to define a room as a single region. However, the movement of the target in the hallway can be different. Thus, the hallway is classified into two types of region, which is shown in Fig. 5.3. The first one is the region on the cross. In this region, the target can either turn right or left, and can also forward or backward. Thus, the constraint condition is less reliable and should not restrict the target estimation. The second type is the corridor with no corners or cross. The target can only move forward or backward, no other options. In this case, the constraint conditions are reliable in this region, which helps us adapt the particle weights.

Still, linear prediction function is applied to predict the target movement and estimate the region according to its movement.

$$\boldsymbol{x}_t = \boldsymbol{F}_t \boldsymbol{x}_{t-1} + \boldsymbol{q}_t \tag{5.20}$$

where $\boldsymbol{x}_t = [p_t^X, p_t^Y]^T$ is the target's movement state; $\boldsymbol{F}_t$ is the linear transition matrix; $\boldsymbol{x}_{t-1}$ is the previous state and $\boldsymbol{q}_t$ is the prediction noise $\boldsymbol{q}_t \sim \mathcal{N}(0, \boldsymbol{Q}_t)$. The region is chosen based on:

$$\begin{cases} X_{min}^k \leq p_t^X \leq X_{max}^k \\ Y_{min}^k \leq p_t^Y \leq Y_{max}^k \end{cases} \tag{5.21}$$

where $[X_{min}^k, X_{max}^k, Y_{min}^k, Y_{max}^k]^T$ denotes the coordinates of the kth region. The constraint conditions for particle sampling are also based on (5.21).

5.4.2 Constraint Sampling

The region constrain is the first constraint condition for particle sampling. However, if the motion prediction is not accurate, the region constraint will limit the particle sampling within a wrong area and further lead to a wrong estimation. Thus, the constrains only based on the linear prediction function is not reliable. We propose a second constraint condition: measurement constraint, which is generated according to the current ranging measurement information. In conventional methods, if the ranging measurement is obtained, the target's location should be

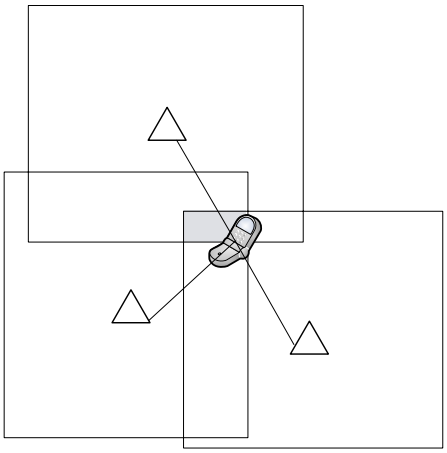


Figure 5.4: The constraint conditions drawn by min-max algorithm

estimated. This estimation is also not reliable since the measurement contains error. However, the ranging measurement can provide some prior information to approximate the area where the target belongs to. Min-max algorithm is a robust and simple algorithm. It draws rectangle area according to the range measurement, and the drawn area is like a box, as shown in Fig. 5.4. Fortunately, the drawn box has the same shape of the region constrains. Thus, it can be used as another constraint condition and it can be combined with the region constrain.

One advantage of using min-max algorithm is that the estimation error of min-max algorithm does not increase when the measurement error is high. It is suitable for indoor localization since the estimation performance is stable no matter of the covariance or distributions of the measurement error. Then, we use it to draw a second constraint region:

$$\begin{cases} s_{X,t}^{min} = max\{a_j^X - z_t^j\}_{j=1}^N \\ s_{X,t}^{max} = min\{a_j^X + z_t^j\}_{j=1}^N \\ s_{Y,t}^{min} = max\{a_j^Y - z_t^j\}_{j=1}^N \\ s_{Y,t}^{max} = min\{a_j^Y + z_t^j\}_{j=1}^N \end{cases} \tag{5.22}$$

where $(a_j^X, a_j^Y)^T$ denotes jth anchor's position; z_t^j is the range measurement for jth anchor. Then, we obtain a box shape of constrain, which is based on the ranging measurement.

The system combines these two conditions to draw an integrated constrains. Sometimes, these two conditions are too different. In Fig. 5.5, three types of joint constraint conditions are illustrated. For type I, the constraint regions are overlapped. As indicated in Fig. 5.5, region constrain contains the measurement constrain. In this case, the minimum area is chosen. For type II, the part of region constrain and measurement constrain are overlapped. It means that the prediction and measurement information lead to the different estimations. The overlapped

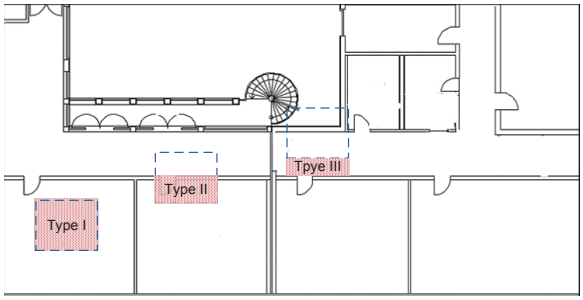


Figure 5.5: The joint constraint region. Three types: type I, the constraint regions are overlapped; type II, part of the constraint regions are overlapped; type III, the constraint regions are significantly different.

part is the tradeoff between prediction and measurement. In this case, the overlapped part is chosen as the joint constraint region. Type III illustrates that prediction and measurement information are too different. In this case, the tradeoff is to choose the area between region constrain and measurement constrain. To summary these three types, the joint constraint conditions are established just like min-max algorithm:

$$\begin{cases} max(X^k_{min}, s^{min}_{X,t}) \leq p^X_t \leq min(X^k_{max}, s^{max}_{X,t}) \\ max(Y^k_{min}, s^{min}_{Y,t}) \leq p^Y_t \leq min(Y^k_{max}, s^{max}_{Y,t}) \end{cases} \tag{5.23}$$

Within the constraint region, the particles can be sampled. In the traditional PF, particles are generated randomly according to the Markov process or based on the pre-assumed Gaussian distribution. However, the constraint region limits the area of the particles. Using Markov process or Gaussian distribution to generate particles may cause loss of generality. According to the max-entropy-principle, if the particles want to represent all of the possibilities, the probability distribution within the area should be uniform distribution. Thus, the particles are uniformly sampled within (5.23).

5.4.3 Likelihood Adaptation

Even with the constrain sampling method, the measurement error also suffers the same problem as the previous chapters mentioned before. Thus, the likelihood adaptation is still needed. The DGM and the procedure are the same. Note that, the general Gaussian model in the DGM for the real system is obtained from the histogram of the ranging error.

5.5 Experiment and Results

The real indoor target tracking system is implemented in a reference system. In this system, several wireless sensor nodes are deployed as anchors either along the corridor or in the offices

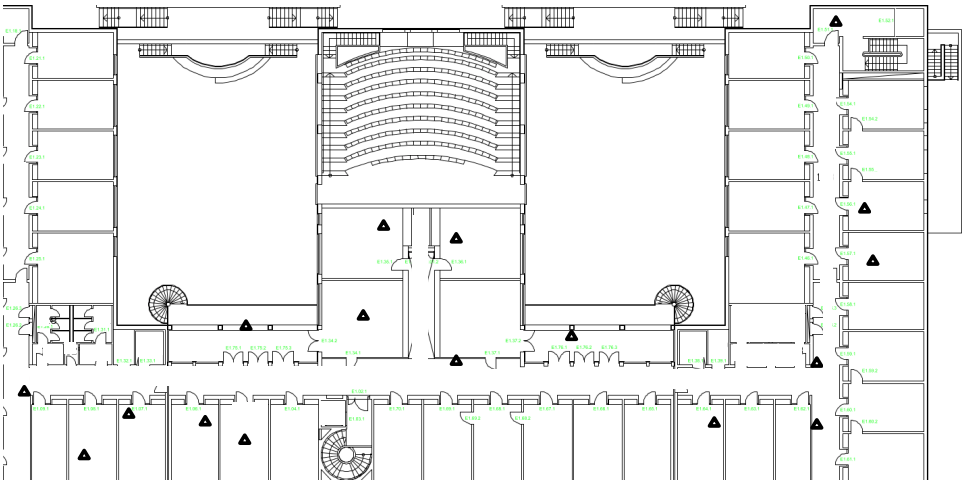


Figure 5.6: Building layout for indoor localization experiment and the robot trajectory. The triangles mark the positions of sensor nodes which are placed either in the offices or along the corridor.

of the research building, in the institute of computer science, Freie Universiteat Berlin. A robot carrying a sensor node as the target moved in the building with constant speed while recording its own positions [128, 135]. Since the error of the record position is much less than the estimated position based on TOF, we consider the record position is the actual position.

All sensors are integrated with the nanoPAN 5375 RF module with the 2.4 GHz transceiver and 1 Mbps data rate for the range measurement, LPC 2387 as micro-controller and CC1101 900 MHz transceiver as the radio transceiver for communications. The data collected from the sensor nodes are also the range measurement values which are based on TOF. At each measurement interval, the target carried by a robot is measured by the sensor nodes, meanwhile, the robot recorded its actual coordinates in the building. Fig. 5.6 depicts the map of the experimental building. The triangles, which are randomly deployed, mark the sensor nodes' positions. All the data used in the experiments are collected by Simon Schmitt, Stephan Adler, Thomas Hillebrandt, Heiko Will in their previous experiments [128, 135]. According to the statistical errors of measurements, the expectation of measurement error is 1 m and the standard deviation is about 5 m in general. Therefore, the simulations mentioned in Chapter 3 is close to the real environment. Besides the whole building, the system is also evaluated in several other indoor scenarios, e.g., the rooms, the small corridors and the hybrid scenarios.

For the localization algorithms, the adaptive particle filters mentioned in the previous chapters are implemented. For the DGM, the general Gaussian distribution follows $\mathcal{N}(1, 5^2)$. Besides, the CA-PF is also applied in the system. For each PF, 20 particles are used for a single estimation. Also, some common used localization algorithms are implemented, such as LLS, NLLS, min-max algorithm, geo-N algorithm and EKF. The estimation performances are compared in each

scenario.

5.5.1 Hybrid Room and Corridor

In the hybrid room and corridor scenario, 16 anchors are deployed in a small area of the building. Only 5 anchors are deployed along the corridor and the rest of anchors are put in the classrooms. Thus, if the robot moves along the corridor, the major ranging measurements are NLOS signals. In this scenario, two trajectories are employed. The first trajectory is just the path through the corridor, which is simple and full of NLOS ranging. And the second trajectory is complicated, as indicated in Fig. 5.7. The robot goes in and out of the classrooms, which approaches to the random walk. The ranging measurement contain hybrid LOS and NLOS signals and the scenarios change frequently. The distribution fitting can not work well for such real application. In this case, the DGM is suitable for modeling the error. The estimated trajectories of the adaptive PFs are presented in Fig. 5.7.

In this scenario, the CA-PF is applied with the layout information based on the small area. The numerical performance comparison is illustrated in Table 5.1. The average error for each algorithms are not 0. Although many algorithms claim themselves are unbiased estimators, there are biases in the real evaluation. Unlike the simulation, in this scenario, almost all the algorithms are biased estimators, except Min-Max algorithm. The RMSE and median error of the adaptive PFs are better than original PFs, which indicates that our proposed method can improve the estimation accuracy in general. However, for Min-Max algorithm and Geo-n algorithm, the RMSE and median error are also as good as the adaptive PFs. The percentage error of the adaptive PFs and Min-Max and Geo-n algorithms are almost the same. When consider the loss rate, Min-Max algorithm and Geo-n algorithm have non-zero loss rate. The adaptive PFs are more robust in this case with 0 loss rate. The A-BPF does not improve too much in this scenario. All the PFs have 0 loss rate except GPF, which indicates that the Gaussian approximation method is not suitable for the indoor scenarion. However, using DGM, the performance can be improved. Just like the results in the simulation in Chapter 3, A-CPF has the most accurate estimation and it is also robust to the environment with 0 loss of tracking. Besides, the estimation error is quite low. Thus, it is suitable to construct the constraint region for the indoor environment. The CDF curves are depicted in Fig. 5.8. In Fig. 5.8(a), it is clearly observed that the CDFs of the adaptive PFs are above the original PFs in general, which indicates the estimation accuracy improvement. In Fig. 5.8(b), the CDFs of A-BPF and A-CPF are still above other algorithms. Therefore, the adaptive PFs are suitable for target tracking in this scenario.

5.5.2 The building

The building of institute of computer science, Freie Universiteat Berlin, is employed for several indoor target tracking experiments. Here, two experiments are listed. In each experiment, many anchors are deployed in the building and the robot travels a long distant throughout the building.

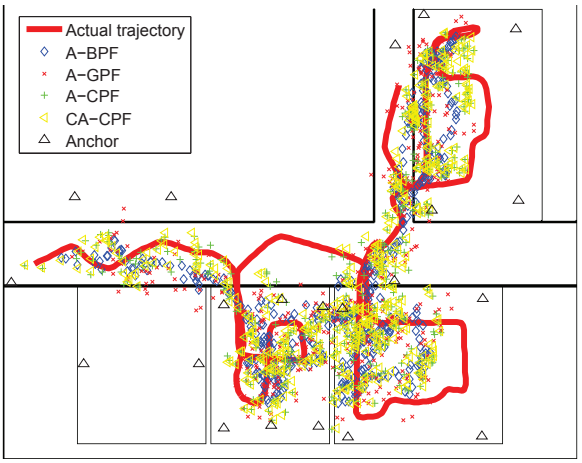
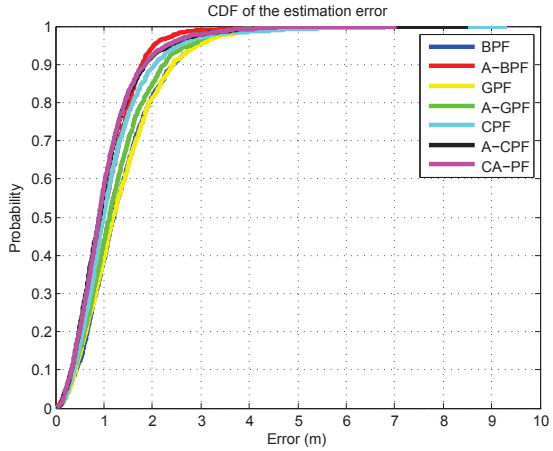


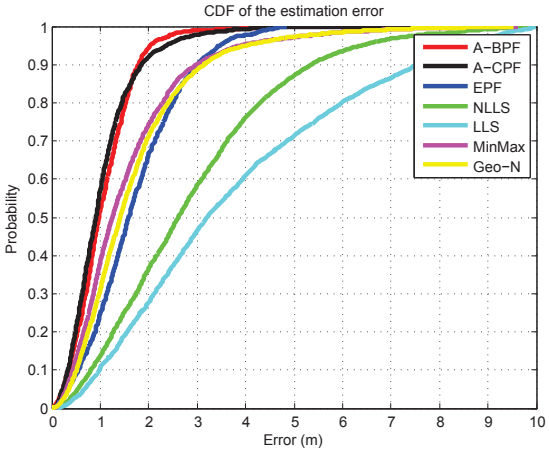
Figure 5.7: Trajectories in the hybrid scenario.

Table 5.1: Estimation Performance comparison (hybrid)

Alg.	AE(m)	RMSE(m)	ME(m)	$\leq 5\%$ (m)	$\geq 95\%$ (m)	Lost Rate
BPF	0.2593	1.5740	1.1994	0.2933	2.9278	0
A-BPF	0.2393	1.1258	0.9037	0.2346	1.8044	0
GPF	0.1367	1.8490	1.4047	0.3437	3.2460	0.44%
A-GPF	0.0370	1.4902	1.1851	0.3118	2.5515	0
CPF	0.1020	1.4358	0.9834	0.2707	2.5382	0
A-CPF	0.2504	1.2601	0.9007	0.2294	2.1678	0
CA-PF	0.2891	1.2253	0.8887	0.2202	2.3774	0
EKF	0.7214	2.3809	2.0248	0.3152	3.2735	0
NLLS	0.5446	2.7980	2.2784	0.4238	4.8836	13.97%
LLS	1.0685	3.5350	2.8666	0.4115	5.6693	15.52%
Min-Max	0.4425	1.5760	1.1441	0. 3002	2.8862	2.22%
Geo-N	0.3888	1.4276	1.1971	0.3142	3.0681	14.08%



(a) Adaptive PFs comparison



(b) Adaptive PFs and other algorithms comparison

Figure 5.8: CDF of the estimation errors.

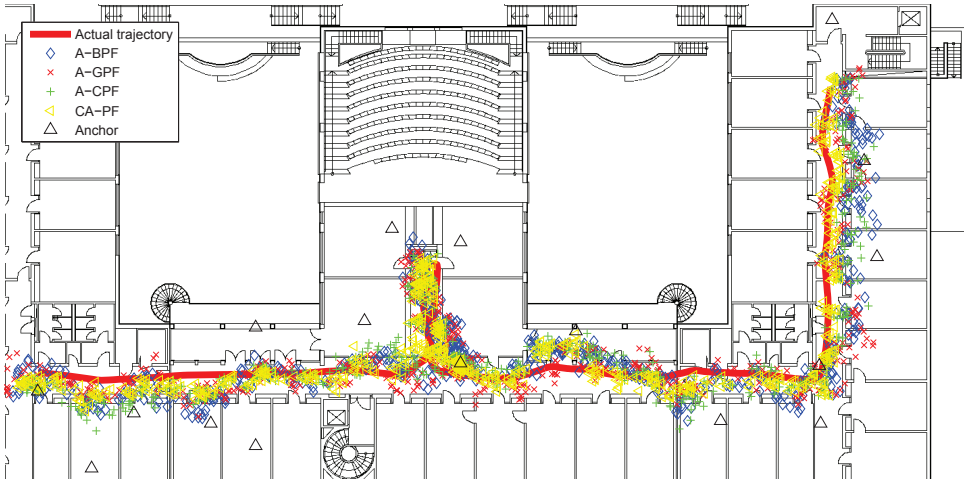


Figure 5.9: Trajectories in the building, with 17 anchors deployed.

In this scenario, CA-PF is implemented. Each room is defined as a constraint region. Besides, the corridor is divided into several regions, where the corners and the crosses are defined the separate regions and the rest parts are defined as another regions.

In the first experiment, 17 anchors are deployed. Two trajectories are examined during the experiment. The estimated trajectories are illustrated in Fig. 5.9. It is clearly observed that the trajectories of CA-PF closely follows the actual trajectory, which indicates its best performance among all the adaptive PFs.

In the second experiment, 25 anchors are used in the building. The robot also moves throughout the building. The estimated results are depicted in Fig. 5.10. It is also clearly observed that all the adaptive PFs can track the target and the trajectory of the CA-PF follows the target even closer.

For a general comparison, the numerical results are presented in Table 5.2. The estimation biases of all the algorithms are larger than in Table 5.1, which are represented by AEs. The RMSEs and MEs of the adaptive PFs, Min-Max algorithm and Geo-n are below 2m, which are more accurate than other algorithms. However, the errors above 95% of Min-Max algorithm and Geo-n algorithm are 1 or 2m higher than the adaptive PFs. Still, Min-max algorithm and Geo-n algorithm have loss rate which are not as robust as adaptive PFs. Compared with all the metrics, CA-PF outperforms other algorithms since it fuses more information than others.

The CDF results are presented in Fig. 5.11. Still, the CDF curves of the adaptive PFs are above other algorithms and original PFs. The CA-PF is still better than others which can be clearly observed.

In this experiment, with sufficient measurement samples, the number of particles for each particle filter scheme is adapted and the estimation performance is checked. The results are drawn

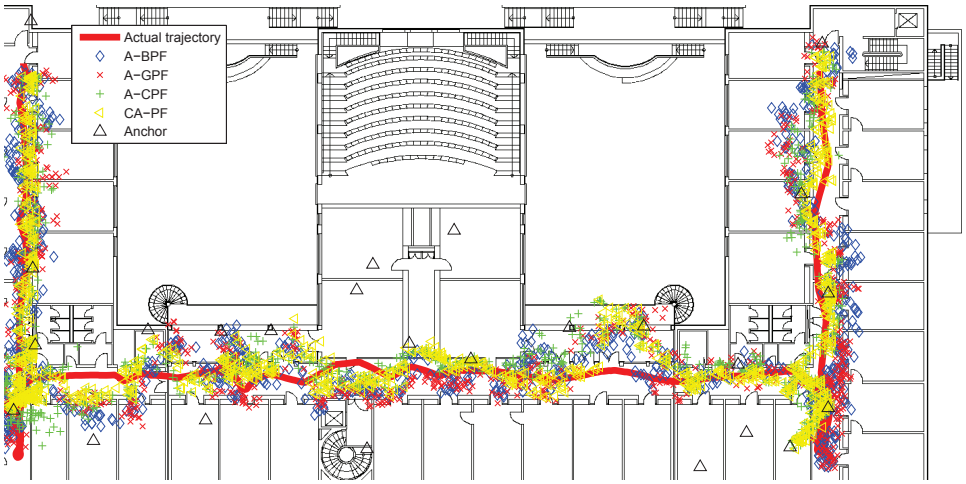
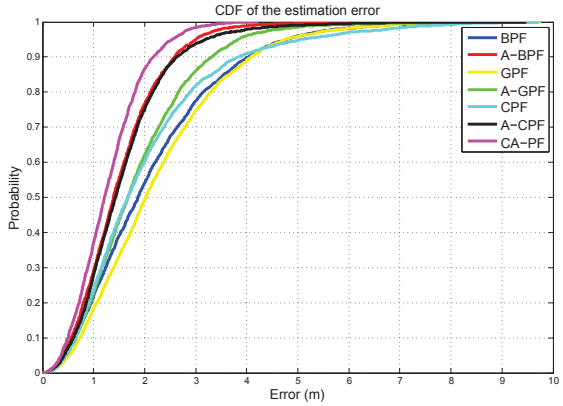
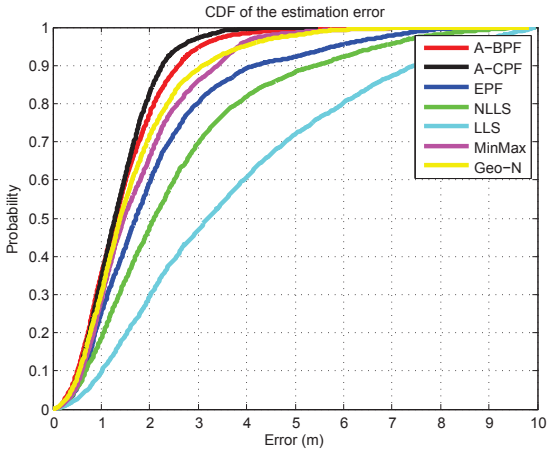


Figure 5.10: Trajectories in the building, with 25 anchors deployed



(a) Adaptive PFs comparison

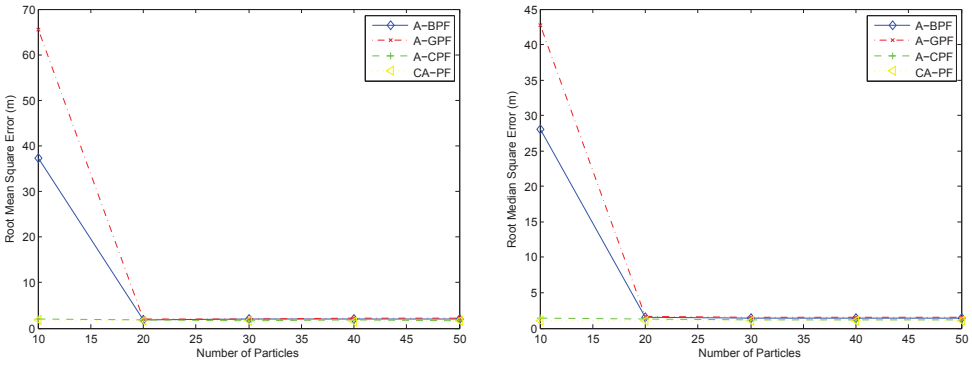


(b) Adaptive PFs and other algorithms comparison

Figure 5.11: CDF of the estimation errors.

Table 5.2: Estimation Performance comparison (building)

Algorithm	AE(m)	RMSE(m)	ME(m)	≤ 5 (m)	$\geq 95\%$ (m)	Lost Rate
BPF	0.5970	2.5560	1.8668	0.4093	4.7631	0
A-BPF	0.3919	1.7235	1.3907	0.3806	3.0011	0
GPF	0.5970	2.6483	2.0221	0.5192	4.7591	1.6%
A-GPF	0.4732	2.1669	1.6730	0.4530	3.8296	0
CPF	0.5185	2.5324	1.6736	0.4652	5.0381	0
A-CPF	0.3802	1.8331	1.4318	0.4207	3.1882	0
CA-PF	0.4196	1.4547	1.2135	0.3609	2.5443	0
EKF	0.3844	2.2128	1.6197	0.5203	5.5536	0
NLLS	0.4536	2.8376	2.0217	0.5336	6.8942	5.08%
LLS	0.5737	4.2883	3.1598	0.6228	8.3369	18.29%
Min-Max	0.7685	1.9764	1.3925	0.5331	6.5582	10.16%
Geo-N	0.6358	1.6613	1.2562	0.4326	5.1637	5.08%

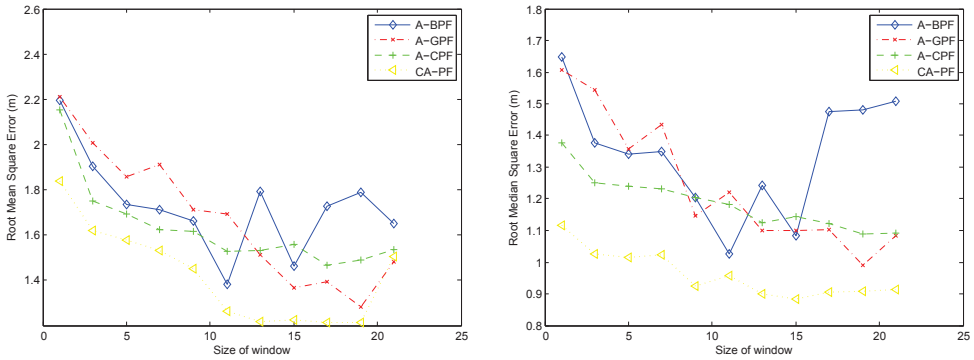


(a) Root mean square error (b) Root median square error

Figure 5.12: Estimation performance changes with different number of particles

in Fig. 5.12. Although more particles can achieve high accurate estimated location, the results depicted in Fig. 5.12 illustrate that only 20 or 30 particles are sufficient for estimation, which reduces much computational complexity. Especially for the A-CPF and the CA-PF, increasing the number of particles does not improve too much of the performance. Because the constraint region is not so large for the indoor environment, thus the generated particles are within a small area, which does not change the overall performance. In this case, a small group of particles is sufficient.

In the pre-processing, the ranging measurements are firstly filtered by median filter and the average filter. The ranging error is reduced in the pre-processing. In this experiment, the performance is also evaluated with the adaptation of the window size of the median filter and the average filter. The number of samples in the window is adapted from 1 to 21. When the number of the samples is 1, it means that no pre-processing is applied. The results are illustrated in Fig.



(a) Root mean square error (b) Root median square error

Figure 5.13: Estimation performance changes with different number of measurement samples

5.13. The estimation error is quite high without pre-processing when the number of the samples is 1. With the increase number of the samples, the estimated error drops accordingly. However, for some particle filters, large samples in the window do not represent low estimated error. For the A-BPF, when the window size is large, the estimated error is also large. Thus, it is not necessary to use large window size in pre-processing, which also leads to high computational complexity. For other adaptive particle filters, the estimated error is decreasing with the large window size.

Note that, the performance for each localization algorithm can change with different scenario. For indoor target tracking, the environment is complicated. In the classroom, all the algorithms show good performance, while the performance degrades in the hybrid environment. It can not be guaranteed that one algorithm is always good in any scenario. However, the constrained based methods, e.g. the A-CPF and the CA-PF, are robust to the noise and environmental changes.

5.6 Summary

The system architecture for real indoor target tracking has been illustrated. The pre-processing and the anchor selection are developed as the assisted components for the localization algorithm. New algorithm, which is CA-PF, is also proposed for the real application. The whole system is evaluated in several scenarios together with many localization algorithms.

In general, the RMSEs and MEs of the adaptive PFs are 0.5m better than the original PFs with 0 loss rate. Other algorithms can even achieve 18.29% loss rates with high RMSEs. Therefore, the proposed adaptive PFs are suitable for the indoor target tracking applications. The computation cost of the adaptive PFs are not high since only a few particles are involved in the estimation. The CA-PF has a higher cost than other adaptive PFs since the CA-PF requires searching the database to construct the layout constrains.

In the design of the localization system, the localization algorithms are implemented together with other components. The pre-processing and anchor selection are introduced to provide reliable data for the estimation. In the pre-processing, increasing the window size properly can reduce the final estimation error. However, increasing the window size may also loss some measurement data if the window is too large. Therefore, the design of localization system should not only rely on the pre-processing part.

Chapter 6

Distributed Particle Filter Solutions

6.1 Introduction

In the previous chapters, all the considered PFs are implemented in a centralized way. Every node forwards the ranging measurement to the fusion center, which receives information and makes decisions about the whole network, through the whole network and the fusion center estimates the target. The network topology is not considered. In the real experiment, the fusion center is within the robot and all the sensor nodes transmit the data through only one hop network routing. However, in a large network infrastructure, the data should be propagated through multi hops to the fusion center, which requires additional topology management. If the fusion center fails or some transfer nodes fail, the whole system can not work well. Thus, a centralized method is not robust. It is feasible to implement distributed algorithm of target tracking within a whole network. Besides, in some scenarios, the local network node also need to know the target state in order to provide location based service.

Distributed particle filter (DPF) is an alternative solution which calculates particle likelihoods based on local information of each sensor node and then fuses the likelihoods as weights to form a global posterior PDF. Several DPF schemes use message passing to propagate particles through a spanning tree or Hamiltonian cycle to calculate global weights [31, 103, 51, 104]. However, these schemes require additional topology management which is not robust in a mobile and vulnerable wireless condition. Recently, to make the distributed particle filter more robust, consensus based estimation algorithms are employed such as gossiping algorithms [105, 106] or consensus algorithms [107, 108]. However, communication overhead is increased due to particle transmissions and gossiping iterations. For a network with N nodes and each node maintain 1000 particles, the communication overhead for a single estimation is $1000 \times N^2$ [113].

One of the solutions is to compress or transform the particles into several parameters such as Gaussian mixture model (GMM) to approximate the posterior PDF. D.Gu $et\ al.$ proposed a

gossip-based expectation-maximization algorithm (EM) algorithm to estimate the parameters of a mixture approximation of the global posterior likelihood [110]. O.Hlinka $et\ al.$ employed a joint likelihood consensus algorithm to fuse estimations of a Gaussian distributed particle filter [111]. Boris $et\ al.$ used Gaussian product approximation method to estimate the global posterior PDF [112]. All of the mentioned algorithms effectively reduce the communication overhead but diminish the estimation accuracy.

An alternative approach is to share particles and weights among different nodes. Sun Hwan $et\ al.$ proposed a Markov chain distributed particle filter (MCDPF) which transmitted particles through the network and calculated the associated weights based on local likelihood functions [109]. The accuracy of the MCDPF is similar to the centralized PF but the algorithm does not drop particles with small weights which leads to redundant communication overhead. Deniz $et\ al.$ used selective gossip to share particles which have significant weights and every sensor node updates particles with maximum weights to achieve consensus [113]. This algorithm, which is similar to our work, can effectively reduce communication overhead if the quantity of selected particles is much smaller than the overall particles. However, the resampling stage of this algorithm updates particles based on local weights which can not ensure nodes converging to consensus. Besides, the estimation can not achieve optimal based on local weights updating.

In this chapter, two novel DPFs are proposed. The first one is named PSG-DPF, while the second one is named K-selective DPF. Both of them achieve the accurate estimation with reducing the communication cost. They are also applicable to the real indoor networks.

6.2 Network Model

Assume a wireless network of N nodes that forms time-varying graphs $\mathcal{G}_t(\mathcal{V}_t, \mathcal{E}_t), t = 0, 1, 2, \ldots$ is the time step. The set of vertices $\mathcal{V}_t = \{1, \ldots, n\}$ represents the participated sensor nodes and the wireless links construct the set of edges $\mathcal{E}_t$, which is a set of unordered node pairs $\{u, v\} \in \mathcal{E}_t$ when u and v is connected at time t. The network model is illustrated in Fig. 6.1. In the wireless network, the nodes can both transmit and receive wireless signals at 2.4GHz frequency band. The links between two nodes are either free space channels or the indoor propagation channels. We assume that the wireless links between two nodes within a certain distance is reliable and the wireless interference can not disturb the communications. In this case, the nodes can communicate with each other directly. In a distributed system, it is assumed that $\mathcal{V}_t$ and $\mathcal{E}_t$ are unaware to all the nodes in the network, but nodes can sense their neighbors and establish a bidirectional wireless link. Then a particle filter is implemented to obtain the

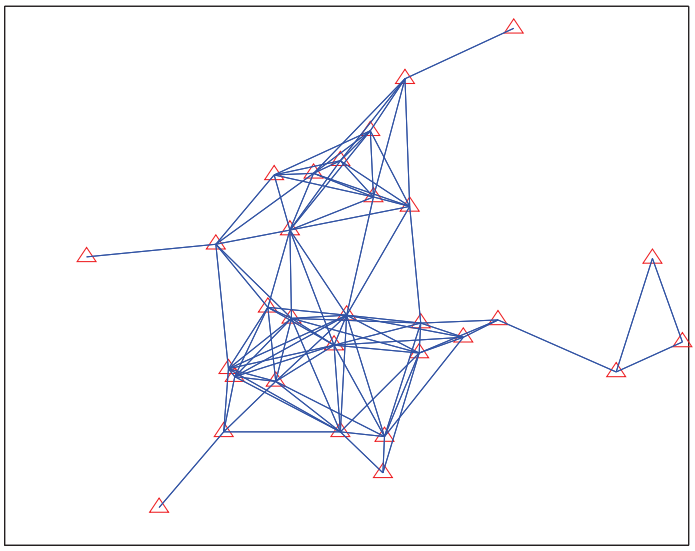


Figure 6.1: Distributed network model

global PDF of the state $p(\boldsymbol{x}_t|\boldsymbol{z}_{1:t})$, which can be expressed as:

$$p(\boldsymbol{x}_t|\boldsymbol{z}_t) = \sum_{i=1}^{N_s} w_t^i \delta(\boldsymbol{x}_t - \boldsymbol{x}_t^i). \tag{6.1}$$

In equation (6.1), N_s is the number of particles; $\boldsymbol{x}_t^i$ is the global particle with associate weight w_t^i which dependents on the measurement joint likelihood function $p(\boldsymbol{z}_t|\boldsymbol{x}_t^i)$, $w_t^i \propto w_{t-1}^i p(\boldsymbol{z}_t|\boldsymbol{x}_t^i)$; and $\boldsymbol{z}_t = \{z_t^j | j \in \mathcal{V}_t\}$ is the measurements set from all the sensor nodes at time step t, where $j \in \mathcal{V}_t$. Assume the measurements and local function models in every sensor node are unknown to others, and the sensor noise are conditionally independent. According to [113], the measurement joint likelihood function $p(\boldsymbol{z}_t|\boldsymbol{x}_t^i)$ can be factorized as:

$$p(\boldsymbol{z}_t|\boldsymbol{x}_t^i) = \prod_{j \in \mathcal{V}_t} p(z_t^j|\boldsymbol{x}_t^i). \tag{6.2}$$

where $p(z_t^j|\boldsymbol{x}_t^i)$ is time and sensor dependent likelihood function. Then, the estimated state can be expressed as:

$$\bar{\boldsymbol{x}}_t = E[\boldsymbol{x}_t] = \int \boldsymbol{x}_t p(\boldsymbol{x}_t|\boldsymbol{z}_t) = \sum_{i=1}^{N_s} w_t^i \boldsymbol{x}_t^i. \tag{6.3}$$

The objective is to choose effective particles $\{\boldsymbol{x}_t^i\}$ from the distributed sensor nodes and calculate associate weights $\{w_t^i\}$ to estimate $\boldsymbol{x}_t$ based on equation (6.1), (6.2), (6.3) via a distributed protocol. If only part of the particles with associate weights are propagated throughout the network for the estimation, some information is lost. Therefore, the distributed implementa-

tions can not achieve the same accuracy as centralized PF. However, if the propagated particles are increased, the communication overhead is increased accordingly. Thus, the design of the distributed protocol should consider the tradeoff between the estimation accuracy and the communication overhead.

6.3 Selective Gossiping Algorithm

Gossip algorithm is an iterative distributed algorithm for in-network processing, where each node maintains a local estimation vector and exchanges its estimation with neighborhoods then updates its local estimations. Usually, the updated estimation is the average value of two nodes. The goal is to reach a consensus, in which all the sensor nodes agree with the same estimated vector. Since the algorithm does not require any specialized routing or topology, it is robust to unreliable wireless network conditions and link or node failure [105].

Selective gossip only transmits and updates the significant elements in the vector iteratively according to a given threshold [136, 137]. Initially, each node maintains a local estimated vector. At the kth iteration, a node j wakes up according to an independent Poisson clock and randomly chooses a neighbor node s. Then node s and m only gossip the significant element and update the average value. The insignificant element, which is lower than the threshold, is not transmitted. Based on this procedure, the estimation vector converges asymptotically. If the original dimension of the estimated vector is very large, selective gossip algorithm effectively reduces the communication cost. Top K-selective gossip extends the usage of the selective gossip, which only selects the K biggest element to share with neighbors. In this case, each node only propagate the particles with K biggest weights throughout the network. The communication cost is further reduced and also the consensus is guaranteed.

Recently, Jie Lu $et\ al.$ proposed pairwise equalizing gossip algorithm to solve distributed convex optimization problems [138]. In pairwise equalizing gossip algorithm, two neighboring nodes exchange both of their local estimations and the local function parameters to update the new estimations. It has been proved that, with sufficient gossip iterations, the whole network can converge to a global optimization asymptotically.

The pariwise selective gossip (PSG) algorithm integrates selective gossiping and pairwise equalizing gossiping to estimate the state in a decentralized way. The PSG obtains the advantage of selective gossiping which has low communication overhead, and the advantage of pairwise equalizing gossiping for its optimal consensus.

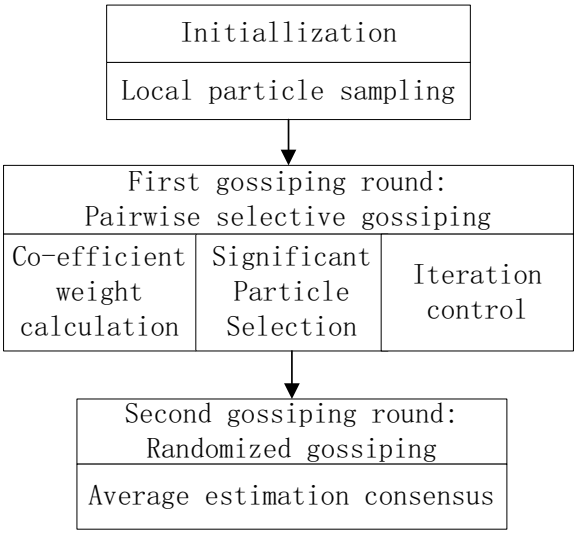


Figure 6.2: Algorithm structure of PSG-DPF

6.4 PSG-DPF Algorithm

Our goal is to construct a DPF in which every node can reach consensus of an estimation. To reach our goal, two rounds of gossip scheme is employed. Initially, each sensor node runs a local particle filter and generates particles independently according to previous state and local information, which is defined as local particle sampling stage. In the first gossiping round, pairwise selective gossiping is used to share particles among nodes and calculate co-efficient weights instead of local weights, and then the algorithm selects the significant particles with maximum co-efficient weights. It ensures every node obtaining particles with global significant weights although consensus is not reached yet. To reduce the communication cost, gossiping iterations are controlled in this round. In the second gossiping round, each node gossips its estimation with neighborhoods and updates the average value until all nodes achieve consensus. The basic algorithm structure is illustrated in Fig. 6.2.

6.4.1 Local Particle Sampling

Initially, we choose local importance sampling scheme, in which every node generates N_s particles $\{\boldsymbol{x}_0^i\}_{i=1}^{N_s}$ according to their local importance density functions $q(\boldsymbol{x}_0^i)$ and local measurement z_0^j at time step $t = 0$, where j denotes the node number. Then the sensor node calculates the local associate weights w_0^i as:

$$w_0^i = \frac{p(\boldsymbol{x}_0^i|z_0^j)}{\sum_{i=1}^{N_s} p(\boldsymbol{x}_0^i|z_0^j)}, \tag{6.4}$$

where $p(\boldsymbol{x}_0^i|z_0^j)$ can be expressed as:

$$p(\boldsymbol{x}_0^i|z_0^j) = \frac{p(z_0^j|\boldsymbol{x}_0^i)p(\boldsymbol{x}_0^i)}{I_s(\boldsymbol{x}_0^i)}. \tag{6.5}$$

Then, sensor nodes choose the significant particles according to the local weights to share with neighbors. If the number of significant particles is smaller than N_s, the communication overhead is effectively reduced. In the future time t, the significant particles will be propagated based on $I_s(\boldsymbol{x}_t^i|\boldsymbol{x}_{t-1}^i)$.

6.4.2 Coefficient Weights Calculation

After obtaining the local weights, each sensor node, $e.g.$ node j, selects several particles with the most significant weights to construct a new particle set N_j and drops the others. The number of particles in N_j is defined as $|N_j|$, where $|N_j| < N_s$. Then, sensor nodes start gossiping their particles with neighborhoods. Based on a Poisson clock, sensor node j wakes up and chooses a neighbor m randomly. This two nodes firstly share their candidate particles and then form a new set $\boldsymbol{x}_t^i \in N_{j,m} = N_j \bigcup N_m$, in which $N_j \bigcap N_m \neq \emptyset$. After that, each node calculates its local likelihood $p(z_t^j|\boldsymbol{x}_t^i)$ or $p(z_t^m|\boldsymbol{x}_t^i)$ for every particle in N_j and exchanges the likelihood values with each other for updating. To ensure two nodes choosing the same particles, co-efficient weights are calculated as follows:

$$w_t^i = \frac{p(z_t^j|\boldsymbol{x}_t^i)p(z_t^m|\boldsymbol{x}_t^i)}{\sum_{i\in N_{j,m}} p(z_t^j|\boldsymbol{x}_t^i)p(z_t^m|\boldsymbol{x}_t^i)}. \tag{6.6}$$

Finally, the pair of gossiping nodes update their particles by selecting $|N_j|$ particles with maximum co-efficient weights w_t^i. After updating the particles, sensor node randomly chooses other neighbors to run the same procedure as stated above.

6.4.3 Significant Particle Selection

In this part, we will prove that PSG has the same performance as resampling stage, which does not need addition resampling schemes like other particle filters. Resampling regenerates the particles by dropping the insignificant values and assigns the equal weights $w_t^i = \frac{1}{N_s}$ to particles [56]. Then the estimation $\boldsymbol{x}_t$ can be represented as:

$$\boldsymbol{x}_t = \sum_{i=1}^{N_s} \frac{1}{N_s} \boldsymbol{x}_t^i. \tag{6.7}$$

In pariwise selective gossip distributed particle filter (PSG-DPF), the globe significant particles are distributed in several local particles sets in sensor nodes, $e.g.$ $N_j, j \in \mathcal{V}_t$. If the number of particles $|N_j|$ is same in every node, then the total number of particles is $N_s = n|N_j|$. If the

particles weights in every node are assigned with $\frac{1}{|N_j|}$, the average estimation of all nodes can be expressed as:

$$\bar{\boldsymbol{x}}_t = \frac{1}{n}\sum_{j\in\mathcal{V}_t}\sum_{i\in N_j}\frac{1}{|N_j|}\boldsymbol{x}_t^i = \frac{1}{n}\sum_{N_j\in\mathcal{R}_n}\bar{\boldsymbol{x}}_{t,j}, \tag{6.8}$$

where $\bar{\boldsymbol{x}}_{t,j}$ is the local estimation in node j at time t. As $N_s = n|N_j|$, $\bar{\boldsymbol{x}}_t$ in equation (6.8) is equal to $\boldsymbol{x}_t$ in equation (6.7), which indicates that the state $\boldsymbol{x}_t$ can be expressed as the average estimation of all sensor nodes. Therefore, our scheme performs the same procedure as the resampling stage in the centralized PF during the particle selection in the PSG and the estimation problem can be converted to an average consensus problem.

6.4.4 Iteration Control

If node j has gossiped with all its neighbors, it almost obtains all the significant particles. Therefore, it is unnecessary to run the selective gossip algorithm for many iterations until the nodes reach consensus. We stop gossiping when the node has gossiped with all neighborhoods, which can further reduce communication overhead.

6.4.5 Estimation Consensus

To achieve consensus, estimations in the sensor nodes, such as $\bar{\boldsymbol{x}}_{t,j}$, are exchanged in the second round of randomized gossiping. Based on the proof in [106], when the iteration $t \to \infty$, all the values converge asymptotically, which is denoted:

$$\bar{\boldsymbol{x}}_{t,j} \to \bar{\boldsymbol{x}}_t = \frac{1}{n}\sum_{j=1}^{n}\bar{\boldsymbol{x}}_{t,j}. \tag{6.9}$$

Thus we obtain the final estimation $\bar{\boldsymbol{x}}_t$ for each sensor node. In practical implementation, we set a threshold τ to indicate consensus. Nodes reach consensus when $|\bar{\boldsymbol{x}}_{t,j} - \bar{\boldsymbol{x}}_{t,m}| < \tau$. The whole procedure of PSG-DPF is illustrated in Alg. 7

6.5 K-selective Gossip DPF

The PSG-DPF can not guarantee the consensus in the first round of gossiping, thus a second round of gossiping is required, which increases the gossiping iterations and the communication overhead. To achieve consensus, the K-selective gossip algorithm can be used. The K-selective gossip only chooses a few particles with the number of k and shares them among nodes.

6.5.1 Local Sampling

Initially, each node generate a random set of particles $\{\boldsymbol{x}_t^{i,j}\}_{i=1}^{N_s}$ in the playing field just like PSG-DPF. Since the particles are randomly sampled, the particle sets of sensors are not identical. The local weights are firstly calculated according to the local likelihood function. Then, the local

Algorithm 7 PSG-DPF

// at time t
for node $j \in \{1, \ldots, n\}$ **do**
  Sample: $N_j = \{\boldsymbol{x}_t^i \sim I(\boldsymbol{x}_t^i | \boldsymbol{x}_{t-1}^i)\}$;
  Find neighbors: $E_t[j] = \{m | \{j, m\} \in \mathcal{E}_t\}$
  // Pairwise Selective Gossip
  while $E_t[j] \neq \emptyset$ **do**
    Randomly choose neighborhoods
    $m \in E_t[j]$, set $E_t[j] = E_t[j] - m$;
    Share particles: $\{\boldsymbol{x}_t^i\}_{i=1}^{N_j} \Leftrightarrow \{\boldsymbol{x}_t^i\}_{i=1}^{N_m}$;
    New particles sets: $N_{j,m} = \{\boldsymbol{x}_t^i\}_{i=1}^{N_j} \bigcup \{\boldsymbol{x}_t^i\}_{i=1}^{N_m}$;
    // calculate weights
    Local likelihood: $p(z_t^j | \boldsymbol{x}_t^i)$, $z_t^j \in \{z_t^j\}_{j=1}^n$;
    Share weights:
    $\{w_t^i \propto p(z_t^j | \boldsymbol{x}_t^i)\}_{i \in N_{j,m}} \Leftrightarrow \{w_t^i \propto p(z_t^m | \boldsymbol{x}_t^i)\}_{i \in N_{j,m}}$;
    Co-efficient weights: equation (6.6)
    Update and resampling: select significant particles
  end while
  // Obtain estimation
  $\bar{\boldsymbol{x}}_{t,j} = \frac{1}{|N_j|} \sum_{i=1}^{|N_j|} \boldsymbol{x}_{t,j}^i$;
  // Average consensus
  Random Gossip: $\bar{\boldsymbol{x}}_{t,j} \rightarrow \bar{\boldsymbol{x}}_t$;
end for

weights are turned into log-likelihood function $\{\log w_t^{i,j}\}_{i=1}^k$, which are stored in the node:

$$w_t^{i,j} \propto p(z_t^j | \boldsymbol{x}_t^{i,j}) \tag{6.10}$$

In this case, every node obtains a local posterior PDF of the target's location. The rest problem is to make the nodes reach a consensus estimation.

6.5.2 Coefficient Weight

We employ top-K selective gossip algorithm for sharing the particles with associate log-likelihood weights because only part of the particles are significant in particle filter. Take a pair of neighborhood nodes, which are j and $j+1$, for instance. In the first iteration, node j chooses k particles $\{\boldsymbol{x}_t^{i,j}\}_{i=1}^k$ with highest local weights to form a new particle set. Then these weights are turned into log-likelihood values $\{\log w_t^{i,j}\}_{i=1}^k$ and shared with neighborhood $j+1$ together with the particles. Every node only shares particles with its neighborhood and does not communicate with the node with two hops or further, which can reduce the communication cost. The data is transmitted through a randomly chosen idle channel which is detected during the sensing period.

When the particles are received, the neighbor sensor $j+1$ stores these particles and local weights of j, and then re-calculated the associate local weights of these particles based on its own measurement value. Thus the particles from j to $j+1$ are classified into two cases:

- If the particles from j already exist in $j+1$, $j+1$ just sends its local associate log-likelihood weights $\{\log \grave{w}_t^{i,j}\}_{i=1}^k$ back to node j. Here, we use $\grave{w}_t^{i,j}$ to denote the local weight based on node $j+1$ of the particle which belongs to sensor j, which is calculated as $\grave{w}_t^{i,j} \propto p(z_t^{j+1} | \boldsymbol{x}_t^{i,j})$
- If the particles from j do not exist in node $j+1$, $j+1$ accepted the new particles. Then it calculates the local weight for these particles, which is $\log \grave{w}_t^{i,j}$ and sends back to j.

Then, two nodes update the log-likelihood values $\log \bar{w}_t^{i,j} = \frac{1}{2}(\log w_t^{i,j} + \log \grave{w}_t^{i,j})$. In this case, the associated weight calculation for each particle from j reaches consensus between the two nodes. If node $j+1$ sends its own particles $\{\boldsymbol{x}_t^{i,j+1}\}_{i=1}^k$ with weights $\{\log w_t^{i,j+1}\}_{i=1}^k$ to node j and j follows the same procedure, then a new identical particle set with the same weights are obtained, which is $\{\boldsymbol{x}_t^{i,j}\}_{i=1}^k \bigcup \{\boldsymbol{x}_t^{i,j+1}\}_{i=1}^k$. Then, both nodes choose the k highest particles and update the particle set. In this case, both particles and weights achieve consensus for the nodes.

For a single particle with significant weight, the log-likelihood weight calculation is based on the averaging gossip procedure, therefore, consensus is reached gradually. If the particle is not important to the most of nodes, it will be dropped. According to the conclusion of K-selective, every node chooses significant particles based on the consensus weights, then the identical particle set is attained finally. Since only significant particles are transmitted and chosen, and the

insignificant particles are dropped during gossiping, the communication procedure of distributed particle filter is acting resampling step gradually. Therefore, no additional resampling is required. Finally, the global estimation is attained $\bar{\boldsymbol{x}}_t = \sum_{i=1}^{k} w_t^i \boldsymbol{x}_t^i$, where $\{\boldsymbol{x}_t^i, w_t^i\}_{i=1}^k$ is the global consensus particle set. The whole algorithm is presented in Alg. 8

Algorithm 8 K-Selective-DPF

// at time t
for Node $j \in \{1, \ldots, N\}$ **do**
  Sample: $\{\boldsymbol{x}_t^{i,j}\}_{i=1}^{N_s}\}$;
  Calculate local weights: $w_t^{i,j} \propto p(z_t^j | \boldsymbol{x}_t^{i,j})$
  Choose k significant particles $\{\boldsymbol{x}_t^{i,j}\}_{i=1}^k$
  // Top K-Selective Gossip
  while Localization procedure **do**
    Randomly choose a neighborhood $j+1$
    Share particles: $\{\boldsymbol{x}_t^{i,j}\}_{i=1}^k \Leftrightarrow \{\boldsymbol{x}_t^{i,j+1}\}_{i=1}^k$;
    New particles sets: $\{\boldsymbol{x}_t^{i,j}\}_{i=1}^k \bigcup \{\boldsymbol{x}_t^{i,j+1}\}_{i=1}^k$;
    // calculate weights
    Local likelihood: $\grave{w}_t^{i,j} \propto p(z_t^{j+1} | \boldsymbol{x}_t^{i,j})$;
    Co-efficient weights: $\log \bar{w}_t^{i,j} = \frac{1}{2}(\log w_t^{i,j} + \log \grave{w}_t^{i,j})$
    Update and choose a new k significant particles
  end while
  // Obtain estimation
  $\bar{\boldsymbol{x}}_{t,j} = \frac{1}{k} \sum_{i=1}^{k} \boldsymbol{x}_t^i$;
end for

6.6 Simulation and Experiment

6.6.1 Simulation

The proposed schemes are evaluated in two simulations of target tracking. In our first simulation, we randomly deploy 30 sensors in a unitary two-dimensional square $100m \times 100m$ region. Nodes within the square region can communicate with each other. One target runs through a rectangle path in anticlockwise direction with a constant speed $0.5m/s$. The starting coordination is $(2, 2)$.

The results of the simulation are averaged by 1000 Monte-Carlo trials. The sensor nodes are randomly deployed in every trial. Sensor node j is assigned coordinations $[a_j^X, a_j^Y]^T$ and the target position state at time t is $\boldsymbol{x}_t = [p_t^X, p_t^Y]^T$. The nodes can measure the distances between the target and communicate with neighborhood nodes. The measurement period is 1 second. It is assumed that the measurement noise is additive Gaussian noise.

In this simulation, four algorithms are compared: the first one is a centralized SIR particle

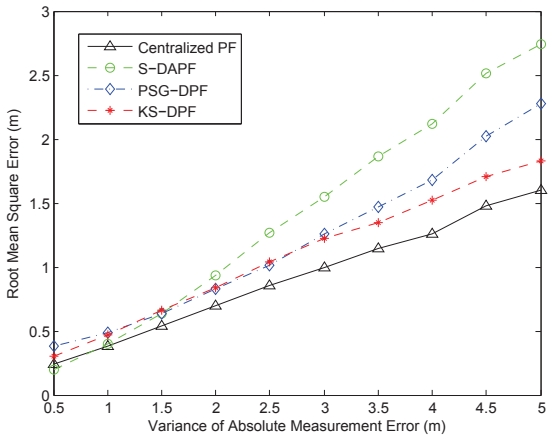


Figure 6.3: Estimation errors for four particle filters by adapting standard derivation of measurement noise.

filter algorithm, named centralized PF, which employs the A-BPF mentioned in the previous chapters; the second one is the proposed scheme, named PSG-DPF; the third one is K-selective particle filter (KS-DPF); and the last one is a distributed auxiliary particle filter based on selective gossip proposed by Deniz $et\ al.$, named selective distributed auxiliry particle filter (S-DAPF), in which weights are calculated according to local likelihoods, $w_t^i \propto w_{t-1}^i p(z_t^j|\boldsymbol{x}_t^i)$ [113]. In this simulation, the centralized PF generates 1000 particles for each iteration, while other algorithms run 200 particles for each node in each iteration.

Figure 6.3 illustrates the RMSE of estimated position of four algorithms. We adapt the standard derivation of measurement noise σ for each nodes from 0 to 5 meters to compare the performance of three algorithms. When the measurement noise is low, such as σ ranges from 0 to 0.5m, the four algorithms have almost the same accuracy. When the measurement noise increases, the centralized PF performs with the lowest RMSE, and PSG-DPF has better accuracy than S-DAPF. This indicates that particle selection based on co-efficient weights can converge to a more precise estimation than local weights based selection in S-DAPF. The KS-DPF achieves the most accurate estimation.

We also evaluate the performance improvement with adapting the number of gossiping particles for the distributed PFs. The centralized PF with fixed number of particles is also compared in this simulation. The number of significant particles changes from 20 to 200 and we compare the accuracy of the DPFs. We also run 1000 Monte-Carlo trials. As shown in Fig. 6.4, RMSE of the KS-DPF is decreasing with rising the number of particles. For S-DAPF, increasing particles can not improve too much. While for PSG-DPF, the performance is also stable when adjusting the number of particles but more accurate than S-DAPF.

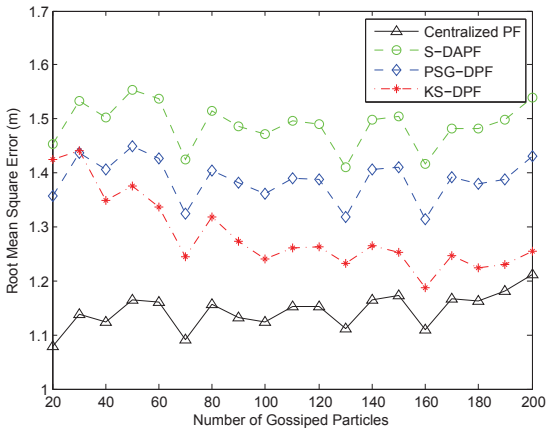


Figure 6.4: Estimation errors by adapting number of gossiping particles.

6.6.2 Indoor Localization Experiment

The proposed algorithms are also evaluated in the real indoor localization. The data used in this experiment is from the test-bed mentioned in the previous chapter 5. In this experiment, nodes with measurements which are lower than 0 or exceeds 15 meters are failure nodes, which can not join the network. In our simulation, the nodes within radius of 15 meters are connected which can gossip with each other. According to our statistical results, σ is 3 meters and the average of the measurement error μ is 1 meter, thus we set our measurement noise follows Gaussian distribution $v_t \sim \mathcal{N}(1, 3^2)$ in our simulation.

We also implement the four algorithms mentioned above to compare their performance. For the centralized PF, we use 1000 particles to estimate the positions. For the S-DAPF, the PSG-DPF and the KS-DPF, 40 particles are generated and gossiped in each node.

Figure 6.5 illustrates the communication overhead for the PSG-DPF at each time step during the simulation. The communication overhead is evaluated as the gossiping iterations of the participated sensor node. The number of gossiping iterations in the S-DAPF and the KS-DPF is set as 100. In PSG-DPF, the participated node just needs to gossip with every neighbor once in the first round gossiping. As Fig. 6.5 shows, the maximum gossip iterations is 12 which includes 6 iterations in the first round and 6 iterations in the second round of gossiping. And the minimum gossip iterations is 2, which means that only two node participated in the estimation, and the two nodes only gossiped once in both first and second round of gossiping. Based on our iteration control scheme, each sensor node almost achieves consensus about the significant particles, thus, sensor nodes can converge to an estimation quite fast in the second round. Although we can set fixed gossiping iterations of the PSG-DPF, it is unnecessary as the particles with coefficient weights converge fast to achieve the same accuracy as the S-DAPF. Therefore, the PSG-DPF effectively reduces communication cost by controlling gossiping iterations. According to

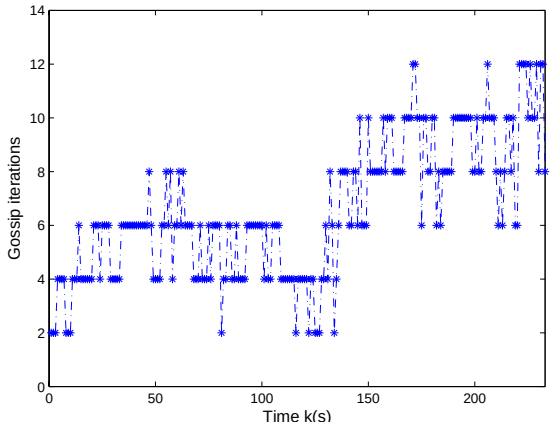


Figure 6.5: Gossip iterations of the PSG-DPF

Table 6.1: Estimation Performance Comparison for the Algorithms

Algorithm	Average	Median	RMSE	Min	Max
Centralized PF	1.0946	0.9968	1.2789	0.0463	3.6174
S-DAPF	1.9669	1.6839	2.3459	0.0549	8.1893
PSG-DPF	1.2834	1.1955	1.4680	0.0410	4.5838
KS-DPF	1.2509	1.1150	1.4429	0.0402	4.1859

Fig. 6.5, the KS-DPF can also apply the iteration control scheme. However, the network scale information is not necessary for PSG-DPF, but is required for the KS-DPF, which uses additional topology management scheme. Thus, for a small dynamic network, the PSG-DPF outperforms the KS-DPF.

The transmitted particles are the same for the DPFs, thus the data amount for one gossiping iteration is almost the same. However the DPFs employ different communication procedure to achieve the final estimation, which leads to different overall communication. For the S-DAPF, it requires two round of weight calculation. Thus, two rounds of gossiping are designed. For each round of gossiping, it contains both selective gossip and max gossip to achieve consensus, which are also two rounds of gossiping. Besides, the S-DAPF only obtains significant particles with maximum local weights instead of global weights, which leads to loss of information. The PSG-DPF also requires two rounds of pairwise gossiping. In the first round of gossiping, nodes choose particles based on coefficient weights between two neighborhood nodes. Thus, the weight calculation is still locally. Besides, nodes can not achieve consensus, which require the second round of averaging gossiping to achieve global estimation. The KS-DPF only employ one gossiping round. Since the weight value are transmitted together with the associate particle, the consensus global weight is obtained without using another round of gossiping. Thus, the KS-DPF requires less communication overhead than the S-DAPF and the PSG-DPF in a general

large network.

6.7 Summary

Reducing communication cost among wireless nodes without sacrificing estimation accuracy is a major concern for distributed particle filter implementations. The novel distributed particle filters are proposed. The first one is the DPF by using a pairwise selective gossiping, PSG-DPF and the second DPF uses the top K-selective gossiping. Both of the algorithms can offer four advantages as follows:

1. The nodes can share particles to calculate co-efficient weights rather than local weights which can achieve global optimal consensus more efficiently and faster than other solutions;
2. No additional resampling stage is required as selective gossiping can automatically choose the significant particles and reduce the communication overhead;
3. Since every node obtains a globally significant particle set, which makes the estimation in every node converge fast;
4. Communication overhead is reduced by controlling the gossiping iterations.

Experimental results demonstrate the performance of our algorithms. According to the performance comparison, the PSG-DPF requires 2 rounds of gossiping and weight calculation is calculated locally. Thus, it is suitable to implemented in a small scale full connected network, where the local estimation is almost the same as the global estimation. For KS-DPF, it is more suitable for a large scale network which requires the global consensus estimation. In general, KS-DPF is almost equivalent to the centralized PF and $0.1m$ better than PSG-DPF.

Chapter 7

Cramér-Rao Bound Analysis

7.1 Introduction

The estimation accuracy is a major concern for all the localization algorithms. The adaptive particle filters are demonstrated their promising estimation performance in the previous chapters. In addition to the performance evaluation metrics in chapter 5, the researchers also attempt to derive the estimation accuracy in a theoretical way. One of the fundamental way is to derive the Cramér-Rao lower bound (CRLB) [139, 140]. CRLB explicitly indicates the optimal estimation performance with the given range measurement noise and anchors' positions. When the measurement noise is independent to the anchor positions, the measurement noise matrix in Fisher information matrix can be eliminated and the inverse of the rest part is obtained as the geometrical dilution of precision (GDOP) [141]. Thus, it is used to analyze the optimal system performance and compared with the localization algorithms [130].

PFs are the location estimators based on Bayesian theorem. The Bayesian estimators use a priori state distribution as the priori information and the current measurement data as the update information to obtain the target's location. The estimation performances are much better than other pure geographic localization algorithms, since the priori information involves. However, these methods employ different processes which show different characters during the estimations. Kalman filter or extended Kalman filter uses recursive process, which apply the state distribution parameters iteratively to obtain a new estimated state. Although particle filter claims that it follows the recursive process, the assigned equal weights in the resampling stage and the simplified weight formulation in many location based particle filters make it as a simple Bayesian model, which estimates the PDF just based on the priori distribution and the measurement likelihood. The previous work proposed an adaptive method of particle filter, which fuses the priori distribution into the measurement likelihood function. It is a new heuristic process based on Bayesian model. The estimation performance is better than the conventional particle filters. S-

ince the three processes fuse the information in different ways, the estimation performance can be different as well.

The CRLBs with and without priori information for a mobile target tracking have been investigated, which demonstrates that the priori information can improve the estimation performance [140]. The basic CRLBs are derived in lots of literatures, one problem remains, which is that researchers do not focus on classifying the ways of using priori information in the previous work [140, 27, 8, 139, 142, **?**]. Due to different processes, the formulation of Fisher information matrix can be quite different. Thus, CRLBs of different process are also different.

In this chapter, the CRLBs are used to model the behavior of the Bayesian estimators, which can present a theoretical performance of the particle filters. Firstly, the Bayesian estimators are classified into three types of processes: the first one is the fundamental Bayesian process, which is like particle filter; the second type is recursive process, which is followed by Kalman filter and extended Kalman filter; the third type is the adaptive process, which is based on the adaptive particle filter. The Fisher information matrixes are formulated for the three types respectively. It is demonstrated that although the prior information can improve the location estimation accuracy, using it in different ways can lead to different performance. According to the comparison, the system can choose the best Bayesian estimator for a typical environment.

Besides the analysis of the Bayesian estimators, the CRLBs also indicate one realistic problem which is hard to analyze in the real system. The difference height between the anchors and the target can mislead the ranging measurement and further degrade the estimation accuracy. The CRLBs formulate the problem in this chapter from a 3D perspective. It is also indicated that, without considering the relative height, the location algorithms based on CRLB and GDOP are unreliable and can not achieve to the optimal performance.

7.2 Cramér-Rao Lower Bound

The CRLB as the optimal performance indicator for unbiased estimator is widely applied in the localization and positioning systems. Theoretical investigations have been researched as the nonlinear problem for wireless localization systems. Tichavsky et al. provides the formulation of recursive posterior CRLB for nonlinear filters based on Bayesian framework [143]. Zuo et al. proposed a conditional CRLB which considered the posterior probability is conditioned on the priori probability [144]. The CRLB for distributed systems is analyzed by Mohammadi $et\ al.$ [145].

For range based wireless localization system, many researches have provided CRLB results for different scenarios. Qi provides the CRLB of wireless system for non-line-of-sight (NLOS) environment [140]. The hybrid LOS/NLOS environment is analyzed and Qi indicated that with a priori knowledge of wireless transmission channel, the estimation performance can be improved

[140]. Shen et al. extends the CRLB to a general framework of the wide band wireless network. The multipath and NLOS effect are both considered and the CRLB with or without priori information are compared in the formulation [146]. Other similar works also give CRLB for different ranging techniques [147, 22]. Although some other methods can be used for performance analysis [25], CRLB is still popular for wireless localization researchers due to its simplicity and general expression.

Besides the estimation accuracy analysis, the geometric matrix with Fisher information matrix can also be employed to formulate the geometric relationship between the target and anchors. Dulman et al. used CRLB to illustrate the geometric impact for localization algorithms [148, 139]. Similar work is also proposed by Tseng et al. [149]. The geometric formulation in CRLB is also integrated with localization algorithms to obtain a more accurate position, $e.g.$ optimal geometric sensor selection for localization [130]; time-difference-of-arrival (TDOA) based algorithms [142, 21] and signal strength differences based fingerprinting method [150]. Also, new algorithm is developed based on the geometric method, which employs a two stage location method to derive the target's position [151].

With multiple information, some variations of CRLB are proposed to further analyzed the estimation performance. The equivalent Fisher information is applied to derive the location performance from the multipath and NLOS parameters [146]. A linear CRLB is proposed which consider the linearized effect and provided the lower bound for such estimator [149].

Although the priori information is mentioned in the literatures, how the priori information influences the estimation performance is not investigated yet. Thus, our goal is to formulate the priori information and provides the optimal way to use the priori information.

7.3 Fisher Information Matrix Formulation

7.3.1 Bayesian Model Revisit

Since the PFs are based on the Bayesian Model, here, the simplified model for the target tracking is presented again.According to the Bayesian estimation framework, the relationship between the estimated state $\boldsymbol{x}_t$ and the measurement $\boldsymbol{z}_t$ follows:

$$\boldsymbol{x}_t = \boldsymbol{F}_t \boldsymbol{x}_{t-1} + \boldsymbol{q}_t \tag{7.1}$$

$$\boldsymbol{z}_t = \boldsymbol{h}_t(\boldsymbol{x}_t) + \boldsymbol{v}_t \tag{7.2}$$

where (7.1) is the linear prediction function and (7.2) is the measurement function. In (7.1), the target's movement is based on the linear transition function, and q_t is the prediction noise,

which follows normal distribution $\mathcal{N}(0, \boldsymbol{Q}_t)$. In (7.2), $\boldsymbol{z}_t = [z_t^1, \ldots, z_t^N]^T$ is the measurement vector; $\boldsymbol{h}_t() = [h_t^1(), \ldots, h_t^N()]^T$ is the nonlinear observation function; $\boldsymbol{v}_t = [v_t^1, \ldots, v_t^N]^T$ is the ranging noise, which is assume as independent noise.

7.3.2 Fisher Information Matrix

CRLB, which is given by the inverse of the Fisher information matrix (FIM), sets the lower limit for the variance (or covariance matrix) of any unbiased estimators of an unknown parameter (or unknown parameters) [152]. If $p(\boldsymbol{x}_t, \boldsymbol{z}_t)$ denotes the joint PDF of observations $\boldsymbol{z}_t$ and the state $\boldsymbol{x}_t$, then the score function is defined as the gradient of its log-likelihood:

$$U(\boldsymbol{x}_t) = \nabla \ln p(\boldsymbol{x}_t, \boldsymbol{z}_t) = \frac{\partial \ln p(\boldsymbol{x}_t, \boldsymbol{z}_t)}{\partial \boldsymbol{x}_t} \tag{7.3}$$

The FIM, $\boldsymbol{I}(\boldsymbol{x}_t)$, is the variance of the score function:

$$\boldsymbol{I}(\boldsymbol{x}_t) = E\left\{ \frac{\partial \ln p(\boldsymbol{x}_t, \boldsymbol{z}_t)}{\partial x_t} \left[\frac{\partial \ln p(\boldsymbol{x}_t, \boldsymbol{z}_t)}{\partial x_t} \right]^T \right\}. \tag{7.4}$$

And CRLB is just the inverse of FIM, and the estimation covariance can not be lower than it:

$$\boldsymbol{Cov}_{\boldsymbol{x}_t}(\tilde{\boldsymbol{x}}_t) \geq \{\boldsymbol{I}(\boldsymbol{x}_t)\}^{-1} \tag{7.5}$$

Since $p(\boldsymbol{x}_t, \boldsymbol{z}_t) = p(\boldsymbol{z}_t|\boldsymbol{x}_t)p(\boldsymbol{x}_t)$ based on the Bayesian theorem, it is easily seen that $\boldsymbol{I}(\boldsymbol{x}_t)$ can be decomposed into:

$$\boldsymbol{I}(\boldsymbol{x}_t) = \boldsymbol{I}_D(\boldsymbol{x}_t) + \boldsymbol{I}_d(\boldsymbol{x}_t)\boldsymbol{I}_p^T(\boldsymbol{x}_t) + \boldsymbol{I}_d^T(\boldsymbol{x}_t)\boldsymbol{I}_p(\boldsymbol{x}_t)\boldsymbol{I}_P(\boldsymbol{x}_t) \tag{7.6}$$

where

$$\begin{cases} \boldsymbol{I}_D(\boldsymbol{x}_t) = E\left\{ \frac{\partial \ln p(\boldsymbol{z}_t)|\boldsymbol{x_t}}{\partial \boldsymbol{x_t}} \left[\frac{\partial \ln p(\boldsymbol{z}_t)|\boldsymbol{x_t}}{\partial \boldsymbol{x_t}} \right]^T \right\} \\ \boldsymbol{I}_d(\boldsymbol{x}_t) = E\left\{ \frac{\partial \ln p(\boldsymbol{z}_t)|\boldsymbol{x_t}}{\partial \boldsymbol{x_t}} \right\} \\ \boldsymbol{I}_p(\boldsymbol{x}_t) = E\left\{ \frac{\partial \ln p(\boldsymbol{x}_t)}{\partial \boldsymbol{x_t}} \right\} \\ \boldsymbol{I}_P(\boldsymbol{x}_t) = E\left\{ \frac{\partial \ln p(\boldsymbol{x}_t)}{\partial \boldsymbol{x_t}} \left[\frac{\partial \ln p(\boldsymbol{x}_t)}{\partial \boldsymbol{x_t}} \right]^T \right\} \end{cases} \tag{7.7}$$

The expectation of score is 0 [115], then $\boldsymbol{I}_d(\boldsymbol{x}_t) = \boldsymbol{I}_p(\boldsymbol{x}_t) = 0$. Then, $\boldsymbol{I}(\boldsymbol{x}_t)$ is decomposed into two parts:

$$\boldsymbol{I}(\boldsymbol{x}_t) = \boldsymbol{I}_D(\boldsymbol{x}_t) + \boldsymbol{I}_P(\boldsymbol{x}_t) \tag{7.8}$$

where $\boldsymbol{I}_D(\boldsymbol{x}_t)$ represents the information obtained from measurement data, and $\boldsymbol{I}_P(\boldsymbol{x}_t)$ represents the priori information.

The range measurement error distributions are different for wireless system in outdoor and

indoor environment. For outdoor localization, the error can be assumed as the Gaussian distribution. For indoor localization, it is better to use Gamma distribution [140]. With the additive Gaussian distribution, $\boldsymbol{I}_D(\boldsymbol{x}_t)$ has been given as follows:

Lemma 1. *CRLB for the Gaussian distribution follows: the mean square error of the unbiased position estimator $Cov_D(\tilde{x}_t) \geq \boldsymbol{I}_D^{-1}(\boldsymbol{x}_t)$, where $I_D(\boldsymbol{x}_t)$ is the FIM, which follows:*

$$\boldsymbol{I}_D(\boldsymbol{x}_t) = \begin{pmatrix} \sum_{j=1}^{n} \frac{1}{R_j} (\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^X})^2 & \sum_{j=1}^{n} \frac{1}{R_j} \frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^X} \frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^Y} \\ \sum_{j=1}^{n} \frac{1}{R_j} \frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^X} \frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^Y} & \sum_{j=1}^{n} \frac{1}{R_j} (\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^Y})^2 \end{pmatrix} \tag{7.9}$$

It is assumed that the measurement noise for ranging follows Gaussian distribution $v_t^j \sim \mathcal{N}(0, R_j)$; the measurement noise for each anchor are conditional independent; $\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^X}$ and $\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^Y}$ are:

$$\begin{cases} \frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^X} = \frac{p_t^X - a_j^X}{\sqrt{(p_t^X - a_j^X)^2 + (p_t^Y - a_j^Y)^2}} \\ \frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^Y} = \frac{p_t^Y - a_j^Y}{\sqrt{(p_t^X - a_j^X)^2 + (p_t^Y - a_j^Y)^2}} \end{cases} \tag{7.10}$$

Proof. For each anchor j, the range measurement is:

$$z_t^j = h_t^j(\boldsymbol{x}_t) + v_t^j = \sqrt{(p_t^X - a_j^X)^2 + (p_t^Y - a_j^Y)^2} + v_t^j \tag{7.11}$$

where $v_t^j \sim \mathcal{N}(0, R_j)$ is independent to other nodes. Then, we construct the measurement function vector:

$$\boldsymbol{h}_t(\boldsymbol{x}_t) \triangleq [h_t^1(\boldsymbol{x}_t), ..., h_t^j(\boldsymbol{x}_t), ..., h_t^n(\boldsymbol{x}_t)]^T \tag{7.12}$$

and the measurement noise covariance matrix is formulated as:

$$\boldsymbol{R}_t = diag(R_1, ..., R_j, ..., R_N) \tag{7.13}$$

So the observed measurement vector $\boldsymbol{z}_t$ is:

$$\begin{aligned} \boldsymbol{z}_t &= \boldsymbol{h}_t(\boldsymbol{x}_t) + \boldsymbol{v}_t \\ &= \begin{pmatrix} h_t^1(\boldsymbol{x}_t) \\ \vdots \\ h_t^j(\boldsymbol{x}_t) \\ \vdots \\ h_t^n(\boldsymbol{x}_t) \end{pmatrix} + \begin{pmatrix} v_t^1 \\ \vdots \\ v_t^j \\ \vdots \\ v_t^n \end{pmatrix} \end{aligned} \tag{7.14}$$

And the likelihood function is calculated as:

$$p(\boldsymbol{z}_t|\boldsymbol{x}_t) = \frac{1}{(2\pi)^{\frac{1}{2}}\det(\boldsymbol{R})^{\frac{1}{2}}} e^{-\frac{1}{2}[\boldsymbol{z}_t-\boldsymbol{h}_t(\boldsymbol{x}_t)]^T\boldsymbol{R}^{-1}[\boldsymbol{z}_t-\boldsymbol{h}_t(\boldsymbol{x}_t)]} \tag{7.15}$$

Accordingly, the likelihood information is:

$$-\log p(\boldsymbol{z}_t|\boldsymbol{x}_t) = c_l + \frac{1}{2}[\boldsymbol{z}_t - \boldsymbol{h}_t(\boldsymbol{x}_t)]^T \cdot \boldsymbol{R}^{-1} \cdot [\boldsymbol{z}_t - \boldsymbol{h}_t(\boldsymbol{x}_t)] \tag{7.16}$$

where c_l is a constant value. Submit (7.16) into (7.4), Fisher information matrix is formulated as:

$$\boldsymbol{I}_D = E\{[\partial\frac{\boldsymbol{h}_t^T(\boldsymbol{x}_t)}{\partial\boldsymbol{x}_t}] \cdot \boldsymbol{R}^{-1} \cdot [\partial\frac{\boldsymbol{h}_t^T(\boldsymbol{x}_t)}{\partial\boldsymbol{x}_t}]^T\} \tag{7.17}$$

Then, (7.9) is obtained, and *Lemma 1* is proofed. □

This proof is a general CRLB formulation, the same expression of (7.9) can be found in the equation (2.173) of H.So et al. work [153]. In *Lemma 1*, the range measurement noise v_t^j is assumed to be normal distribution. But in indoor localization, the non-line-of-sight and multipath effect strongly influence the wireless signal transmission. As stated in the previous chapters before, the Gamma distribution is suitable for modeling the noise[71, 119, 120]. Thus, the FIM for Gamma distribution case is expressed as:

Lemma 2. *If the range measurement noise follows Erlang distribution* $n_t^j \sim \Gamma(\alpha,\beta)(\theta) = \frac{\beta^\alpha}{\Gamma(\alpha)}\theta^{\alpha-1}e^{-\beta\theta}$*, where* $\alpha \in \mathbb{N}$ *and the Gamma function is:*

$$\Gamma(\alpha) = (\alpha-1)! \tag{7.18}$$

Then, the Fisher information matrix $\boldsymbol{I}_D(\boldsymbol{x}_t)$ *is expressed as:*

For $\alpha^j = 1$*:*

$$\boldsymbol{I}_D = \begin{pmatrix} I_{11} & I_{12} \\ I_{21} & I_{22} \end{pmatrix} \tag{7.19}$$

where each component is:

$$\begin{cases} I_{11} = \sum_{j=1}^{N}(\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^X})^2(\beta^j)^2 + \sum_{j\neq i}^{N}\beta^j\beta^i\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^X}\frac{\partial h_t^i(\boldsymbol{x}_t)}{\partial p_t^X} \\ I_{12} = I_{21} = \sum_{j=1}^{N}\sum_{i=1}^{N}\beta^j\beta^i\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^X}\frac{\partial h_t^i(\boldsymbol{x}_t)}{\partial p_t^Y} \\ I_{22} = \sum_{j=1}^{N}(\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^Y})^2(\beta^j)^2 + \sum_{j\neq i}^{N}\beta^j\beta^i\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^Y}\frac{\partial h_t^i(\boldsymbol{x}_t)}{\partial p_t^Y} \end{cases} \tag{7.20}$$

For $\alpha_i = 2$*,* $I_{2\times2}$ *is infinite and useless.*

For $\alpha_i > 2$ and $\alpha \in \mathbb{N}$:

$$I_D(\boldsymbol{x}_t) = \begin{pmatrix} \sum_{j=1}^{N} (\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^X})^2 \frac{(\beta^j)^2}{\alpha^j - 2} & \sum_{j=1}^{N} \frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^X} \frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^Y} \frac{(\beta^j)^2}{\alpha^j - 2} \\ \sum_{j=1}^{N} \frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^X} \frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^Y} \frac{(\beta^j)^2}{\alpha^j - 2} & \sum_{j=1}^{N} (\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^Y})^2 \frac{(\beta^j)^2}{\alpha^j - 2} \end{pmatrix} \tag{7.21}$$

Proof. For Erlang distribution, the proof is complicated. For some parameters, it is difficult to derive an analytical expression. If the measurement noise follows Gamma distribution:

$$v_t^j \sim \Gamma(\alpha^j, \beta^j)(x) = \frac{(\beta^j)^{\alpha^j}}{\Gamma(\alpha^j)} x^{\alpha^j - 1} e^{-\beta^j x} \tag{7.22}$$

where, to obtain an analytical expression, it is assumed that $\alpha_j \in \mathbb{N}$ for simplicity, and the Gamma function, in this case, is:

$$\Gamma(\alpha_j) = (\alpha_j - 1)! \tag{7.23}$$

Thus, the likelihood of measurement z_t^j at position $\boldsymbol{x}_t = [p_t^X, p_t^Y]^T$ is

$$\begin{aligned} p(z_t^j | \boldsymbol{x}_t) = &= \Gamma(\alpha^j, \beta^j)(h_t^j(\boldsymbol{x}_t)) \\ &= \frac{\beta_j^{\alpha^j}}{\Gamma(\alpha^j)} h_t^j(\theta)^{\alpha^j - 1} e^{-\beta^j h_t^j(\theta)} \end{aligned} \tag{7.24}$$

If the errors for the range measurements are independent, the joint PDF is:

$$p(\boldsymbol{z}_t | \boldsymbol{x}_t) = \prod_{j=1}^{N} p(z_t^j | \boldsymbol{x}_t) \tag{7.25}$$

Then, the likelihood information is the sum of log functions:

$$\log p(\boldsymbol{z}_t | \boldsymbol{x}_t) = \sum_{j=1}^{N} \log p(z_t^j | \boldsymbol{x}_t) \tag{7.26}$$

Thus, the FIM is constructed as:

$$\begin{aligned} I_D(\boldsymbol{x}_t) &= E(\frac{\partial \log p(z^t | \boldsymbol{x}_t)}{\partial \boldsymbol{x}_t})^2 \\ &= \begin{pmatrix} I_{11} & I_{12} \\ I_{21} & I_{22} \end{pmatrix} \end{aligned} \tag{7.27}$$

where each element is formulated as:

$$\begin{cases} I_{11} = E(\sum_{j=1}^{N} \frac{\partial \log p(z_t^j|\boldsymbol{x}_t)}{\partial p_t^X})^2 \\ I_{12} = I_{21} = E(\sum_{j=1}^{N} \frac{\partial \log p(z_t^j|\boldsymbol{x}_t)}{\partial p_t^X} \sum_{j=1}^{N} \frac{\partial \log p(z_t^j|\boldsymbol{x}_t)}{\partial p_t^Y}) \\ I_{22} = E(\sum_{j=1}^{N} \frac{\partial \log p(z_t^j|\boldsymbol{x}_t)}{\partial p_t^Y})^2 \end{cases} \tag{7.28}$$

Then, each element in this FIM is calculated as follows:

For I_{11}, extend the expression:

$$\begin{aligned} I_{11} &= E(\sum_{j=1}^{N} \frac{\partial \log p(z_t^j|\boldsymbol{x}_t)}{\partial p_t^X})^2 \\ &= \sum_{j=1}^{N} E(\frac{\partial \log p(z_t^j|\boldsymbol{x}_t)}{\partial p_t^X})^2 \\ &+ \sum_{i \neq j}^{N} E(\frac{\partial \log p(z_t^i|\boldsymbol{x}_t)}{\partial p_t^X}) E(\frac{\partial \log p(z_t^j|\boldsymbol{x}_t)}{\partial p_t^X}) \end{aligned} \tag{7.29}$$

where I_{11} contains two parts, the first part is related to the covariance of score functions and the second part relies on the expectation of score functions. The score for each element in (7.29) is:

$$\frac{\partial \log p(z_t^j|\boldsymbol{x}_t)}{\partial p_t^X} = \frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^X} (\frac{\alpha^j - 1}{h_t^j(\boldsymbol{x}_t)} - \beta^j) \tag{7.30}$$

where $\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^X}$ is:

$$\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^X} = \frac{p_t^X - a_j^X}{\sqrt{(p_t^X - a_j^X)^2 + (p_t^Y - a_j^Y)^2}} \tag{7.31}$$

Then, the expectation of the score is expressed as:

$$\begin{aligned} E(\frac{\partial \log p(z_t^j|\boldsymbol{x}_t)}{\partial p_t^X}) &= \int_0^\infty \frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^X} (\frac{\alpha^j - 1}{h_t^j(\boldsymbol{x}_t)} - \beta^j) \frac{(\beta^j)^{\alpha^j}}{\Gamma(\alpha^j)} h_t^j(\boldsymbol{x}_t)^{\alpha^j - 1} e^{-\beta^j h_t^j(\boldsymbol{x}_t)} dh_t^j(\boldsymbol{x}_t) \\ &= \frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^X} [(\alpha^j - 1) \int_0^\infty \frac{(\beta^j)^{\alpha^j}}{\Gamma(\alpha^j)} h_t^j(\boldsymbol{x}_t)^{\alpha^j - 2} e^{-\beta^j h_t^j(\boldsymbol{x}_t)} dh_t^j(\boldsymbol{x}_t) - \beta^j] \end{aligned} \tag{7.32}$$

In (7.32), the integration of PDF is 1:

$$\int_0^\infty \frac{(\beta^j)^{\alpha^j}}{\Gamma(\alpha^j)} h_t^j(\boldsymbol{x}_t)^{\alpha^j - 1} e^{-\beta^j h_t^j(\boldsymbol{x}_t)} dh_t^j(\boldsymbol{x}_t) = 1 \tag{7.33}$$

For $\alpha^j = 1$, which makes the Gamma distribution as the exponential distribution:

$$E(\frac{\partial \log p(z_t^j|\boldsymbol{x}_t)}{\partial p_t^X}) = -\beta^j \frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^X} \tag{7.34}$$

For $\alpha^j > 1$ and $\alpha_i \in \mathbb{N}$, using $\Gamma(\alpha^j) = (\alpha^j - 1)\Gamma(\alpha^j - 1)$, and submit it into (7.32), (7.32) is obtained to be 0 as illustrated in (7.35).

$$\begin{aligned} E(\frac{\partial \log p(z_t^j|\boldsymbol{x}_t)}{\partial p_t^X}) = \frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^X}[(\alpha^j - 1)\int_0^\infty \frac{\beta^j(\beta^j)^{\alpha^j-1}}{(\alpha^j-1)\Gamma(\alpha^j-1)} h_t^j(\boldsymbol{x}_t)^{\alpha^j-2} \\ \times e^{-\beta^j h_t^j(\boldsymbol{x}_t)} dh_t^j(\boldsymbol{x}_t) - \beta^j] \\ = 0 \end{aligned} \tag{7.35}$$

Then, the expectation of the score function is obtained, and the covariance of score function is to be derived. The second order moment of score is formulated as:

$$(\frac{\partial \log p(z_t^j|\boldsymbol{x}_t)}{\partial p_t^X})^2 = (\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^X})^2[\frac{(\alpha^j-1)^2}{h_t^j(\boldsymbol{x}_t)^2} + (\beta^j)^2 - \frac{2\beta^j(\alpha^j-1)}{h_t^j(\boldsymbol{x}_t)}] \tag{7.36}$$

Accordingly, the covariance is (7.37):

$$\begin{aligned} E(\frac{\partial \log p(z_t^j|\boldsymbol{x}_t)}{\partial p_t^X})^2 = \int_0^\infty (\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^X})^2[(\beta^j)^2 + \frac{(\alpha^j-1)^2}{(h_t^j(\boldsymbol{x}_t))^2} \\ - \frac{2\beta^j(\alpha^j-1)}{h_t^j(\boldsymbol{x}_t)}]\frac{(\beta^j)^{\alpha^j}}{\Gamma(\alpha^j)}(h_t^j(\boldsymbol{x}_t))^{\alpha^j-1}e^{-\beta^j h_t^j(\boldsymbol{x}_t)}dh_t^j(\boldsymbol{x}_t) \\ = (\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^X})^2[(\beta^j)^2 \\ + \int_0^\infty \frac{(\alpha^j-1)^2(\beta^j)^{\alpha^j}}{\Gamma(\alpha^j)}(h_t^j(\boldsymbol{x}_t))^{\alpha^j-3}e^{-\beta^j h_t^j(\boldsymbol{x}_t)}dh_t^j(\boldsymbol{x}_t) \\ - \int_0^\infty \frac{2(\alpha^j-1)(\beta^j)^{\alpha^j+1}}{\Gamma(\alpha^j)}(h_t^j(\boldsymbol{x}_t))^{\alpha^j-2}e^{-\beta^j h_t^j(\boldsymbol{x}_t)}dh_t^j(\boldsymbol{x}_t)] \end{aligned} \tag{7.37}$$

The formulation also depends on three cases:

For $\alpha^j > 1$,:

$$E(\frac{\partial \log p(z_t^j|\boldsymbol{x}_t)}{\partial p_t^X})^2 = (\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^X})^2(\beta^j)^2 \tag{7.38}$$

For $\alpha^j = 2$:

$$E(\frac{\partial \log p(z_t^j|\boldsymbol{x}_t)}{\partial p_t^X})^2 \to \infty \tag{7.39}$$

For $\alpha^j > 2$ and $\alpha \in \mathbb{N}$, also using $\Gamma(\alpha^j) = (\alpha^j - 1)\Gamma(\alpha^j - 1)$ and (7.33):

$$E(\frac{\partial \log p(z_t^j|\boldsymbol{x}_t)}{\partial p_t^X})^2 = (\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^X})^2 \frac{(\beta^j)^2}{\alpha^j - 2} \tag{7.40}$$

Then, I_{11} is expressed in three cases accordingly:
For $\alpha^j = 1$:

$$I_{11} = \sum_{j=1}^{N} (\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^X})^2 (\beta^j)^2 + \sum_{i \neq j}^{N} \beta^i \beta^j \frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^X} \frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^X} \tag{7.41}$$

For $\alpha^j = 2$:

$$I_{11} = \rightarrow \infty \tag{7.42}$$

which is useless.
For $\alpha_i > 2$ and $\alpha \in \mathbb{N}$:

$$I_{11} = \sum_{j=1}^{N} (\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^X})^2 \frac{(\beta^j)^2}{\alpha^j - 2} \tag{7.43}$$

The procedure and expression format of I_{22} are the same as I_{11}. Thus, the derive of I_{22} is skipped and the expression of I_{22} is: For $\alpha^j = 1$:

$$I_{22} = \sum_{j=1}^{N} (\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^Y})^2 (\beta^j)^2 + \sum_{i \neq j}^{N} \beta^i \beta^j \frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^Y} \frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^Y} \tag{7.44}$$

For $\alpha^j = 2$:

$$I_{22} = \rightarrow \infty \tag{7.45}$$

which is useless.
For $\alpha_i > 2$ and $\alpha \in \mathbb{N}$:

$$I_{22} = \sum_{j=1}^{N} (\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^Y})^2 \frac{(\beta^j)^2}{\alpha^j - 2} \tag{7.46}$$

The expression of $I_{21} = I_{12}$ can be derived as follows:

$$\begin{aligned} I_{12} &= E(\sum_{j=1}^{N}\sum_{i=1}^{N}\frac{\partial \log p(z_t^j|\boldsymbol{x}_t)}{\partial p_t^X}\frac{\partial \log p(z_t^i|\boldsymbol{x}_t)}{\partial p_t^Y}) \\ &= E\{\sum_{j=1}^{N}\frac{\partial \log p(z_t^j|\boldsymbol{x}_t)}{\partial p_t^X}\frac{\partial p(z_t^j|\boldsymbol{x}_t)}{\partial p_t^Y} + \sum_{j\neq i}^{N}\frac{\partial \log p(z_t^j|\boldsymbol{x}_t)}{\partial p_t^X}\frac{\partial \log p(z_t^i|\boldsymbol{x}_t)}{\partial p_t^Y}\} \\ &= \sum_{j=1}^{N}E(\frac{\partial \log p(z_t^j|\boldsymbol{x}_t)}{\partial p_t^X}\frac{\partial p(z_t^i|\boldsymbol{x}_t)}{\partial p_t^Y}) + \sum_{j\neq i}^{N}E(\frac{\partial \log p(z_t^j|\boldsymbol{x}_t)}{\partial p_t^X}\frac{\partial p(z_t^i|\boldsymbol{x}_t)}{\partial p_t^Y}) \\ &= \sum_{j=1}^{N}E(\frac{\partial \log p(z_t^j|\boldsymbol{x}_t)}{\partial p_t^X}\frac{\partial \log p(z_t^j|\boldsymbol{x}_t)}{\partial p_t^Y}) + \sum_{j\neq i}^{N}E(\frac{\partial \log p(z_t^j|\boldsymbol{x}_t)}{\partial p_t^X})E(\frac{\partial \log p(z_t^i|\boldsymbol{x}_t)}{\partial p_t^Y}) \end{aligned} \tag{7.47}$$

Like I_{11}, the formulation of I_{12} can be also classified into 3 cases:

For $\alpha_i = 1$:

$$\begin{aligned} I_{12} &= \sum_{j=1}^{N}\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^X}\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^Y}(\beta^j)^2 + \sum_{j\neq i}^{N}\beta^j\beta^i\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^X}\frac{\partial h_t^i(\boldsymbol{x}_t)}{\partial p_t^Y} \\ &= \sum_{j=1}^{N}\sum_{i=1}^{N}\beta^j\beta^i\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^X}\frac{\partial h_t^i(\boldsymbol{x}_t)}{\partial p_t^Y} \end{aligned} \tag{7.48}$$

For $\alpha_i = 2$:

$$I_{12} =\rightarrow \infty \tag{7.49}$$

For $\alpha_i > 2$ and $\alpha \in \mathbb{N}$:

$$I_{12} = \sum_{j=1}^{N}\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^X}\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^Y}\frac{(\beta^j)^2}{\alpha^j - 2} \tag{7.50}$$

Then, *Lemma 2* is proofed. □

7.3.3 Priori Information

The priori information can be used in three ways. The first one is to fuse the priori probability with measurement data based on the Bayesian theorem. If the priori probability density function is correct, the estimation can be accurate. However, the accurate priori information can not always be obtained. In most cases, the priori information is derived from the previous state estimation. Thus, the second one is the recursive process, which estimates the prior information together with the state for the next time step. Finally, we proposed a likelihood adaptation method for wireless target tracking [45]. This method is different from the above two, which

fuse the priori information into the likelihood function. Then, we define this method as the adaptive process. Based on the concepts of the estimation processes, the FIM are formulated in different ways.

Basic Bayesian Process

For basic Bayesian process, we assume the probability density distribution of $\boldsymbol{x}_t$ is known with current given information, which means that with a given $\boldsymbol{x}_{t-1}$, the probability $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ follows $\mathcal{N}(\boldsymbol{x}_t - \boldsymbol{F}_t\boldsymbol{x}_{t-1}, \boldsymbol{Q}_t)$, where $\boldsymbol{I}_P(\boldsymbol{x}_t)$ can be expressed as:

$$\begin{aligned}\boldsymbol{I}_P(\boldsymbol{x}_t) &= E\left\{\left[\frac{\partial \ln p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})}{\partial \boldsymbol{x}_t}\right]^2\right\} \\ &= \boldsymbol{F}_t^T\boldsymbol{Q}_t^{-1}\boldsymbol{F}_t\end{aligned} \tag{7.51}$$

Then, based on Bayesian theorem, the posterior PDF $p(\boldsymbol{x}_t|\boldsymbol{z}_t) \propto p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})p(\boldsymbol{z}_t|\boldsymbol{x}_t)$. Then, $\boldsymbol{I}(\boldsymbol{x}_t)$ is formulated as:

$$\boldsymbol{I}(\boldsymbol{x}_t) = \boldsymbol{F}_t^T\boldsymbol{Q}_t^{-1}\boldsymbol{F}_t + \boldsymbol{I}_D(\boldsymbol{x}_t) \tag{7.52}$$

where $\boldsymbol{I}_D(\boldsymbol{x}_t)$ can be expressed as (7.9) for outdoor environment or (7.21) for indoor environment.

Basic Bayesian process is a simple static process. It can improve the estimation with a given priori state distribution and measurement error distribution. But in this formulation, the previous state $\boldsymbol{x}_{t-1}$ is assumed to be accurate. However, the priori information is also to be estimated for the target tracking applications. If the previous state estimation is not accurate in the real case, recursive methods, such as Kalman filter or extended Kalman filter, estimate the distributions recursively and the priori information can converge to a stable value. In such real case, the Fisher information matrix should be formulated in a recursive manner.

Recursive Process

The FIM of recursive process has been initially formulated by Petr $et\ al.$[143]. Taking the priori information into account, the priori probability is the transition probability $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$. However, both $\boldsymbol{x}_t$ and $\boldsymbol{x}_{t-1}$ should be estimated during the process. Thus, the information about $\boldsymbol{x}_{t-1}$ is also not accurate. Then, the estimated parameters for Fisher information is $[\boldsymbol{x}_t, \boldsymbol{x}_{t-1}]^T$. For recursive process, $\boldsymbol{I}(\boldsymbol{x}_t)$ is calculated recursively, which involves $\boldsymbol{I}(\boldsymbol{x}_{t-1})$ as the previous Fisher information matrix. Then, $\boldsymbol{I}(\boldsymbol{x}_t)$ is given as:

$$\boldsymbol{I}(\boldsymbol{x}_t) = \boldsymbol{A} - \boldsymbol{B}(\boldsymbol{I}(\boldsymbol{x}_{t-1}) + \boldsymbol{C})^{-1}\boldsymbol{B}^T \tag{7.53}$$

where

$$\begin{aligned}\boldsymbol{A} &= \boldsymbol{I}_P(\boldsymbol{x}_t) + \boldsymbol{I}_D(\boldsymbol{x}_t) = \boldsymbol{F}_t^T \boldsymbol{Q}_t^{-1} \boldsymbol{F}_t + \boldsymbol{I}_D(\boldsymbol{x}_t) \\ \boldsymbol{B} &= -\boldsymbol{F}_t^T \boldsymbol{Q}_t^{-1} \\ \boldsymbol{C} &= \boldsymbol{F}_t^T \boldsymbol{Q}_t^{-1} \boldsymbol{F}_t\end{aligned} \tag{7.54}$$

where $\boldsymbol{I}(\boldsymbol{x}_t)$ is a part of $\boldsymbol{I}([\boldsymbol{x}_t, \boldsymbol{x}_{t-1}]^T)$; $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$ are parts of $\boldsymbol{I}([\boldsymbol{x}_t, \boldsymbol{x}_{t-1}]^T)$, which represents the previous and current information. For detail description, please refer to [143]. In both cases, $\boldsymbol{I}_D(\boldsymbol{x}_t)$ is the same as (7.9) or (7.21) in the indoor scenario. In (7.53), it is clearly observed that both priori distribution and previous estimated distribution are involved in the state estimation. With an estimated priori state distribution, the CRLB can converge to a certain value.

Adaptive Process

For the adaptive process, the priori information is integrated into the likelihood function [45]. The key idea is to use priori information to adapt the likelihood function to the actual measurement error distribution, then a more accurate location is obtained. The predicted state $\boldsymbol{x}_t$ is transferred as the predicted measurement $h_t(\boldsymbol{F}_t \boldsymbol{x}_{t-1})$. Although the likelihood function is adapted between the predicted measurement and actual measurement data, the purpose is to make the observed measurement data approach to the real value. Then, the problem can turn into the approximation problem, which is:

$$\bar{\boldsymbol{z}}_t = \theta \boldsymbol{h}_t(\boldsymbol{F}_t \boldsymbol{x}_{t-1}) + (1-\theta)\boldsymbol{z}_t \tag{7.55}$$

where θ is the belief factor; system adapts the measurement $\bar{z}_t$ to the actual value by assigning the belief weight. The optimal θ, which can achieve the minimum error, is:

$$\theta = \frac{\boldsymbol{R}_t}{\frac{\partial \boldsymbol{h}_t(\boldsymbol{x}_t)}{\partial \boldsymbol{x}_t} \boldsymbol{Q}_t [\frac{\partial \boldsymbol{h}_t(\boldsymbol{x}_t)}{\partial \boldsymbol{x}_t}]^T + \boldsymbol{R}_t} \tag{7.56}$$

Since the measurement is adapted by the priori information, the adapted measurement $\bar{z}_t$ is the combination of predicted measurement and observed data. Then, the error covariance does not follow the original error distribution, but changes accordingly. If the measurement noise also follows Gaussian distribution, we can easily derive the covariance of adapted measurement:

$$\begin{aligned}\boldsymbol{Cov}(\bar{\boldsymbol{z}}_t) &= E\{(\boldsymbol{h}_t(\boldsymbol{x}_t) - \bar{\boldsymbol{z}}_t)(\boldsymbol{h}_t(\boldsymbol{x}_t) - \bar{\boldsymbol{z}}_t)^T\} \\ &= E\{||\theta \frac{\partial \boldsymbol{h}_t(\boldsymbol{x}_t)}{\partial p_t^X} \boldsymbol{q}_t + (1-\theta)\boldsymbol{v}_t||^2\} \\ &= \theta \frac{\partial \boldsymbol{h}_t(\boldsymbol{x}_t)}{\partial p_t^X} \boldsymbol{Q}_t [\frac{\partial \boldsymbol{h}_t(\boldsymbol{x}_t)}{\partial p_t^X}]^T \theta^T + (1-\theta)\boldsymbol{R}_t(1-\theta)^T \\ &= \bar{\boldsymbol{R}}_t\end{aligned} \tag{7.57}$$

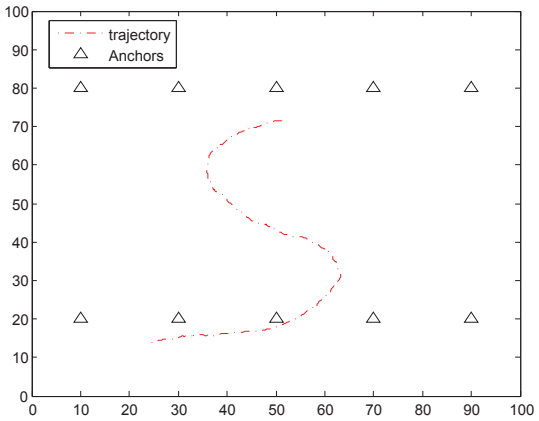


Figure 7.1: Simulation trajectory with 10 anchors

If the measurement noise is independent to each anchor, the covariance matrix can be simplified as $\bar{\boldsymbol{R}}_t = diag_{n\times n}(\bar{R}_j)$, where $\bar{R}_j$ can be derived independently in each anchor. Since priori information is considered together into the measurement data, $\boldsymbol{I}_P(\boldsymbol{x}_t) = 0$ and $\boldsymbol{I}(\boldsymbol{x}_t)$ only depends on $\boldsymbol{I}_D(\boldsymbol{x}_t)$. However, $\boldsymbol{I}_D(\boldsymbol{x}_t)$ should be re-formulated as:

$$\boldsymbol{I}_D(\boldsymbol{x}_t)=\begin{pmatrix} \sum_{j=1}^{n} \bar{R}_j^{-1}(\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^X})^2 & \sum_{j=1}^{n} \bar{R}_j^{-1}\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^X}\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^Y} \\ \sum_{j=1}^{n} \bar{R}_j^{-1}\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^X}\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^Y} & \sum_{j=1}^{n} \bar{R}_j^{-1}(\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^Y})^2 \end{pmatrix} \tag{7.58}$$

where we use the new measurement error covariance $\bar{R}_j$ instead of the original R_j; the framework is still the same as FIM without priori information.

In the adaptive process, we still assume that the previous estimation $\boldsymbol{x}_{t-1}$ is accurate. With a given priori state distribution, the current state is obtained through the adaptation method. As mentioned in our previous work, our method has robust performance in the high noise environment, but does not improve much with small measurement noise. If the priori distribution is highly unreliable, our method does not improve much either.

7.4 CRLB Analysis for Bayesian Model

7.4.1 Environment

We set up several simulations to evaluate the performance improvement using priori information for different processes. The playing field is $100 \times 100m^2$. Ten anchors are uniformly deployed. The target chooses a random path walking through the playing field as illustrated in Fig. 7.1. The localization estimation of the mobile target is based on the ranging measurement. The range data is obtained through TOA technique.

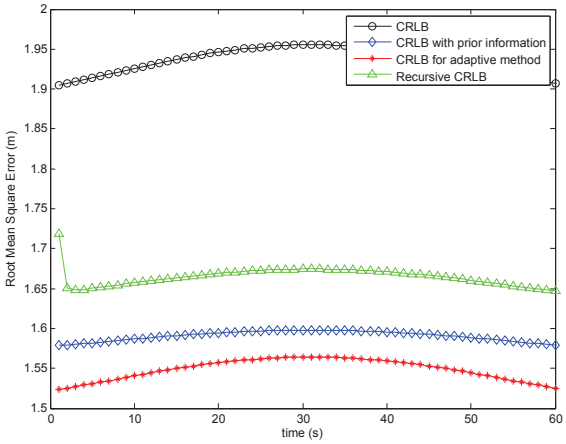


Figure 7.2: Cramér-Rao lower bound for target tracking

The CRLB without priori information is compared with the CRLBs of three processes. The simulation results are averaged by 1000 Monte-Carlo trial. For each trial, the target randomly chooses a path, and the CRLBs are derived accordingly. Root mean square error (RMSE) is mainly compared as the estimation accuracy metric.

The measurement error variance for a single anchor is $R_j = 3m^2$, which is assumed as Gaussian distribution. The prior distribution is also assumed as Gaussian distribution $\mathcal{N}(\boldsymbol{x}_t - \boldsymbol{F}_t(\boldsymbol{x}_{t-1}), \boldsymbol{Q}_t)$, where $\boldsymbol{Q}_t = diag(2, 2)$. The CRLBs are depicted in Fig. 7.2.

7.4.2 Performance Comparison

As indicated in Fig. 7.2, the CRLB without priori information has the highest RMSE. The three processes with priori information have better performance. With an accurate previous state estimation $\boldsymbol{x}_{t-1}$, the basic Bayesian process outperforms recursive process, since the priori information is not accurate in the recursive process. The recursive process converges fast. Since the priori information is also a parameter to be estimated, the recursive process approaches mostly to the real localization case. The adaptive method has the lowest RMSE. Thus, with a given priori information, our proposed likelihood adaptation method is accurate for location estimation.

We also adapt the measurement noise environment to examine the performance change for each process. The measurement error covariance for each anchor changes from $R_j = 0.5m^2$ to $R_j = 5.5m^2$. Also, the results are averaged by 1000 Monte-Carlo trials, as illustrated in Fig. 7.3. The CRLB without priori information still has the highest RMSE. The CRLB of basic Bayesian process is better than the CRLB of recursive process. The performance of adaptive process changes a lot during the increase of measurement error covariance. When the error covariance is small, the adaptive process can not improve estimation but increases the estimation error. As analyzed in our previous work, if the error is small, our method does not improve much [45].

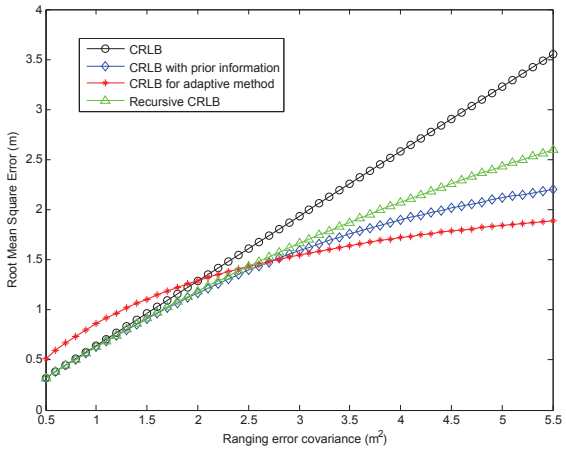


Figure 7.3: Cramér-Rao lower bound with multiple measurement noise environment

In this case, it would rather use only the measurement data to obtain the estimation. However, with increasing the error covariance, the adaptive process improve the estimation gradually, and shows the best performance when the error is quite high. Thus, our method is feasible for the harsh environment with very high noise.

7.4.3 Unreliable Priori Information

In the previous simulations, we assume the priori information is reliable, in which the prediction covariance is not so high. We change the prediction covariance and check the performance of the CRLBs in the unreliable priori information cases. The prediction covariance is adapted from 0.5 to 4 m^2. As illustrated in Fig. 7.4, with the increase of the prediction covariance, the CRLBs of basic Bayesian process and recursive process converge to the CRLB without priori information. It indicates that the estimation performance does not improve much with unreliable information. The CRLB of adaptive process increases quite significantly. When the prediction covariance is so high, the RMSE of adaptive process increase dramatically and is even the highest of all the curves.

However, this simulation with high prediction covariance can not happen in the real case. If the priori information is unreliable, system will abandon such information and just relies on the measurement data. Besides, the priori information can be improved according to the previous estimation. In the recursive process, although the priori information is not reliable initially, it can still be obtained through the recursive process just as the family of Kalman filters do. Thus, this simulation indicates that it is better to use a recursive process when the priori information is unreliable.

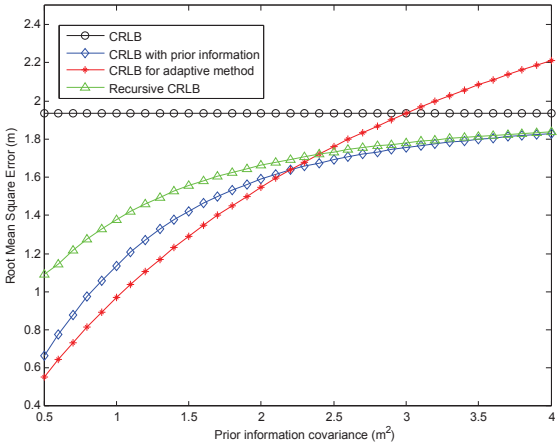


Figure 7.4: Cramér-Rao lower bound with multiple priori information

7.4.4 Multiple Anchors

Unlike the 10 uniformly deployed anchors, the anchors are randomly deployed in Fig. 7.5. For each number of anchors, the simulation is run 1000 trials. The number of anchors changes from 5 to 100. The ranging measurement error covariance is still 3 m^2. To approach to the real localization system, the prior distribution is still assumed as Gaussian distribution $\mathcal{N}(\boldsymbol{x}_t - \boldsymbol{F}_t(\boldsymbol{x}_{t-1}), \boldsymbol{Q}_t)$, where $\boldsymbol{Q}_t = diag(2, 2)$. The averaged RMSE are depicted in Fig. 7.5.

The CRLBs converge gradually to a low RMSE with increasing anchors. In this case, the CRLBs of basic Bayesian process and recursive process approach to the CRLB without priori information, which means that with sufficient measurement data, the priori information plays less important role. However, the adaptive process still show significant improvement. In Fig. 7.5, our adaptive method still has the lowest RMSE, which indicates that fusing the proper priori information into the likelihood function can improve the estimation performance a lot.

7.5 3D Geometric Analysis

Since most location system estimate the target's two dimension position: x and y coordinates, the estimation results are displayed in a 2D playing field. And the range is assumed as the 2D distance between anchor and the target. However, one important factor is ignored: the relative height between the anchor and the target. In 2D location estimation, the anchor and target are assumed to be placed on the same level. But in the real experiments and applications, the anchor and target are not always on the same level. Although the relative height can be ignored in a large open area, such as cell network where the distance between mobile phone and base station is much longer than the relative height, in the small area, $e.g.$ indoor building, where the anchors are deployed just near the target, the relative height can lead to large estimation

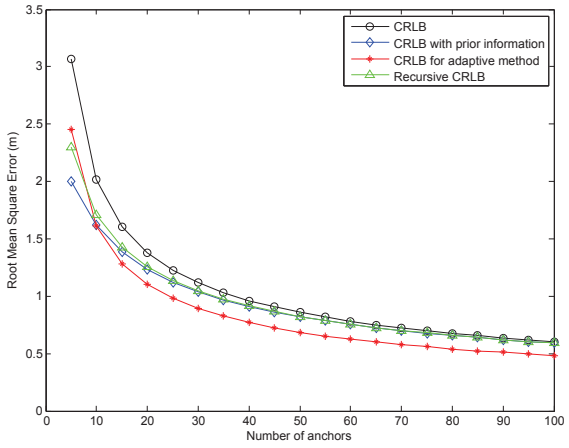


Figure 7.5: Cramér-Rao lower bound with different number of anchors

error. Thus, the impact of height difference between the target and anchors are important to the position estimation for the indoor target tracking systems. In this chapter, the CRLB is also used to analyze such impact and illustrate how the height difference influence the estimation error.

7.5.1 3D-ranging

Here, we define the height difference between anchor and target as **relative height** $\triangle z$, which is a positive variable in Z axel. If the relative height is 0 or assumed to be 0 in the simulation, we define the range measurement as **2D-ranging**. The measurement for each anchor is formulated as:

$$z_t^j = h_t^j(\boldsymbol{x}_t) + v_t^j = \sqrt{(p_t^X - a_j^X)^2 + (p_t^Y - a_j^Y)^2} + v_t^j \tag{7.59}$$

where z_t^j denotes the measurement for jth anchor; $h_t^j(\boldsymbol{x}_t)$ is the measurement function; $\boldsymbol{x}_t = (p_t^X, p_t^Y)^T$ is the target's coordinates; $(a_j^X, a_j^Y)^T$ denotes the anchor's position; v_t^j is the measurement noise, which is independent to other range measurement noise. If the relative height between anchor and target is not 0, which is always applicable in the real case, the measurement depends on 3D coordinates, define the range measurement as **3D-ranging**. The 3D-ranging for each anchor is formulated as:

$$z_t^j = h_t^j(\boldsymbol{x}_t) + v_t^j = \sqrt{(p_t^X - a_j^X)^2 + (p_t^Y - a_j^Y)^2 + \triangle z^2} + v_t^j \tag{7.60}$$

In the real application, the anchors can be deployed with different relative heights to the target, which makes the analysis rather complicated. It is assumed that the relative height is fixed during simulation for simplicity, which can depict a straightforward figure for analysis.

Fig. 7.6 illustrates the difference between 2D-ranging and 3D-ranging. Suppose a person or robot carrying a mobile device walking in the building. Anchor 1 and Anchor 2 are deployed on the roof, and the distance to the target depends on the target position on the ground and the relative height. Even target is just below the anchor, the measurement is still not 0 due to $\triangle z$.

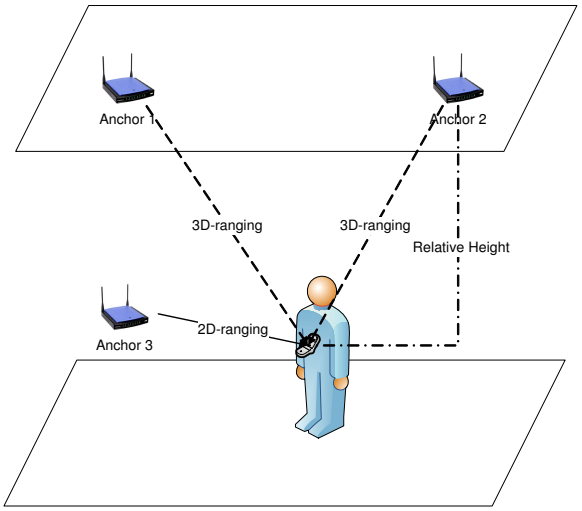


Figure 7.6: Demonstration of 3D-ranging.

Anchor 3 is adapted on the same plane of the target. In this case, the range measurement is the 2D-ranging. However, the 2D-ranging is an ideal case, and the anchors can not be adapted automatically to the same plane of the targets.

No matter of the 2D-ranging or the 3D-ranging, the position estimation is still for the 2D coordinates in the playing field, which means the system just wants to obtain $\boldsymbol{x}_t = (p_t^X, p_t^Y)^T$ and not $\triangle z$. But for CRLB analysis, if $\triangle z$ is involved into the calculation and used as the prior information, can the estimation accuracy be improved?

7.5.2 FIM for 3D-ranging

In 3D-ranging, the measurement follows as (7.60). If we ignore $\triangle z$, and estimate the 2D position coordinates (p_t^X, p_t^Y), we derive the Fisher information matrix as:

Theorem 1. *For normal distribution, FIM is expressed as:*

$$I_{3D}(\boldsymbol{x}_t)=\begin{pmatrix} \sum_{j=1}^{N} R_j^{-1}(\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^X})^2 & \sum_{j=1}^{N} R_j^{-1}\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^X}\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^Y} \\ \sum_{j=1}^{N} R_j^{-1}\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^X}\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^Y} & \sum_{j=1}^{N} R_j^{-1}(\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^Y})^2 \end{pmatrix} \tag{7.61}$$

For the Gamma distribution, the FIM is expressed as:

$$I_{3D}(\boldsymbol{x}_t)=\begin{pmatrix} \sum_{j=1}^{N}(\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^X})^2\frac{\beta_j^2}{\alpha_j-2} & \sum_{j=1}^{N}\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^X}\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^Y}\frac{\beta_j^2}{\alpha_j-2} \\ \sum_{j=1}^{N}\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^X}\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^Y}\frac{\beta_j^2}{\alpha_j-2} & \sum_{j=1}^{N}(\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^Y})^2\frac{\beta_j^2}{\alpha_j-2} \end{pmatrix} \tag{7.62}$$

where $\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^X}$ *and* $\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^Y}$ *are:*

$$\begin{cases} \frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^X} = \frac{p_t^X - a_j^X}{\sqrt{(p_t^X - a_j^X)^2 + (p_t^Y - a_j^Y)^2 + \triangle z^2}} \\ \frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^Y} = \frac{p_t^Y - a_j^Y}{\sqrt{(p_t^X - a_j^X)^2 + (p_t^Y - a_j^Y)^2 + \triangle z^2}} \end{cases} \tag{7.63}$$

Proof. For 3D-ranging, the measurement is adapted as:

$$z_t^j = h_t^j(\boldsymbol{x}_t, \triangle z) + v_t^j = \sqrt{(p_t^X - a_j^X)^2 + (p_t^Y - a_j^Y)^2 + \triangle z^2} + v_t^j \tag{7.64}$$

where $\boldsymbol{x}_t = [p_t^X, p_t^Y]^T$ is 2D position coordinates without the relative height. Then substitute z_t^j into (7.17), we obtain (7.61) in the normal distribution case. Note that, the partial differential of measurement function is:

$$\frac{\partial \boldsymbol{h}_t(\boldsymbol{x}_t, \triangle z)}{\partial p_t^X} = [\frac{\partial h_t^T(\boldsymbol{x}_t, \triangle z)}{\partial p_t^X}, \frac{\partial h_t^T(\boldsymbol{x}_t, \triangle z)}{\partial p_t^Y}]^T \tag{7.65}$$

For Gamma distribution, substitute z_t^j into (7.26), and follow the same procedure as (7.29), then (7.62) is attained. Thus, *Theorem 1* is proofed. □

It is clearly observed that, the FIM in (7.61) and (7.62) contain $\triangle z^2$, which is the main difference compare to the FIM in 2D-ranging. If $\triangle z^2$ is small, (7.61) and (7.62) are almost the same to (7.9) and (7.21). However, if $\triangle z^2$ is big, the FIMs can be significantly different. Thus, for the 2D estimation, the CRLB is not reliable to indicate the real localization performance and even for some anchor selection algorithms. If $\triangle z^2$ is big enough, which means that the anchors are much higher than the target, increasing $\triangle z^2$ does not change the formulation too much. In this case, the 2D localization system approaches to the performance of GPS, then the GDOP-based algorithms are available. However, in the indoor localization, the anchors can not approach to the satellites, and the conventional CRLB is not applicable.

Another feasible solution for 3D-ranging is to estimate both 2D coordinates and the relative height. Then, it turns to be a 3D positioning problem. Here, we assume all the anchors are deployed on the same height, which means not relative height exists among anchors. And $\triangle z^2$ is fixed to all the anchors. Then, FIM for 3D ranging is obtained:

Theorem 2. *For normal distribution, the FIM is expressed as (7.66):*

$$I_{3D}(\boldsymbol{x}_t) = \begin{pmatrix} \sum_{j=1}^N R_j^{-1} (\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^X})^2 & \sum_{j=1}^N R_j^{-1} \frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^X} \frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^Y} & \sum_{j=1}^N R_j^{-1} \frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^X} \frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial \triangle z} \\ \sum_{j=1}^N R_j^{-1} \frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^X} \frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^Y} & \sum_{j=1}^N R_j^{-1} (\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^Y})^2 & \sum_{j=1}^N R_j^{-1} \frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^Y} \frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial \triangle z} \\ \sum_{j=1}^N R_j^{-1} \frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^X} \frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial \triangle z} & \sum_{j=1}^N R_j^{-1} \frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^Y} \frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial \triangle z} & \sum_{j=1}^N R_j^{-1} (\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial \triangle z})^2 \end{pmatrix} \tag{7.66}$$

And for Gamma distribution, the FIM is expressed as (7.67):

$$I_{3D}(\boldsymbol{x}_t)=\begin{pmatrix} \sum_{j=1}^{N}\frac{\beta_j^2}{\alpha_j-2}(\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^X})^2 & \sum_{j=1}^{N}\frac{\beta_j^2}{\alpha_j-2}\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^X}\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^Y} & \sum_{j=1}^{N}\frac{\beta_j^2}{\alpha_j-2}\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^X}\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial \triangle z} \\ \sum_{j=1}^{N}\frac{\beta_j^2}{\alpha_j-2}\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^X}\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^Y} & \sum_{j=1}^{N}\frac{\beta_j^2}{\alpha_j-2}(\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^Y})^2 & \sum_{j=1}^{N}\frac{\beta_j^2}{\alpha_j-2}\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^Y}\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial \triangle z} \\ \sum_{j=1}^{N}\frac{\beta_j^2}{\alpha_j-2}\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^X}\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial \triangle z} & \sum_{j=1}^{N}\frac{\beta_j^2}{\alpha_j-2}\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^Y}\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial \triangle z} & \sum_{j=1}^{N}\frac{\beta_j^2}{\alpha_j-2}(\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial \triangle z})^2 \end{pmatrix} \tag{7.67}$$

In (7.66) and (7.67) $\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^X}$ *and* $\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial p_t^Y}$ *follow (7.63), and* $\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial \triangle z}$ *is expressed as:*

$$\frac{\partial h_t^j(\boldsymbol{x}_t)}{\partial \triangle z}=\frac{\triangle z}{\sqrt{(p_t^X-a_j^X)^2+(p_t^Y-a_j^Y)^2+\triangle z^2}} \tag{7.68}$$

Although $\triangle z$ is a parameter to be estimated, the performance accuracy only concerns about 2D playing field. Thus, the estimation error of $\triangle z$ can be ignored. Then, the CRLB in this case is formulated as:

$$Cov_{\boldsymbol{x}_t}(\tilde{x}_t) \geq diag(I_{3D}^{-1}(1,1), I_{3D}^{-1}(2,2)) \tag{7.69}$$

Proof. In *Theorem 2*, we attempt to construct the state $\boldsymbol{x}_t$, which contains relative height. When the relative height is a parameter to be estimated, the measurement model does not change, but the estimation vector changes, $\boldsymbol{x}_t = [p_t^X, p_t^Y, \triangle z]^T$. Then, the partial differential of measurement function is re-constructed as:

$$\frac{\partial \boldsymbol{h}_t(\boldsymbol{x}_t)}{\partial p_t^X}=[\frac{\partial h_t^T(\boldsymbol{x}_t)}{\partial p_t^X}, \frac{\partial h_t^T(\boldsymbol{x}_t)}{\partial p_t^Y}, \frac{\partial h_t^T(\boldsymbol{x}_t)}{\partial \triangle z}]^T \tag{7.70}$$

The FIM calculation for normal distribution follows the same procedure as *Lemma 1*. And substitute (7.70) into (7.17), we obtain (7.66).

For Gamma distribution, the procedure is complex, which requires to calculate each element. The FIM can be constructed as:

$$I_{3D}(\boldsymbol{x}_t)=E(\frac{\partial \log p(\boldsymbol{z}_t|\boldsymbol{x}_t)}{\partial p_t^X})^2=\begin{pmatrix} I_{11} & I_{12} & I_{13} \\ I_{21} & I_{22} & I_{23} \\ I_{31} & I_{32} & I_{33} \end{pmatrix} \tag{7.71}$$

where each element in FIM is:

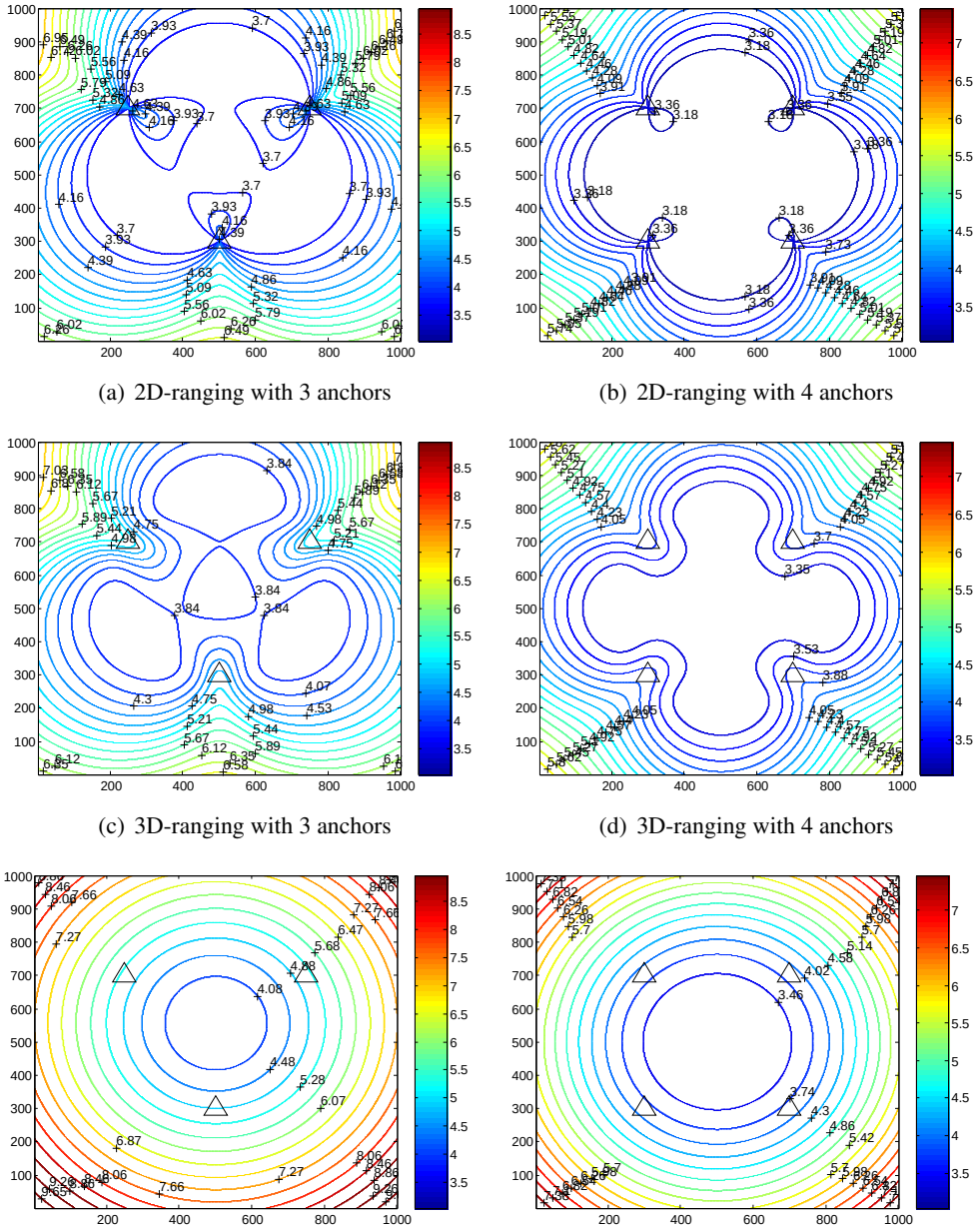
$$\begin{cases} I_{11} = E(\sum_{j=1}^{N} \frac{\partial \log p(z_t^j|\boldsymbol{x}_t)}{\partial p_t^X})^2 \\ I_{12} = I_{21} = E(\sum_{j=1}^{N} \frac{\partial \log p(z_t^j|\boldsymbol{x}_t)}{\partial p_t^X} \sum_{j=1}^{N} \frac{\partial \log p(z_t^j|\boldsymbol{x}_t)}{\partial p_t^Y}) \\ I_{13} = I_{31} = E(\sum_{j=1}^{N} \frac{\partial \log p(z_t^j|\boldsymbol{x}_t)}{\partial p_t^X} \sum_{j=1}^{N} \frac{\partial \log p(z_t^j|\boldsymbol{x}_t)}{\partial \triangle z}) \\ I_{22} = E(\sum_{j=1}^{N} \frac{\partial \log p(z_t^j|\boldsymbol{x}_t)}{\partial p_t^Y})^2 \\ I_{23} = I_{32} = E(\sum_{j=1}^{N} \frac{\partial \log p(z_t^j|\boldsymbol{x}_t)}{\partial p_t^Y} \sum_{j=1}^{N} \frac{\partial \log p(z_t^j|\boldsymbol{x}_t)}{\partial \triangle z}) \\ I_{33} = E(\sum_{j=1}^{N} \frac{\partial \log p(z_t^j|\boldsymbol{x}_t)}{\partial \triangle z})^2 \end{cases} \tag{7.72}$$

For each element, follow the same procedure as (7.29). I_{11}, I_{12}, I_{21} and I_{22} are still the same. The calculation for other elements is similar, then (7.67) is attained. Thus, *Theorem 2* is proofed. □

7.5.3 Contours Comparison

When the measurement noise follows identical independent distribution (i.i.d), the parameters R_j, α^j and β^j can be ignored and CRLB is simplified into GDOP, which means that the geographic shapes for CRLBs in normal distribution case and Gamma distribution case are the same. Here, we only compares the contours of CRLB in normal distribution case. We regularly deploy 3 and 4 anchors, and calculate CRLB for every position in the whole playing field. To simulate the real indoor localization, we set the relative height $\triangle z = 1m$. The measurement noise follows i.i.d normal distribution $v_t^j \sim \mathcal{N}(0, \sigma_j^2)$, where $\sigma_j = 3$.

Fig. 7.7 illustrates the analytical results of CRLB in different scenarios. Fig. 7.7(a) and Fig. 7.7(b) are the CRLBs for 2D-ranging with 3 and 4 anchors deployed in the playing field. Fig. 7.7(c) and Fig. 7.7(d) depict the 3D-ranging CRLB. Ignoring the CRLB values, the geometric shapes are significant different between Fig. 7.7(a), Fig. 7.7(b) and Fig. 7.7(c), Fig. 7.7(d). In 2D-ranging case, contours converge to the anchor positions. However, the contours in 3D-ranging case do not converge and CRLB near the anchors are higher than the central positions of the playing field. And Fig. 7.7(b) and 7.7(d) are more likely to the real world case. If we add the relative height $\triangle z$ as the third parameter of the estimated state, the CRLB results are plotted in Fig. 7.7(e) and 7.7(f). It is clearly observed that the geometric shapes turn into circles, which are quite different to Fig. 7.7(c) and Fig. 7.7(d). Besides, CRLB in Fig. 7.7(c) and 7.7(d)



(a) 2D-ranging with 3 anchors

(b) 2D-ranging with 4 anchors

(c) 3D-ranging with 3 anchors

(d) 3D-ranging with 4 anchors

(e) 3D-ranging with 3 anchors and relative height estimation

(f) 3D-ranging with 4 anchors and relative height estimation

Figure 7.7: CRLB Analysis Results

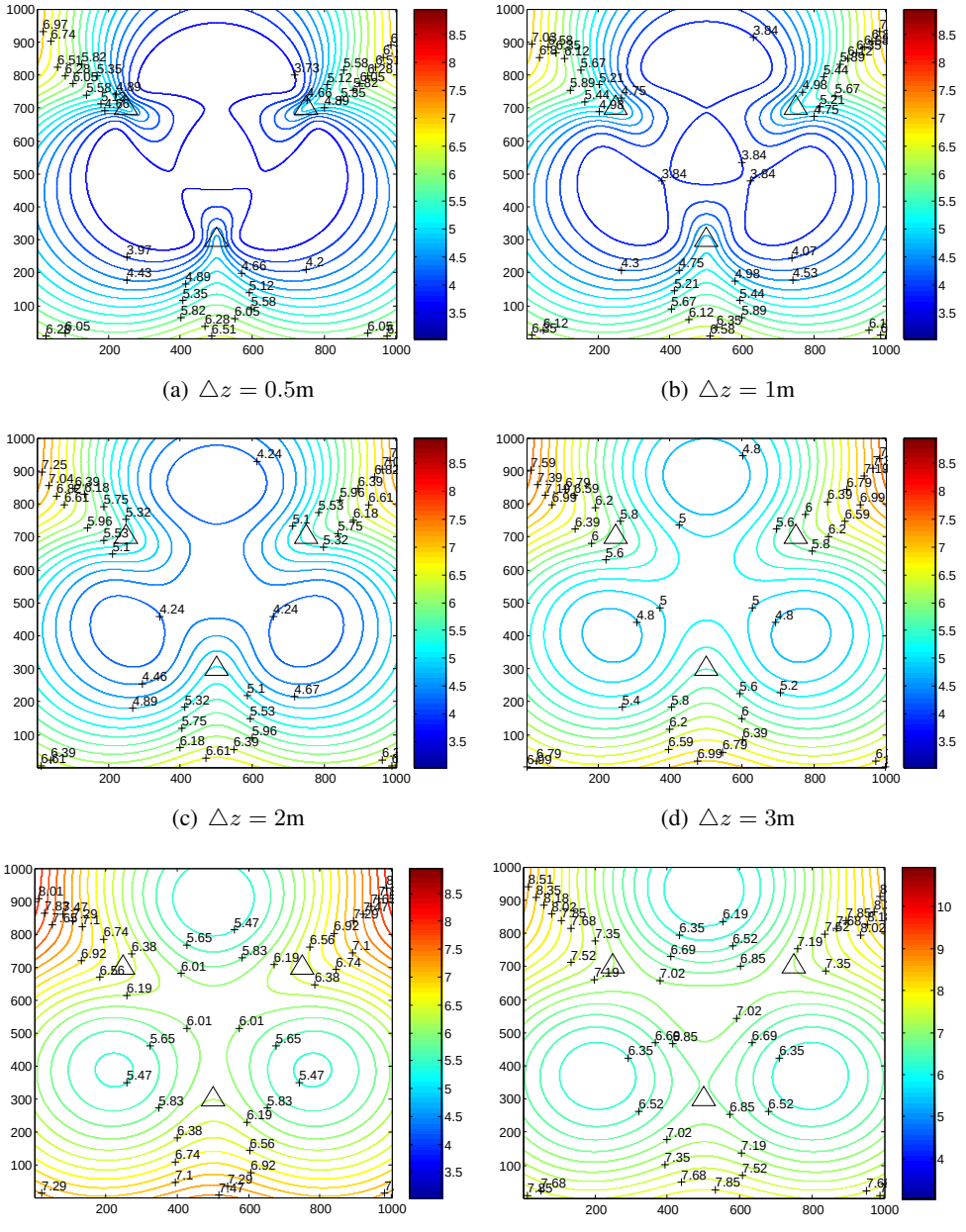
are lower than CRLB at the same positions in Fig. 7.7(e) and 7.7(f). Therefore, estimating $\triangle z$ together with 2D position to obtain a more accurate position is not a good solution. 2D position estimation with 3D-ranging is preferred if the $\triangle z$ is unknown. Besides, algorithms based on GDOP or CRLB are not reliable.

We adapt the relative height to investigate the impact for CRLB. The relative height $\triangle z$ is set to be 0.5m to 5m. Fig. 7.8 illustrates the CRLB results when 3 anchors are deployed. When $\triangle z$ is small, the geometric shape is similar to Fig. 7.7(a), but the contours do not converge to the positions of anchors. When the relative height increases, the minimum contours turn into circle in Fig. 7.8(f). Thus, in cell-phone localization system, the base station are much higher than the user, $\triangle z$ is an important factor to be considered.

7.6 Summary

Bayesian estimation method is widely used as the non-linear filter for mobile localization and tracking applications in the wireless systems. Analyzing CRLB can help us understand the performance of Bayesian method. The Bayesian methods are classified into three types according to how to use the priori information, which are basic Bayesian process, recursive process and adaptive process. The CRLBs of these processes are formulated and compared in the simulations. According to the simulation results, with an accurate and reliable the basic Bayesian process and adaptive process show much better performance than recursive process. Recursive process can converge the priori distribution gradually when the priori information is not reliable. Thus, it is useful in the non-parametric real case. When the priori information is highly unreliable, the performance of adaptive process is degraded dramatically. Therefore, to obtain an accurate location of the target, which method to choose depends on the environment and what information is attained.

The target and anchors are not always on the same height level in the real application. The relative height can influence the performance of location estimation, and it should be considered in the algorithm design. The CRLB analysis indicates that the estimation accuracy degrades and the geometric contour shape changes from 2D-ranging to 3D-ranging. Thus, the traditional algorithms based on CRLB or GDOP are unreliable in the 3D-ranging case.



(a) $\triangle z = 0.5$m (b) $\triangle z = 1$m

(c) $\triangle z = 2$m (d) $\triangle z = 3$m

(e) $\triangle z = 4$m (f) $\triangle z = 5$m

Figure 7.8: CRLB analysis with different $\triangle z$ and 3 anchors

Chapter 8

Conclusion and Future Work

In this thesis, the main goal is to implement a reliable algorithm for the indoor wireless target tracking system. The target position is derived according to the noisy ranging measurements from anchors and other related information. The adaptive particle filters are proposed in this work to improve the estimation accuracy though the high noise, hybrid LOS/NLOS rang-based measurement data. The estimation performances of the adaptive particle filters are evaluated through the simulations, experiments and the CRLB analysis. The results indicate that the adaptive PFs are applicable for the indoor target tracking application.

The system for real target tracking application based on wireless sensor network is constructed. The architecture of the tracking system with the implemented components is introduced. In addition, the design of the distributed PFs are discussed in this work.

8.1 Contribution

The major contribution of this work are five folds, which are listed below:

(1) The adaptive PFs are proposed for the wireless target tracking system. In this work, the impact of the instantaneous measurement noise for the likelihood function is analyzed. The prediction information is employed to adapt the likelihood calculation in the PF. The adaptive likelihood function is tuned by a belief factor θ, and the optimal value is derived through a minimum KLD method. Since the calculation of the likelihood is dependent to the other parts of the PF, the adaptation method can be integrated with many PF algorithms. The A-BPF, A-GPF, and A-CPF are proposed accordingly.

(2) A dynamic Gaussian modeling method for range based measurement error is proposed, which attempts to describe the noise within a dynamic framework. The measurement noise, no matter of LOS or NLOS measurement, can be modeled based on a general Gaussian distribution. This general model is not obtained according to the fitting parameters from the statistical

histogram, but based on the maximum probability and the shape of the distribution. The instantaneous error is considered as the drift from the general Gaussian model. Thus, the proposed likelihood adaptation method is used for the non-Gaussian case.

(3) A context-aware particle filter (CA-PF) which fuses the building layout information is proposed for the real system. The CA-PF overcomes the drawbacks of map matching based PFs, which limits the estimation within a constrain area, uses large data base and can not reduce the error. It construct a joint constraint area via building layout information and the ranging measurements. This method greatly improves the estimation accuracy without imposing computation costs. Besides, for real indoor localization, this method does not need too many particles for estimation. The estimation performance outperforms other adaptive PFs and other localization algorithms.

(4) The components within the real localization system architecture which help the localization algorithm improve the estimation are introduced, e.g. initialization, pre-processing, anchor selection. These components are necessary for the real wireless system, because they jointly filter the reliable information for the PFs to obtain an accurate position. Besides the proposed adaptive PFs, several localization algorithms can also be integrated within the system.

(5) Two communication protocols are designed for the DPF implementation, which are the PSG-DPF and the KS-DPF. For some special cases, the PFs are required to be implemented in a distributed way. The goal is to reduce the communication overhead and guarantee the estimation performance. The pairwise selective gossip algorithm is applied for the DPF design, which reduces the communication cost significantly and requires no resampling stage.

8.2 Conclusion

The proposed adaptive PFs are evaluated in both simulations and real experiments. In simulations, the estimation performance is evaluated in multiple simulation environments to verify the improvement of the adaptive PFs in Chapter 3. In the simulations, multiple simulation environments are constructed by setting different parameters, e.g., tuning the measurement error covariance, deploying different number of anchors, choosing the trajectories randomly. Thus, the simulation can comprehensively indicate the estimation performance of the adaptive PFs in many situations. Then, consider the real experiments, adaptive PFs using DGM are evaluated in the real experiments in Chapter 5. When the measurement error variance for a single anchor is set as $5m^2$ with about 20 deployed anchors, the simulation and experiment results are exactly the same. Therefore, both the simulation and real experiment can indicate the estimation improvements of the adaptive PFs. In addition, the performance of the algorithms are also evaluated according to the CRLB analysis in Chapter 7. For the DPFs, the evaluations are carried out in the simulation in Chapter 6, since the real anchors can not perform gossip communication in

the reference system. According to both simulations and real experiments, the conclusions are listed below:

(1) The likelihood adaptation method in the adaptive PFs significantly improve the estimation performance in a high noise environment. This method uses a belief factor θ to tune the likelihood function between the predicted measurement and the noisy measurement. The adaptive PFs based on the method achieve about $1.5m$ of RMSE in simulation, which is $0.5m$ to $1m$ better than the original PFs. If the prior information is reliable and properly tuned with the optimal θ, the estimated error is reduced. However, according to the simulation results and the CRLB analysis, the performance can not be improved with unreliable prediction. Besides, if the measurement error is low, the adaptive PFs do not perform better than the original PFs.

(2) Using DGM, the adaptive PFs can be implemented in the non-Gaussian environment. The DGM can describe multiple distribution of the measurement error dynamically. If the DGM is properly constructed, the adaptive PFs outperform the PFs based on typical distribution. For different environments, the estimation performance of the adaptive PFs are different. The A-GPF obtains more accurate positions than the A-BPF in the Gamma distribution scenario, but the A-BPF is better in the Rayleigh distribution case. In general, the RMSEs of the adaptive PFs are still controlled within $2m$ in simulations.

(3) According to the simulation and experimental results, A-CPF has the best estimation performance of all the adaptive PFs, which is $0.5m$ better than others in average. When considering the real building layout information, CA-PF is even better than A-CPF in the real target tracking system. The estimation error of CA-PF is about $0.4m$ better than A-CPF. For other algorithms, the RMSEs are almost more than $2m$. The other metrics of the algorithms are still not as good as adaptive PFs. The number of particles for the adaptive PFs is not large for the target tracking system. Especially for the CA-PF, only 20 or 30 particles are enough for the location estimation. Besides, the estimation performance is also determined by the number of anchors. The simulation indicates that more anchors achieve lower estimation error, which is from $2.5m$ to $1.5m$ in average.

(4) Selective gossip algorithm is suitable for implementing the DPF. Both of the PSG-DPF and KS-DPF can achieve similar accuracy of the centralized PF. The PSG-DPF does not need to know the overall size of the network. However, the communication overhead of the PSG-DPF is larger than the KS-DPF, since the PSG-DPF uses two round gossiping. Both of the proposed DPFs do not require additional resampling stage, which reduce the computation cost.

8.3 Future Work

The main goal for designing the adaptive PFs is to obtain an accurate target position using wireless system. Since various situations with multiple techniques should also be considered, it

is expected to investigate the following questions:

(1) The ranging measurement noise in this work is conditional independent. Recently, a combined fusion scheme is popular, which combines one measurement technique with another, e.g. combining RSS with TOA. Since the sources of measurement are from the same target, the measurement noise are not independent. In this case, the formulation of the optimal θ should be re-calculated. Besides, the impact of the combination should be also analyzed through simulations and experiments, or even CRLB analysis.

(2) DGM can describe the complicated environment, and the parameters are still obtained from the statistical histogram. For the real system, the histogram can not be correctly obtained. Thus, a nonparametric estimation method is required to estimate the parameters of the DGM in a real environment. The expectation and maximization (EM) algorithm is expected to be employed. Then, the adaptive PFs with the DGM can be extended freely to any systems.

(3) For indoor localization system, it is not just estimate the target position in a 2D map. The real implementation should also consider the relative height between the target and deployed anchors, which influences the estimation accuracy as analyzed in chapter 7. Besides, it is also required to know which floor is the target in for real system. Thus, 3D tracking is also necessary for the indoor system. Since the height estimation is not strict, the computation complexity does not increase too much.

(4) The location estimation can also be improved via combining other information. Using the IMU, the target is located even without anchors. However, localization using IMU (also denoted as dead reckoning) is only the compensation method for the range-based system, or vice versa. A feasible scheme using the range measurement and IMU information simultaneously is required and expected to improve the estimation accuracy.

(5) The proposed algorithms should be tested in a various situations. The performance should be robust to the environmental changes, e.g. from outdoor to indoor. In addition, it is required to develop a wireless localization system which is feasible for multiple wireless techniques. The expected system can be implemented in any equipment, e.g. smartphone, sensor node or other wireless devices. The range measurements are obtained from multiple wireless techniques e.g. WSN, WiFi, LTE, or cognitive radio network. Then, the algorithms in the localization system can self-adapt the likelihood function and obtain the estimation according to the multiple wireless techniques.

(6) For the real DPF implementation, both of the hardware and software should be considered. For a single node, the local PF design should be energy efficient and also considers the communication constrains. The media access control protocol integrated with gossip algorithm for the particle sharing is considered for the wireless network. Besides, the data formate is essential for the gossip algorithm, since it determines the communication overhead. Therefore, designing a typical DPF for a specific network is a future research direction.

Bibliography

[1] I. Guvenc and C.-C. Chong, "A survey on toa based wireless localization and nlos mitigation techniques," *Communications Surveys & Tutorials, IEEE*, vol. 11, no. 3, pp. 107–124, 2009.

[2] G. Mao, B. Fidan, and B. Anderson, "Wireless Sensor Network Localization Techniques," *Computer Networks*, vol. 51, no. 10, pp. 2529–2553, 2007.

[3] C.-F. Lin, *Modern navigation, guidance, and control processing*. Prentice Hall Englewood Cliffs, New Jersey, 1991, vol. 2.

[4] K. Axel, "Location-Based Services: Fundamentals and Operation," *John Wiely & Sons*, 2005.

[5] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of Wireless Indoor Positioning Techniques and Systems," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 37, no. 6, pp. 1067–1080, 2007.

[6] A. Boukerche, H. A. Oliveira, E. F. Nakamura, and A. A. Loureiro, "Localization systems for wireless sensor networks," *IEEE Wireless Communications*, vol. 14, no. 6, pp. 6–12, 2007.

[7] A. Mahtab Hossain, H. N. Van, Y. Jin, and W.-S. Soh, "Indoor localization using multiple wireless technologies," in *Mobile Adhoc and Sensor Systems, 2007. MASS 2007. IEEE Internatonal Conference on*. IEEE, 2007, pp. 1–8.

[8] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero III, R. L. Moses, and N. S. Correal, "Locating the Nodes: Cooperative Localization in Wireless Sensor Networks," *Signal Processing Magazine, IEEE*, vol. 22, no. 4, pp. 54–69, 2005.

[9] M. Anisetti, C. A. Ardagna, V. Bellandi, E. Damiani, S. Reale *et al.*, "Advanced localization of mobile terminal in cellular network," *Int'l J. of Communications, Network and System Sciences*, vol. 1, no. 01, p. 95, 2008.

[10] J. Biswas and M. Veloso, "Wifi localization and navigation for autonomous indoor mobile robots," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 4379–4384.

[11] A. Hatami and K. Pahlavan, "Hybrid toa-rss based localization using neural networks." in

GLOBECOM, 2006.

[12] K. Doğançay and H. Hmam, "Optimal angular sensor separation for aoa localization," *Signal Processing*, vol. 88, no. 5, pp. 1248–1260, 2008.

[13] P. Kułakowski, J. Vales-Alonso, E. Egea-López, W. Ludwin, and J. García-Haro, "Angle-of-arrival localization based on antenna arrays for wireless sensor networks," *Computers & Electrical Engineering*, vol. 36, no. 6, pp. 1181–1186, 2010.

[14] W. Qin, H. Jie, Z. Qianxiong, L. Bingfeng, and Y. Yanwei, "Ranging error classification based indoor toa localization algorithm [j]," *Chinese Journal of Scientific Instrument*, vol. 12, p. 034, 2011.

[15] K. Whitehouse, C. Karlof, and D. Culler, "A practical evaluation of radio signal strength for ranging-based localization," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 11, no. 1, pp. 41–52, 2007.

[16] M. Bouet and A. L. Dos Santos, "Rfid tags: Positioning principles and localization techniques," in *Wireless Days, 2008. WD'08. 1st IFIP*. IEEE, 2008, pp. 1–5.

[17] B.-S. Chen, C.-Y. Yang, F.-K. Liao, and J.-F. Liao, "Mobile Location Estimator in a Rough Wireless Environment Using Extended Kalman-Based IMM and Data Fusion," *Vehicular Technology, IEEE Transactions on*, vol. 58, no. 3, pp. 1157–1169, March 2009.

[18] K. Achutegui, L. Martino, J. Rodas, C. J. Escudero, and J. Miguez, "A multi-model particle filtering algorithm for indoor tracking of mobile terminals using rss data," in *Control Applications,(CCA) & Intelligent Control,(ISIC), 2009 IEEE*. IEEE, 2009, pp. 1702–1707.

[19] M. Heidari and K. Pahlavan, "A new statistical model for the behavior of ranging errors in toa-based indoor localization," in *Wireless Communications and Networking Conference, 2007. WCNC 2007. IEEE*. IEEE, 2007, pp. 2564–2569.

[20] S. Adler, T. Hillebrandt, S. Pfeiffer, J. Schiller, and H. Will, "Demo: Distance measurement in wireless sensor network with low cost components," in *Indoor Positioning and Indoor Navigation (IPIN 2011)*. IEEE.

[21] R. Kaune, J. Horst, and W. Koch, "Accuracy Analysis for TDOA Localization in Sensor Networks," in *Information Fusion (FUSION), 2011 Proceedings of the 14th International Conference on*. IEEE, 2011, pp. 1–8.

[22] Z. Yang and Y. Liu, "Quality of trilateration: Confidence-based iterative localization," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 21, no. 5, pp. 631–640, 2010.

[23] J. J. Robles, J. S. Pola, and R. Lehnert, "Extended Min-Max Algorithm for Position Estimation in Sensor Networks," in *Positioning Navigation and Communication (WPNC), 2012 9th Workshop on*. IEEE, 2012, pp. 47–52.

[24] H. Will and M. Hillebrandt, T.and Kyas, "The Geo-n Localization Algorithm," in *Indoor*

Positioning and Indoor Navigation (IPIN), 2012 International Conference on, 2012, pp. 1–10.

[25] J. Wang, P. Urriza, Y. Han, and D. Cabric, "Weighted Centroid Localization Algorithm: Theoretical Analysis and Distributed Implementation," *Wireless Communications, IEEE Transactions on*, vol. 10, no. 10, pp. 3403–3413, 2011.

[26] L. Haiyong, L. Hui, Z. Fang, and P. Jinghua, "An iterative clustering-based localization algorithm for wireless sensor networks," *China Communications*, vol. 8, no. 1, pp. 58–64, 2011.

[27] I. Guvenc, C.-C. Chong, and F. Watanabe, "Analysis of a Linear Least-Squares Localization Technique in LOS and NLOS Environments," in *Vehicular Technology Conference, 2007. VTC2007-Spring. IEEE 65th*. IEEE, 2007, pp. 1886–1890.

[28] S. Venkatesh and R. M. Buehrer, "A linear programming approach to nlos error mitigation in sensor networks," in *Proceedings of the 5th international conference on Information processing in sensor networks*. ACM, 2006, pp. 301–308.

[29] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A Tutorial on Particle filters for Online Nonlinear/non-Gaussian Bayesian Tracking," *Signal Processing, IEEE Transactions on*, vol. 50, no. 2, pp. 174–188, 2002.

[30] A. Doucet and A. Johansen, "A Tutorial on Particle Filtering and Smoothing: Fifteen Years Later," 2009.

[31] X. Sheng, Y. Hu, and P. Ramanathan, "Distributed Particle Filter with GMM Approximation for Multiple Targets Localization and Tracking in Wireless Sensor Network," in *Proceedings of the 4th international symposium on Information processing in sensor networks*. IEEE, 2005, p. 24.

[32] I. M. Rekleitis, "A particle filter tutorial for mobile robot localization," *Centre for Intelligent Machines, McGill University, Tech. Rep. TR-CIM-04-02*, 2004.

[33] D. Fox, "Kld-sampling: Adaptive particle filters," in *Advances in neural information processing systems*, 2001, pp. 713–720.

[34] D. Fox, S. Thrun, W. Burgard, and F. Dellaert, "Particle filters for mobile robot localization," in *Sequential Monte Carlo methods in practice*. Springer, 2001, pp. 401–428.

[35] C. Kwok, D. Fox, and M. Meila, "Adaptive real-time particle filters for robot localization," in *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, vol. 2. IEEE, 2003, pp. 2836–2841.

[36] A. Kushki, K. N. Plataniotis, and A. N. Venetsanopoulos, "Kernel-based Positioning in Wireless Local Area Networks," *Mobile Computing, IEEE Transactions on*, vol. 6, no. 6, pp. 689–705, 2007.

[37] Y. Song and H. Yu, "A New Hybrid TOA/RSS Location Tracking Algorithm for Wireless Sensor Network," in *Signal Processing, 2008. ICSP 2008. 9th International Conference*

on, Oct. 2008, pp. 2645–2648.

[38] M. Klepal, S. Beauregard *et al.*, "A backtracking particle filter for fusing building plans with pdr displacement estimates," in *Positioning, Navigation and Communication, 2008. WPNC 2008. 5th Workshop on*. IEEE, 2008, pp. 207–212.

[39] F. Evennou, F. Marx, and E. Novakov, "Map-aided indoor mobile positioning system using particle filter," in *Wireless Communications and Networking Conference, 2005 IEEE*, vol. 4. IEEE, 2005, pp. 2490–2494.

[40] H. Wang, H. Lenz, A. Szabo, J. Bamberger, and U. D. Hanebeck, "Wlan-based pedestrian tracking using particle filters and low-cost mems sensors," in *Positioning, Navigation and Communication, 2007. WPNC'07. 4th Workshop on*. IEEE, 2007, pp. 1–7.

[41] S. Beauregard, M. Klepal *et al.*, "Indoor pdr performance enhancement using minimal map information and particle filters," in *Position, Location and Navigation Symposium, 2008 IEEE/ION*. IEEE, 2008, pp. 141–147.

[42] R. Harle, "A survey of indoor inertial positioning systems for pedestrians," 2013.

[43] R. Faragher, C. Sarno, and M. Newman, "Opportunistic radio slam for indoor navigation using smartphone sensors," in *Position Location and Navigation Symposium (PLANS), 2012 IEEE/ION*. IEEE, 2012, pp. 120–128.

[44] Y. Zhao, Y. Yang, and M. Kyas, "Comparing Centralized Kalman Filter Schemes for Indoor Positioning in Wireless Sensor Network," in *Indoor Positioning and Indoor Navigation (IPIN), 2011 International Conference on*. IEEE, 2011, pp. 1–10.

[45] ——, "'Likelihood Adaptation of Particle Filter for Target Tracking using Wireless Sensor Networks," in *IEEE Globecom 2013*. IEEE, 2013.

[46] ——, "An Adaptive Likelihood Fusion Method using Dynamic Gaussian Model for Indoor Target Tracking," in *Internatinal Conference on Acoustics, Speech, and Signal Processing*. IEEE, 2014, pp. 1–4.

[47] ——, "Dynamic searching particle filtering scheme for indoor localization in wireless sensor network," in *Positioning Navigation and Communication (WPNC), 2012 9th Workshop on*. IEEE, 2012, pp. 65–70.

[48] ——, "PSG-DPF: Distributed Particle Filter using Pairwise Selective Gossiping for Wireless Sensor Network," in *Vehicular Technology Conference, 2013. VTC2013-Spring.* IEEE, 2013, pp. 1–4.

[49] ——, "2D Geometrical Performance for Localization Algorithms from 3D Perspective," in *Indoor Positioning and Indoor Navigation (IPIN), 2013 International Conference on*. IEEE, 2013, pp. 1–10.

[50] ——, "Cramér-Rao Lower Bound Analysis for Wireless Localization Systems using Priori Information," in *11th Workshop on Positioning, Navigation and Communication 2014 (WPNC'14)*. IEEE, 2014, pp. 1–10.

[51] L. Ong, B. Upcroft, M. Ridley, T. Bailey, S. Sukkarieh, and H. Durrant-Whyte, "Consistent Methods for Decentralised Data Fusion using Particle Filters," in *Multisensor Fusion and Integration for Intelligent Systems, 2006 IEEE International Conference on*. IEEE, 2006, pp. 85–91.

[52] R. Chou, Y. Boers, M. Podt, and M. Geist, "Performance Evaluation for Particle Filters," in *Information Fusion (FUSION), 2011 Proceedings of the 14th International Conference on*. IEEE, 2011, pp. 1–7.

[53] M. Vrettas, D. Cornford, and M. Opper, "Estimating Parameters in Stochastic Systems: A variational Bayesian approach," *Physica D: Nonlinear Phenomena*, 2011.

[54] J. Mandel, L. Cobb, and J. Beezley, "On the Convergence of the Ensemble Kalman Filter," *Arxiv preprint arxiv:0901.2951*, 2009.

[55] X. Hu, T. Schon, and L. Ljung, "A Basic Convergence Result for Particle Filtering," *Signal Processing, IEEE Transactions on*, vol. 56, no. 4, pp. 1337–1348, 2008.

[56] N. Bergman, "Recursive Bayesian Estimation," *Navigation and tracking applications Linkoping Studies in Science and Technology. Dissertations*, no. 579, 1999.

[57] J. Kotecha and P. Djuric, "Gaussian Particle Filtering," *Signal Processing, IEEE Transactions on*, vol. 51, no. 10, pp. 2592–2601, 2003.

[58] D. Crisan and A. Doucet, "A Survey of Convergence Results on Particle Filtering Methods for Practitioners," *Signal Processing, IEEE Transactions on*, vol. 50, no. 3, pp. 736–746, 2002.

[59] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P.-J. Nordlund, "Particle Filters for Positioning, Navigation, and Tracking," *Signal Processing, IEEE Transactions on*, vol. 50, no. 2, pp. 425–437, 2002.

[60] K. Okuma, A. Taleghani, N. De Freitas, J. J. Little, and D. G. Lowe, "A boosted particle filter: Multitarget detection and tracking," in *Computer Vision-ECCV 2004*. Springer, 2004, pp. 28–39.

[61] J. Carpenter, P. Clifford, and P. Fearnhead, "Improved particle filter for nonlinear problems," *IEE Proceedings-Radar, Sonar and Navigation*, vol. 146, no. 1, pp. 2–7, 1999.

[62] N. Chopin, "A sequential particle filter method for static models," *Biometrika*, vol. 89, no. 3, pp. 539–552, 2002.

[63] R. Douc and O. Cappé, "Comparison of resampling schemes for particle filtering," in *Image and Signal Processing and Analysis, 2005. ISPA 2005. Proceedings of the 4th International Symposium on*. IEEE, 2005, pp. 64–69.

[64] C. Morelli, M. Nicoli, V. Rampa, and U. Spagnolini, "Hidden markov models for radio localization in mixed los/nlos conditions," *Signal Processing, IEEE Transactions on*, vol. 55, no. 4, pp. 1525–1542, 2007.

[65] S. Farahmand, S. Roumeliotis, and G. Giannakis, "Set-membership constrained particle

filter: Distributed adaptation for sensor networks," *Signal Processing, IEEE Transactions on*, vol. 59, no. 9, pp. 4122–4138, 2011.

[66] M. K. Pitt and N. Shephard, "Filtering via simulation: Auxiliary Particle Filters," *Journal of the American statistical association*, vol. 94, no. 446, pp. 590–599, 1999.

[67] C. Musso, N. Oudjane, and F. Le Gland, "Improving Regularised Particle Filters," in *Sequential Monte Carlo methods in practice*. Springer, 2001, pp. 247–271.

[68] R. Van Der Merwe, A. Doucet, N. De Freitas, and E. Wan, "The Unscented Particle Filter," *Advances in Neural Information Processing Systems*, pp. 584–590, 2001.

[69] A. Doucet, N. De Freitas, K. Murphy, and S. Russell, "Rao-Blackwellised Particle Filtering for Dynamic Bayesian Networks," in *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 2000, pp. 176–183.

[70] S. Särkkä, A. Vehtari, and J. Lampinen, "Rao-Blackwellized Particle Filter for Multiple Target Tracking," *Information Fusion*, vol. 8, no. 1, pp. 2–15, 2007.

[71] Y. Qi, H. Kobayashi, and H. Suda, "Analysis of Wireless Geolocation in a Non-Line-Of-Sight Environment," *Wireless Communications, IEEE Transactions on*, vol. 5, no. 3, pp. 672–681, 2006.

[72] Q. Wang, S. R. Kulkarni, and S. Verdú, "Divergence estimation for multidimensional densities via-nearest-neighbor distances," *Information Theory, IEEE Transactions on*, vol. 55, no. 5, pp. 2392–2405, 2009.

[73] M. Nicoli, C. Morelli, and V. Rampa, "A Jump Markov Particle Filter for Localization of Moving Terminals in Multipath Indoor Scenarios," *Signal Processing, IEEE Transactions on*, vol. 56, no. 8, pp. 3801–3809, Aug. 2008.

[74] B. Turgut and R. P. Martin, "A multi-hypothesis particle filter for indoor dynamic localization," in *Local Computer Networks, 2009. LCN 2009. IEEE 34th Conference on*. IEEE, 2009, pp. 742–749.

[75] B. Turgut and R. Martin, "Restarting Particle Filters: an Approach to Improve the Performance of Dynamic Indoor Localization," in *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*. IEEE, 2009, pp. 1–7.

[76] L. Pishdad and F. Labeau, "Indoor positioning using particle filters with optimal importance function," in *Positioning Navigation and Communication (WPNC), 2012 9th Workshop on*. IEEE, 2012, pp. 77–82.

[77] P. Davidson, J. Collin, and J. Takala, "Application of particle filters for indoor positioning using floor plans," in *Ubiquitous Positioning Indoor Navigation and Location Based Service (UPINLBS), 2010*. IEEE, 2010, pp. 1–4.

[78] J. Pinchin, C. Hide, and T. Moore, "A particle filter approach to indoor navigation using a foot mounted inertial navigation system and heuristic heading information," in *Indoor*

Positioning and Indoor Navigation (IPIN), 2012 International Conference on. IEEE, 2012, pp. 1–10.

[79] C. Ascher, C. Kessler, M. Wankerl, and G. Trommer, "Dual imu indoor navigation with particle filter based map-matching on a smartphone," in *Indoor Positioning and Indoor Navigation (IPIN), 2010 International Conference on*. IEEE, 2010, pp. 1–5.

[80] J. Suutala, K. Fujinami, and J. Roning, "Persons tracking with gaussian process joint particle filtering," in *Machine Learning for Signal Processing (MLSP), 2010 IEEE International Workshop on*. IEEE, 2010, pp. 160–165.

[81] T. Hanselmann, Y. Zhang, M. Morelande, M. I. M. Nor, J. W. J. Tan, X.-S. Zhou, and Y. W. Law, "Self-localization in wireless sensor networks using particle filtering with progressive correction," in *Communications and Networking in China (CHINACOM), 2010 5th International ICST Conference on*. IEEE, 2010, pp. 1–6.

[82] A. Kushki, K. Plataniotis, and A. Venetsanopoulos, "Intelligent Dynamic Radio Tracking in Indoor Wireless Local Area Networks," *Mobile Computing, IEEE Transactions on*, vol. 9, no. 3, pp. 405–419, March 2010.

[83] J. Prieto, S. Mazuelas, A. Bahillo, P. Fernandez, R. M. Lorenzo, and E. J. Abril, "Adaptive Data Fusion for Wireless Localization in Harsh Environments," *Signal Processing, IEEE Transactions on*, vol. 60, no. 4, pp. 1585–1596, 2012.

[84] B. Khaleghi, A. Khamis, and F. Karray, "Multisensor Data Fusion: a Review of the State-of-the-Art," *Information Fusion*, 2011.

[85] R. Haenni and S. Hartmann, "Modeling Partially Reliable Information Sources: A General Approach based on Dempster–Shafer Theory," *Information fusion*, vol. 7, no. 4, pp. 361–379, 2006.

[86] D. Dubois and H. Prade, "Possibility Theory in Information Fusion," in *Information Fusion, 2000. FUSION 2000. Proceedings of the Third International Conference on*, vol. 1. IEEE, 2000, pp. PS6–P19.

[87] E. J. Wright and K. B. Laskey, "Credibility Models for Multi-Source Fusion," in *Information Fusion, 2006 9th International Conference on*. IEEE, 2006, pp. 1–7.

[88] A. Stordal, H. Karlsen, G. Nævdal, H. Skaug, and B. Vallès, "Bridging the Ensemble Kalman Filter and Particle Filters: the Adaptive Gaussian Mixture Filter," *Computational Geosciences*, vol. 15, no. 2, pp. 293–305, 2011.

[89] P. Blanchart, L. He, and F. Le Gland, "Information fusion for indoor localization," in *Information Fusion, 2009. FUSION'09. 12th International Conference on*. IEEE, 2009, pp. 2083–2090.

[90] V. Savic, A. Athalye, M. Bolic, and P. M. Djuric, "Particle filtering for indoor rfid tag tracking," in *Statistical Signal Processing Workshop (SSP), 2011 IEEE*. IEEE, 2011, pp. 193–196.

[91] L. Geng, M. F. Bugallo, and P. M. Djuric, "Tracking with rfid asynchronous measurements by particle filtering," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 4051–4055.

[92] W. Choi, C. Pantofaru, and S. Savarese, "A general framework for tracking multiple people from a moving camera," 2012.

[93] F. Zhao, J. Shin, and J. Reich, "Information-Driven Dynamic Sensor Collaboration for Tracking Applications," *IEEE Signal processing magazine*, vol. 19, no. 2, pp. 61–72, 2002.

[94] F. Zhao, J. Liu, J. Liu, L. Guibas, and J. Reich, "Collaborative Signal and Information Processing: An Information-Directed Approach," *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1199–1209, 2003.

[95] B. J. Julian, M. Angermann, M. Schwager, and D. Rus, "Distributed robotic sensor networks: An information-theoretic approach," *The International Journal of Robotics Research*, vol. 31, no. 10, pp. 1134–1154, 2012.

[96] T. Shima and S. J. Rasmussen, *UAV cooperative decision and control: challenges and practical approaches*. SIAM, 2009, vol. 18.

[97] H. Aghajan and A. Cavallaro, *Multi-camera networks: principles and applications*. Academic press, 2009.

[98] P. Corke, T. Wark, R. Jurdak, W. Hu, P. Valencia, and D. Moore, "Environmental wireless sensor networks," *Proceedings of the IEEE*, vol. 98, no. 11, pp. 1903–1917, 2010.

[99] J. Ko, C. Lu, M. B. Srivastava, J. A. Stankovic, A. Terzis, and M. Welsh, "Wireless sensor networks for healthcare," *Proceedings of the IEEE*, vol. 98, no. 11, pp. 1947–1960, 2010.

[100] T. Zhao and A. Nehorai, "Distributed sequential bayesian estimation of a diffusive source in wireless sensor networks," *Signal Processing, IEEE Transactions on*, vol. 55, no. 4, pp. 1511–1524, 2007.

[101] S. Farahmand, S. Roumeliotis, and G. Giannakis, "Particle filter adaptation for distributed sensors via set membership," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*. IEEE, 2010, pp. 3374–3377.

[102] O. Hlinka, F. Hlawatsch, and P. M. Djuric, "Distributed particle filtering in agent networks: A survey, classification, and comparison," *Signal Processing Magazine, IEEE*, vol. 30, no. 1, pp. 61–81, 2013.

[103] M. Coates, "Distributed Particle Filters for Sensor Networks," in *Proceedings of the 3rd international symposium on Information processing in sensor networks*. ACM, 2004, pp. 99–107.

[104] W. Gao, H. Zhao, C. Song, and J. Xu, "A New Distributed Particle Filtering for WSN Target Tracking," in *2009 International Conference on Signal Processing Systems*. IEEE, 2009, pp. 334–337.

[105] A. Dimakis, S. Kar, J. Moura, M. Rabbat, and A. Scaglione, "Gossip algorithms for distributed signal processing," *Proceedings of the IEEE*, vol. 98, no. 11, pp. 1847–1864, 2010.

[106] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *Information Theory, IEEE Transactions on*, vol. 52, no. 6, pp. 2508–2530, 2006.

[107] R. Olfati-Saber, J. Fax, and R. Murray, "Consensus and Cooperation in Networked Multi-Agent Systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[108] A. Mohammadi and A. Asif, "Consensus-based Distributed Unscented Particle Filter," in *Statistical Signal Processing Workshop (SSP), 2011 IEEE*. IEEE, 2011, pp. 237–240.

[109] S. Lee and M. West, "Markov Chain Distributed Particle Filters (MCDPF)," in *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*. IEEE, 2009, pp. 5496–5501.

[110] D. Gu, "Distributed Particle Filter for Target Tracking," in *Robotics and Automation, 2007 IEEE International Conference on*. IEEE, 2007, pp. 3856–3861.

[111] O. Hlinka, O. Sluciak, F. Hlawatsch, P. Djuric, and M. Rupp, "Distributed Gaussian Particle Filtering using Likelihood Consensus," in *Proc. ICASSP*. IEEE, 2011.

[112] B. Oreshkin and M. Coates, "Asynchronous Distributed Particle Filter via Decentralized Evaluation of Gaussian Products," in *Information Fusion (FUSION), 2010 13th Conference on*. IEEE, 2010, pp. 1–8.

[113] D. Ustebay, M. Coates, and M. Rabbat, "Distributed Auxiliary Particle Filters using Selective Gossip," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*. IEEE, 2011, pp. 3296–3299.

[114] G. Rogova and V. Nimier, "Reliability in Information Fusion: Literature Survey," in *Proceedings of the Seventh International Conference on Information Fusion*, 2004, pp. 1158–1165.

[115] T. Cover and J. Thomas, *Elements of Information Theory*. Wiley-interscience, 2006.

[116] G. Hendeby, R. Karlsson, F. Gustafsson, and N. Gordon, "Performance Issues in Non-Gaussian Filtering Problems," in *Nonlinear Statistical Signal Processing Workshop, 2006 IEEE*. IEEE, 2006, pp. 65–68.

[117] S. Boyd and L. Vandenberghe, "Convex Optimization." Cambridge Univ Pr, 2004, pp. 291–293.

[118] Y. Yang, Y. Zhao, and M. Kyas, "Weighted Least-squares by Bounding-box (B-WLS) for NLOS Mitigation of Indoor Localization," in *Proceedings of Vehicular Technology Conference 2013 spring*. IEEE, 2013.

[119] S. Mazuelas, F. Lago, J. Blas, A. Bahillo, P. Fernandez, R. Lorenzo, and E. Abril, "Prior NLOS Measurement Correction for Positioning in Cellular Wireless Networks," *Vehicular Technology, IEEE Transactions on*, vol. 58, no. 5, pp. 2585–2591, 2009.

[120] J. Prieto, A. Bahillo, S. Mazuelas, R. Lorenzo, J. Blas, and P. Fernandez, "NLOS Mitigation based on Range Estimation Error Characterization in an RTT-based IEEE 802.11 Indoor Location System," in *Intelligent Signal Processing, 2009. WISP 2009. IEEE International Symposium on*. IEEE, 2009, pp. 61–66.

[121] B. Alavi and K. Pahlavan, "Modeling of the toa-based distance measurement error using uwb indoor radio measurements," *Communications Letters, IEEE*, vol. 10, no. 4, pp. 275–277, 2006.

[122] S. Gezici, H. Kobayashi, and H. V. Poor, "Nonparametric Nonline-of-Sight Identification," in *Vehicular Technology Conference, 2003. VTC 2003-Fall. 2003 IEEE 58th*, vol. 4. IEEE, 2003, pp. 2544–2548.

[123] S. Venkatraman and J. Caffery Jr, "Statistical Approach to Non-Line-Of-Sight BS Identification," in *Wireless Personal Multimedia Communications, 2002. The 5th International Symposium on*, vol. 1. IEEE, 2002, pp. 296–300.

[124] L. Cong and W. Zhuang, "Nonline-Of-Sight Error Mitigation in Mobile Location," *Wireless Communications, IEEE Transactions on*, vol. 4, no. 2, pp. 560–573, 2005.

[125] K. Yu and Y. Guo, "Improved positioning algorithms for nonline-of-sight environments," *Vehicular Technology, IEEE Transactions on*, vol. 57, no. 4, pp. 2342–2353, 2008.

[126] V. Honkavirta, T. Perala, S. Ali-Loytty, and R. Piché, "A Comparative Survey of WLAN Location Fingerprinting Methods," in *Positioning, Navigation and Communication, 2009. WPNC 2009. 6th Workshop on*. IEEE, 2009, pp. 243–251.

[127] Y. Yang, Y. Zhao, and M. Kyas, "A non-parametric modeling of time-of-flight ranging error for indoor network localization," in *Global Communications Conference (GLOBECOM), 2013 IEEE*. IEEE, 2013, pp. 189–194.

[128] S. Schmitt, H. Will, B. Aschenbrenner, T. Hillebrandt, and M.Kyas, "A Reference System for Indoor Localization Testbeds," in *Internet Conference on Indoor Positioning and Indoor Navigation, IPIN 2012*, 2012, pp. 1–4.

[129] H. Hashemi, "The indoor radio propagation channel," *Proceedings of the IEEE*, vol. 81, no. 7, pp. 943–968, 1993.

[130] A. N. Bishop, B. Fidan, B. Anderson, K. Doğançay, and P. N. Pathirana, "Optimality Analysis of Sensor-Target Localization Geometries," *Automatica*, vol. 46, no. 3, pp. 479–492, 2010.

[131] H. Will, T. Hillebrandt, Y. Yuan, Z. Yubin, and M. Kyas, "The membership degree min-max localization algorithm," in *Ubiquitous Positioning, Indoor Navigation, and Location Based Service (UPINLBS), 2012*. IEEE, 2012, pp. 1–10.

[132] A. Chehri, P. Fortier, and P.-M. Tardif, "Geo-location with wireless sensor networks using non-linear optimization," *Proceedings of International Journal of Computer Science and Network Security (IJCSNS)*, pp. 145–154, 2008.

[133] Y. Yang, Y. Zhao, and M. Kyas, "A Statistics-based Least-squares (SLS) Method for Non-line-of-sight Error of Indoor Localization," in *Proc. IEEE WCNC'13*, 2013, pp. 1–6.

[134] M. St-Pierre and D. Gingras, "Comparison between the unscented kalman filter and the extended kalman filter for the position estimation module of an integrated navigation information system," in *Intelligent Vehicles Symposium, 2004 IEEE*. IEEE, 2004, pp. 831–835.

[135] S. Schmitt, "Aufbau und Evaluation eines virtuellen Testbeds zur Indoor Lokalisierung in drahtlosen Sensornetzwerken," 2013.

[136] D. Ustebay, R. Castro, and M. Rabbat, "Selective gossip," in *Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), 2009 3rd IEEE International Workshop on*. IEEE, 2009, pp. 61–64.

[137] ——, "Efficient Decentralized Approximation via Selective Gossip," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 5, no. 4, pp. 805–816, 2011.

[138] J. Lu, C. Tang, P. Regier, and T. Bow, "A Gossip Algorithm for Convex Consensus Optimization over Networks," no. 99. IEEE, 2011, pp. 1–15.

[139] S. Dulman, P. Havinga, A. Baggio, and K. Langendoen, "Revisiting the Cramer-Rao Bound for Localization Algorithms," *4th IEEE/ACM DCOSS Work-in-progress paper*, 2008.

[140] Y. Qi, "Wireless Geolocation in a Non-Line-Of-Sight Environment," Ph.D. dissertation, Princeton University, 2003.

[141] R. B. Langley, "Dilution of Precision," *GPS world*, vol. 10, no. 5, pp. 52–59, 1999.

[142] B. Yang and J. Scheuing, "Cramer-Rao Bound and Optimum Sensor Array for Source Localization from Time Differences Of Arrival," in *Acoustics, Speech, and Signal Processing, 2005. Proceedings.(ICASSP'05). IEEE International Conference on*, vol. 4. IEEE, 2005, pp. iv–961.

[143] P. Tichavsky, C. Muravchik, and A. Nehorai, "Posterior Cramer-Rao Bounds for Discrete-Time Nonlinear Filtering," *Signal Processing, IEEE Transactions on*, vol. 46, no. 5, pp. 1386–1396, 1998.

[144] L. Zuo, R. Niu, and P. K. Varshney, "Conditional Posterior Cramer-Rao Lower Bounds for Nonlinear Sequential Bayesian Estimation," *Signal Processing, IEEE Transactions on*, 2010.

[145] A. Mohammadi and A. Asif, "Distributed Posterior Cramer-Rao Lower Bounds for Nonlinear Sequential Bayesian Estimation," *IEEE 7th Sensor Array and Multichannel Signal Processing Workshop (SAM)*, 2012.

[146] Y. Shen and M. Z. Win, "Fundamental Limits of Wideband Localization: Part I A General Framework," *Information Theory, IEEE Transactions on*, 2010.

[147] E. Arias-de Reyna and P. M. Djuric, "Indoor Localization with Range-based Measure-

ments and Little Prior Information," *IEEE Sensors Journal*, 2013.

[148] S. O. Dulman, A. Baggio, P. J. Havinga, and K. G. Langendoen, "A Geometrical Perspective on Localization," in *Proceedings of the first ACM international workshop on Mobile entity localization and tracking in GPS-less environments*. ACM, 2008, pp. 85–90.

[149] P.-H. Tseng and K.-T. Feng, "Geometry-Assisted Localization Algorithms for Wireless Networks," *Mobile Computing, IEEE Transactions on*, 2013.

[150] A. M. Hossain and W.-S. Soh, "Cramer-Rao Bound Analysis of Localization using Signal Strength Difference as Location Fingerprint," in *INFOCOM, 2010 Proceedings IEEE*. IEEE, 2010, pp. 1–9.

[151] Y. Ling, S. Alexander, and R. Lau, "On Quantification of Anchor Placement," in *INFOCOM, 2012 Proceedings IEEE*. IEEE, 2012, pp. 2192–2200.

[152] H. L. Van Trees, "Optimum Array Processing (Detection, Estimation, and Modulation Theory, Part IV)," *Wiley-Interscience, Mar*, no. 50, p. 100, 2002.

[153] H. So, "Source Localization: Algorithms and Analysis," *Handbook of Position Location: Theory, Practice, and Advances*, pp. 25–66, 2011.

List of Figures

2.1 The working procedure of particle filter. 13

3.1 Example: the two likelihood functions. The solid curve represents the actual likelihood function obtained without measurement error and the dash curve represents the deviated likelihood function by the instantaneous measurement error 31
3.2 The architecture of particle filter integrated with adaptive likelihood method. The common particle filter does not contain the prior measurement and the belief factor. 32
3.3 Network Deployment and Target Trajectory. The triangle marks the positions of sensor nodes and the dash line marks the target's trajectory 40
3.4 Estimation Error Comparison for Different Algorithms with different measurement error variances 42
3.5 Optimal θ comparison between the simulation results and our calculation based on (3.31) 43
3.6 Root mean square error (RMSE) comparison for different algorithms with different number of particles 44
3.7 Root mean square error (RMSE) comparison for different algorithms with different ranging error variances 45
3.8 Root mean square error (RMSE) comparison for different algorithms with different number of particles 46
3.9 Root mean square error (RMSE) comparison for different algorithms with different number of anchors 47

4.1 The Hybrid LOS/NLOS Indoor Environment. 50
4.2 Indoor ranging error fitting 55
4.3 General Gaussian model fitting for other distributions and statistical histogram. 57
4.4 Dynamic Gaussian Modeling for Indoor Ranging Error. 58

5.1 TOF working procedure 63

5.2 System Architecture 64
5.3 Region partition: Type I: room or corridor without cross; Type II: corridor on the cross. 74
5.4 The constraint conditions drawn by min-max algorithm 75
5.5 The joint constraint region. Three types: type I, the constraint regions are overlapped; type II, part of the constraint regions are overlapped; type III, the constraint regions are significantly different. 76
5.6 Building layout for indoor localization experiment and the robot trajectory. The triangles mark the positions of sensor nodes which are placed either in the offices or along the corridor. 77
5.7 Trajectories in the hybrid scenario. 79
5.8 CDF of the estimation errors. 80
5.9 Trajectories in the building, with 17 anchors deployed. 81
5.10 Trajectories in the building, with 25 anchors deployed 82
5.11 CDF of the estimation errors. 82
5.12 Estimation performance changes with different number of particles 83
5.13 Estimation performance changes with different number of measurement samples 84

6.1 Distributed network model 89
6.2 Algorithm structure of PSG-DPF 91
6.3 Estimation errors for four particle filters by adapting standard derivation of measurement noise. 97
6.4 Estimation errors by adapting number of gossiping particles. 98
6.5 Gossip iterations of the PSG-DPF 99

7.1 Simulation trajectory with 10 anchors 114
7.2 Cramér-Rao lower bound for target tracking 115
7.3 Cramér-Rao lower bound with multiple measurement noise environment 116
7.4 Cramér-Rao lower bound with multiple priori information 117
7.5 Cramér-Rao lower bound with different number of anchors 118
7.6 Demonstration of 3D-ranging. 119
7.7 CRLB Analysis Results 123
7.8 CRLB analysis with different $\triangle z$ and 3 anchors 125

List of Tables

4.1 Ranging error fitting 55

5.1 Estimation Performance comparison (hybrid) 79
5.2 Estimation Performance comparison (building) 83

6.1 Estimation Performance Comparison for the Algorithms 99

List of Symbols

$D_{KL}(p_1||p_2)$ The Kullback-Leibler divergence
$I_s(\boldsymbol{x}_t^i)$ Importance density function
K_t Kalman gain
N_s The total number of the particles
P_t^- Prior estimation error
P_t Posterior estimation error
T_{RTT} Round trip time
$[X_{min}^k, X_{max}^k, Y_{min}^k, Y_{max}^k]^T$ The coordinates of the kth layout constraint region
$\boldsymbol{F}_t$ The Transition matrix
$\boldsymbol{I}(\boldsymbol{x}_t)$ Fisher information matrix
$\boldsymbol{I}_D(\boldsymbol{x}_t)$ Fisher information matrix based on the measurement data
$\boldsymbol{I}_P(\boldsymbol{x}_t)$ Priori information
$\boldsymbol{Q}_t$ The covariance of the prediction error
$\bar{\boldsymbol{x}}_t$ The estimated state
$\hat{\boldsymbol{x}}_t$ The predicted state
$\hat{\boldsymbol{z}}_t$ The predicted measurement
$\boldsymbol{\mu}_t$ The expectation of the Gaussian distribution
$\boldsymbol{\theta}$ Belief factor
$\boldsymbol{a}_j = [a_j^X, a_j^Y]^T$ The coordinates of the anchor position
$\boldsymbol{q}_t$ Prediction noise
$\boldsymbol{x}_t^i = [p_t^{X,i}, p_t^{Y,i}]^T$ The particle sample coordinate
$\boldsymbol{x}_t^i$ Particle sample
$\boldsymbol{x}_t = [p_t^X, p_t^Y]^T$ The coordinates of the target position
$\boldsymbol{x}_t$ The target position state
$\boldsymbol{z}_t$ Measurement vector
$\mathcal{E}_t$ The set of the network edges
$\mathcal{G}_t(\mathcal{V}_t, \mathcal{E}_t)$ Time-varying graph
$\mathcal{V}_t = \{1, \ldots, n\}$ The set of the network nodes

π_v The probability density function
$\triangle z$ Relative height
$c(\boldsymbol{x}_t, \boldsymbol{z}_t)$ The constraint function
f_t Prediction function
$p(\boldsymbol{x}_t|\boldsymbol{z}_{1:t})$ Sequential posterior probability density function
$p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ Transition probability
$p(\boldsymbol{z}_t|\boldsymbol{x}_t)$ Measurement likelihood function
$p(\boldsymbol{z}_t|\boldsymbol{z}_{1:t})$ Normalized constant
p_A The actual assumed distribution
p_{AL} Adaptive likelihood function
v_t^j Local measurement noise for anchor j
w_t^i Associate weight for the particle $\boldsymbol{x}_t^i$
z_t^j The local measurement for the anchor j

Glossary

A-BPF adaptive bootstrap particle filter
A-CPF adaptive constraint particle filter
A-GPF adaptive Gaussian particle filter
AE average error

BPF bootstrap particle filter

CA-PF context aware particle filter
CPF constraint particle filter
CRLB Cramér-Rao lower bound

DGM dynamic Gaussian model
DPF distributed particle filter

EKF extended Kalman filter
EM expectation-maximization algorithm

FIM Fisher information matrix

GDOP geometrical dilution of precision
GMM Gaussian mixture model
GPF Gaussian particle filter
GPS global positioning system

IMU inertial measurement unit

KLD Kullback-Leibler divergence
KS-DPF K-selective particle filter

LF likelihood function
LLS linear least squares method

MCDPF Markov chain distributed particle filter

MCMC Markov chain Monte-Carlo method
ME median error
ML maximum likelihood

NLLS nonlinear least squares method
NLOS non-line-of-sight

PDF probability density function
PF particle filter
PSG pariwise selective gossip
PSG-DPF pariwise selective gossip distributed particle filter

RBPF Rao-Blackwellised particle filter
RMeSE root median square error
RMSE root mean square error
RPF regularized particle filter
RSS received signal strength
RTT round trip time

S-DAPF selective distributed auxiliry particle filter
SIS sequential importance sampling

TDOA time-difference-of-arrival
TOA time-of-arrival
TOF time-of-flight

UKF unscented Kalman filter
UPF unscented particle filter

WSN wireless sensor network

About the Author

For reasons of data protection, the curriculum vitae is not included in the online version