Localization and Object-Tracking in an Ultrawideband Sensor Network

by

Cheng Chang

Chang, Cheng (M.S., Electrical Engineering)

Localization and Object-Tracking in an Ultrawideband Sensor Network

Thesis directed by Prof. Anant Sahai

Geometric information is essential for sensor networks. We study two kinds of geometric information. One is the positions of the nodes. The estimation of the positions of the nodes is called the localization problem. The other is the positions of objects in the sensor network.

For the localization problem, We will study the Cramer Rao lower bound on it. For the anchor-free localization problem where no nodes have known positions, we propose a new bound on the variance of the estimation error, because the Fisher Information Matrix is singular. For the anchored localization problem using only local information, we derive a lower bound to the Cramer Rao bound on the position estimation. We find that the Cramer Rao bounds in both cases are invariant under zooming of the whole sensor network. We will also propose a novel two-step localization scheme. In the first step, we estimate an anchor-free coordinate system around every node. In the second step, we combine all the anchor-free coordinate systems together. Then using the anchored position information of some nodes, we transfer the anchor-free coordinate system into an anchored coordinate system.

For the object position estimation problem, we study it in different scenarios in terms of number of nodes. There are three scenarios: single transmitter and single receiver, multiple transmitter (receiver) and single receiver (transmitter), multiple transmitter and multiple receiver. For each scenario, we give a position estimation scheme and analyze the performance of our scheme. The Cramer Rao bound for each scenario is also computed. We are particularly interested in the behavior of the Cramer Rao bound when the number of sensors in the network grows to infinity. We find that

the Cramer Rao bound on object tracking is proportional to the reciprocal of the total received SNR.

Dedication

To my parents.

Acknowledgements

I 'd like to thank my advisor, Prof. Anant Sahai, for introducing me this exciting project and for his patience with me. I'd also like to thank Prof. Kannan Ramchandran, Prof. David Tse, Dr. Massimo Franceschetti, Dan Hazen, Hansen Bow, Lenny Grokop, Moshe Malkin, Jiangang Yao and Gabor Pete for their help and inspirational discussions.

Contents

Chapter

1 Introduction 1

1.1 Localization in a Sensor Network 3

1.1.1 Related Work 3

1.1.2 Overview 4

1.2 Tracking Objects by a UWB Sensor Network 8

1.2.1 Overview 8

1.2.2 Related Work 11

2 Localization 13

2.1 Problem Formulation 13

2.1.1 Anchored Localization Problem 14

2.1.2 Anchor-free Localization Problem 15

2.2 Cramer-Rao Lower Bound on Localization 16

2.2.1 The Cramer-Rao bound on Anchored Localizaion 16

2.2.2 The Cramer-Rao bound on Anchor-free Localization 30

2.3 2-Step Anchored Localization Scheme 45

2.3.1 Step 1 : Anchor-free Coordinates Estimation 46

2.3.2 Step 2 : Anchored Coordinates Estimation 54

2.3.3 Summary and further discussion 55

2.3.4 Simulation Results 58
2.4 Conclusions and Future Work 62

3 Tracking Objects by a Sensor Network 64
3.1 Tracking Single Object in a Single Transmitter, Single Receiver Sensor Network (STSR) 64
3.1.1 A Simple Estimation Scheme 65
3.1.2 Cramer-Rao bound analysis 70
3.1.3 A Consistent Estimation Scheme 76
3.2 Single Object Tracking in a Multiple-sensor Sensor Network (MSSO) . . 81
3.2.1 Cramer-Rao Bound 82
3.2.2 A Position Estimation Scheme 86
3.2.3 MSSO in a sensor network with a single receiver 95
3.2.4 Cramer-Rao bound for the outside region 111
3.2.5 A linear algorithm for a small sensor network 124
3.3 Multiple Object Tracking by a Multiple-Sensor Sensor network (MSMO) 130
3.3.1 An Exhaustive Search Algorithm 131
3.3.2 Hough Transform Inspired Algorithm 132
3.4 Conclusions and Future Work 143

Bibliography 145

Appendix

A Uniqueness of the positions of a point set 148

B Proof of Eqn.2.25 150

C An Example of M/K Equivalence Class Estimation 154

D There is no subspace in which parameter estimation can be done for the anchor-free coordinate estimation problem 155

E Non-convexity of Eqn.2.65 158

F Multiple local minimum of Eqn.2.69 of variable d_{12} 159

G The bias of the ML anchor-free coordinate estimation 161

H The optimization problem in Eqn. 2.83 164

I STSR Object Tracking of Constant Acceleration Motion 166

J Degenerate Cases for STSR 170

K Convergence of $\chi(A_N)$ 173

L Confusing Paths for STSR 175

M Stability of the Cost function 181

N Synchronization and Distance Estimation in a Sensor Network 185

Figures

Figure

1.1 A wireless sensor network 1

1.2 Multi-path measurement 9

2.1 A sensor network with partial position information 13

2.2 Direct path distances 14

2.3 Geometric illustration of α_{ij} 17

2.4 Definition of N_l 24

2.5 Bounds on anchored localization 1 27

2.6 Bounds on anchored localization 2 28

2.7 CR bound on anchor-free localization 1 40

2.8 CR bound on anchor-free localization 2 41

2.9 CR bound on anchor-free localization 3 42

2.10 CR bound on anchor-free localization 4 43

2.11 CR bound on anchor-free localization 5 44

2.12 Estimation of anchor-free coordinate 49

2.13 Flowchart of localization algorithm 57

2.14 Localization simulation result 1 59

2.15 Localization simulation result 2 60

2.16 Localization simulation result 3 61

3.1 Single transmitter single receiver object tracking 65
3.2 Single transmitter single receiver object tracking with motion model . . 66
3.3 Condition Number Plot 69
3.4 α_t and β_t 72
3.5 STSR Cramer-Rao Bound Plot1 74
3.6 STSR Cramer-Rao Bound Plot2 75
3.7 Flowchart of a consistent scheme 77
3.8 An illustration of MSSO 81
3.9 α, CR bound for STSR 84
3.10 Theoretical CR bound vs. Experimental CR bound 86
3.11 Flowchart of MSSO position estimation 87
3.12 MSSO simulation result 1 93
3.13 MSSO simulation result 2 94
3.14 MTSR, N >> 1 96
3.15 Theoretical CR bound vs Experimental CR bound 98
3.16 MTSR, N ∼1 98
3.17 Setup 1 of the sensors 102
3.18 Setup 2 of the sensors 103
3.19 Setup 3 of the sensors 103
3.20 Positions of the test points 104
3.21 Cramer-Rao Bound Plot for a single-receiver network 1 (no motion model) 104
3.22 Cramer-Rao Bound Plot for a single-receiver network 1 (motion model) 105
3.23 Cramer-Rao Bound Plot for a single-receiver network 2(no motion model) 106
3.24 Cramer-Rao Bound Plot for a single-receiver network 2 (motion model) 107
3.25 Cramer-Rao Bound Plot for a single-receiver network 3(no motion model) 108
3.26 Cramer-Rao Bound Plot for a single-receiver network 4(no motion model) 109
3.27 Cramer-Rao Bound Plot for some points 110

3.28 The object is far away from the sensor network 112
3.29 Normalized Cramer-Rao Bound for far away points (Euclidean metric) . 115
3.30 The object is far away from the sensor network (Polar coordinates) . . 116
3.31 Normalized Cramer-Rao Bound for far away points (Polar coordinates) . 120
3.32 Normalized Cramer-Rao Bound (Polar coordinate) $(-10, 10) \times (-10, 10)$ 121
3.33 Normalized Cramer-Rao Bound (Polar coordinate)$(-3, 3) \times (-3, 3)$ . . . 122
3.34 The geometric interpretation of α 123
3.35 2 setups of 2 transmitter 2 receiver network 125
3.36 Cramer-Rao bounds of 2 transmitter 2 receiver network 126
3.37 Condition number of A for 2 transmitter 2 receiver network 127
3.38 Estimation Error for 2 Tx, 2 Rx network 129
3.39 An illustration of MSMO 130
3.40 Simulation result of MSMO 1 137
3.41 Simulation result of MSMO 2 138
3.42 Flowchart of MSMO position estimation 139
3.43 Discretized error 140

A.1 Uniqueness of the positions of a point set 149

C.1 An example of M/K Equivalence Class Estimation 154

D.1 Setup of the point sets as counter examples 156
D.2 Setup of the point sets as counter examples 157

E.1 Non-convexity of Eqn.2.65 158

F.1 Multiple minimum of Eqn.2.65 160

G.1 Setup of the sensors 161

I.1 Condition Number Plot 2 169

J.1 Degenerate cases for STSR 171

L.1 Good pair 176

L.2 Confusing paths 178

L.3 S(t) and S'(t) 180

M.1 Sensor setup 182

M.2 Cost Function 183

M.3 Score function 184

Chapter 1

Introduction

With the recent development in wireless communication devices and low power electronics systems, it is possible to have a wireless network consisting of nodes equipped with sensors and antennas, i.e. a sensor network [19]. The nodes of the sensor network can work both independently and cooperatively. The possible tasks for sensor network are extremely broad, ranging from communication to sensing physical variables. In Fig.1.1, we have a sensor network consisted of 7 nodes, each node has a sensing unit and a communication system. Notice that the connectivity graph is not always complete due to the power limits or inter-sensor interferences [16].

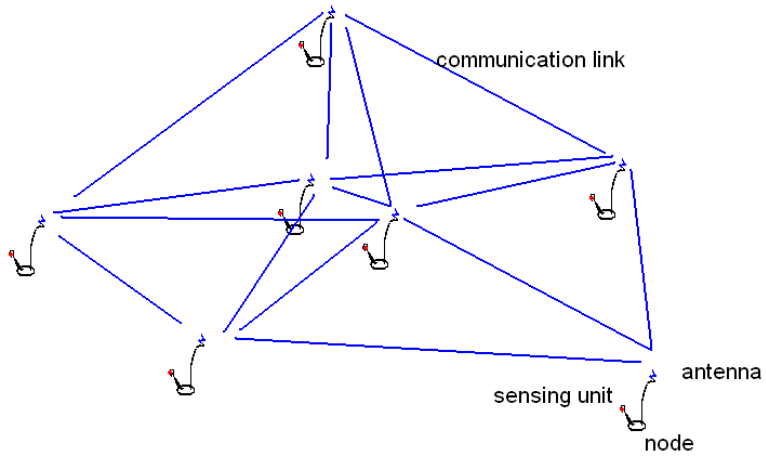


Figure 1.1: A wireless sensor network

Ultra Wideband (UWB)is a relatively new technology. The basic concept is to

communicate using very wide bandwidth signals which spread the signal energy over frequency and time[26]. Since UWB signals have extremely wide bandwidth, the signals can provide very precise distance measurements as long as the transmitter and receiver are synchronized. Communication in UWB requires tracking the channels in coherent communication schemes. From the channel estimates, we can extract distance measurements. The distance measurements are free since no extra efforts are needed. There are two kinds of distance measurements. The first kind is direct path distance measurements, which tells the receiver how far away the transmitter is located. The second kind is multi-path (transmitter-object-receiver) distance measurements, which happen if there are objects that reflect the signals.

In a sensor network equipped with UWB devices, we can extract direct path and multi-path distance information and tell where the nodes are (localization problem) and where the surrounding objects are (object tracking problem).

1.1 Localization in a Sensor Network

The localization problem is both fundamental and important. In many sensor network tasks, the location awareness of the nodes is one of most important system parameters [19]. No matter if the high level task is routing, parameter estimation or tracking other objects in the sensor network, to understand the geometrical setup of the sensor network is very helpful if not crucial. In chapter 2, we are going to study the localization problem. More precisely, we are going to study the problem of position estimation of the nodes, given direct-path distance measurements between nodes and possibly locations of some nodes which are variously called anchors [6] or beacons[23] in the literature.

1.1.1 Related Work

Localization has been studied extensively. There are three major category of localization schemes that vary by what kind of geometric information they need. The first one is based only on the connectivity information [23], [6], [29], [28] and [32], i.e. based on if node i can directly communicate with node j, or anchor k. All the algorithms have two steps, the first step is to get geometric information, specifically the distances between nodes and anchors, from connectivity information. And the second step is to determine the position of the nodes from the geometric information extracted from step 1. In order to convert connectivity information into geometric information on 2D different algorithms are studied in those papers, DV-Distance (first estimate an average HOP length between anchors, then compute the distance from a node to anchor by multiplying the hop counts) and similar ideas in [28],[6] [32] and [29]. In [23] connectivity information is converted into convex constraints on geometric entities. A thorough comparison of the localization schemes in [28]and [6] can be found in [24].

The second category is localization based on both Euclidean distance information

and angular information. It is studied in [22],[27] and [33], the singularity of the Fisher Information matrix is mentioned in [27] for the case where there is not enough global geometric information. The estimation of angular information needs special antenna elements which makes this kind of localization schemes less practical.

The third category is localization based solely on the Euclidean distances between nodes and between nodes and anchors [35], [7], [4] and [39]. The first two schemes estimate the coordinates of the nodes based on the measured or estimated distances between the nodes and the anchors directly. The latter two schemes first estimate the anchor-free coordinate system then embed the anchor-free coordinate system into the coordinate system with anchors. In [20] and [15], the schemes of estimating Euclidean distances are discussed. In our report we will propose a 2-step anchored localization scheme based on Euclidean distances. The first step is to estimate an anchor-free coordinate system. There's a similar scheme in [4]. The second step is to combine the anchor-free coordinate systems and eventually combine all anchor-free coordinate systems with an anchored coordinate system, thus every node gets its own position in the anchored coordinate system. How to combine two coordinates together is also studied in [4] and [39]. We are going to focus on this kind of localization problem in our report.

Localization is indeed an estimation problem. In [1] the Cramer-Rao lower bounds are calculated for several specific geometric setups. And as for anchor-free coordinate estimation where there is no anchors, as mentioned in [27], the Fisher information matrix is singular. Thus the standard Cramer-Rao bound analysis fails as mentioned in [30].

1.1.2 Overview

In a wireless sensor network, there are two kinds of nodes. One is nodes with known position (anchors), the positions are possibly from GPS devices[33]. The second kind is nodes without known positions. In general, the density of nodes without known position can be much higher than the density of anchors because GPS equipment can be

expensive and calibration can be time consuming. The localization problem is to recover the positions of the nodes without known positions with the help of the positions of the anchors and some geometric relations among nodes and anchors.

In a UWB sensor network, the sensors are capable of sampling the received signals at a very high sampling rate. And if the whole sensor network is synchronized, it is possible to measure the distances between any two nodes. To illustrate, we present a simple distance measure scheme as following. At a preassigned time t_i which is also known by the whole sensor network, only node i sends out a signal while all other nodes listen. The signal arrives at node j in the range of node i at time t_i^j. Then the distance d_{ij} between the ith node and the jth node can be calculated at node j, $c(t_i^j - t_i)$, where c is the speed of light. $t_i^j - t_i$ is well known as TOA (time of arrival) in the radar literature [5]. Here we assume that we have a perfect clock at each node. We leave the synchronization issues to Appendix N and will give a linear estimation scheme which estimates the offsets of clocks and distances simultaneously.

The power and communication ability in each sensor node is very limited. So a distributed localization scheme is more preferable than a centralized scheme. However, it is easier to analyze the localization performance bound in the centralized flavor. Suppose that there is a central computer collecting all those distances, our goal is to design a scheme for the central computer to estimate to 2-d positions of each node. Given observations of distances between nodes, the underlying parameters to be estimated are the positions of the nodes. Notice that with only distances between the nodes, it is impossible to determine the anchored 2-d positions of the nodes. It turns out that at least 3 nodes with known anchored 2-d positions are needed as reviewed in Appendix A, also shown in [33].

In order to better understand the nature of the localization problem, we study the Cramer-Rao lower bound on the localization problem in Section 2. The Cramer-Rao [34] bound is a lower bound on the error variance of any unbiased estimators and is widely

used to tell how difficult an estimation problem is. We will calculate the Cramer-Rao lower bound for two different kinds of localization problems. The first is the **Anchored Localization Problem** in which we have the the knowledge of the anchored positions of at least 3 nodes. In this case, the goal is to estimate the anchored positions of all nodes. The second is the **Anchor-free Localization Problem** where no universal positions are known. Thus only relative positions can be estimated. It turns out that the nature of the Cramer-Rao bound on the two problems are quite different.

For the Anchored Localization problem, we will show that the Cramer-Rao bound is invariant under zooming. Then we will give a lower bound on the Cramer-Rao bound. To compute this lower bound, only local (neighbor) information is needed. We observe that it converges to Cramer-Rao bound if the local area is expanded which means that we can use local geometry to predict the accuracy of the position estimation. And we find that this lower bound is inversely proportional to the number of its neighbor nodes. For the anchor-free localization problem, we will first show that the Fisher Information Matrix is singular, thus the standard Cramer-Rao bound analysis does not work. We will explain why the Fisher Information matrix is singular given the geometry of the localization problem based on the work in [36] and propose a new estimation framework. This framework tries to explain the singularity of a class of estimation problems. Based on our framework, we derived a Cramer-Rao-like bound on anchor-free coordinate estimation. It turns out that this bound is also highly dependent on the average number of neighbor nodes, which again implies that local information can well predict the accuracy of the position estimation.

Then in section 3 we will give our 2-step anchored localization algorithm. The first step is to find an anchor-free coordinate system. This can be done distributedly at each node. We are using a scheme similar to [4]. The scheme here is based on ML estimation. The second step is to combine the anchor-free coordinate systems and eventually combine all anchor-free coordinate systems with an anchored coordinate

system generated by the anchors. Thus every node gets its own position in the anchored coordinate system. How to combine two coordinates together is also studied in [4] and [39]. We propose a different scheme. The key difference from the their schemes is that our scheme is optimal under the MSE criteria. Finally we will analyze the performance of our algorithm.

1.2 Tracking Objects by a UWB Sensor Network

The positions of objects in the wireless sensor environment is useful information for its own sake. Furthermore, the communication channel is largely determined by the objects in the sensor network field. Tracking the position of the objects could simplify the channel estimation problem, which has the potential to benefit the communication system. In chapter 3 we will study the object tracking problem. More precisely, we will study how to track the position of objects given the locations of the sensors and the multi-path (transmitter-object-receiver) distances reflected by the objects.

1.2.1 Overview

In a UWB sensor network where transmitters and receivers have known locations, it is possible to estimate the location of the objects in the field. Similar to the simple distance measure model of the localization problem, transmitters can send out impulses with very high bandwidth. If there are objects presented in the field, the received signal is a combination of direct-path signal and reflected signal from the object. If the A/D converters of the receivers have a very high sampling rate, it is possible to estimate the multi-path distances of the reflection.

A simple example from real experimental data is as shown in Fig.1.2, the receiver compares the received signal $A(t)$ (the green curve in Fig1.2) with the control signal $B(t)$ (the blue curve in Fig1.2). The difference is the multi-path reflection from the objects $C(t)$ (the red curve in Fig1.2). $t_d = Td - Tr$ is the estimated time difference between the direct transmission of the impulse and the multi-path reflection of the impulse, where Td, Tr are the TOA (time of arrival) of direct path and multi-path signals respectively. The distance difference between the multi-path and the direct path is $t_d c$, where c is the speed of light. We assume that the locations of the sensors are known, and thus the distance between the transmitter and the receiver d_s. So the multi-path length is

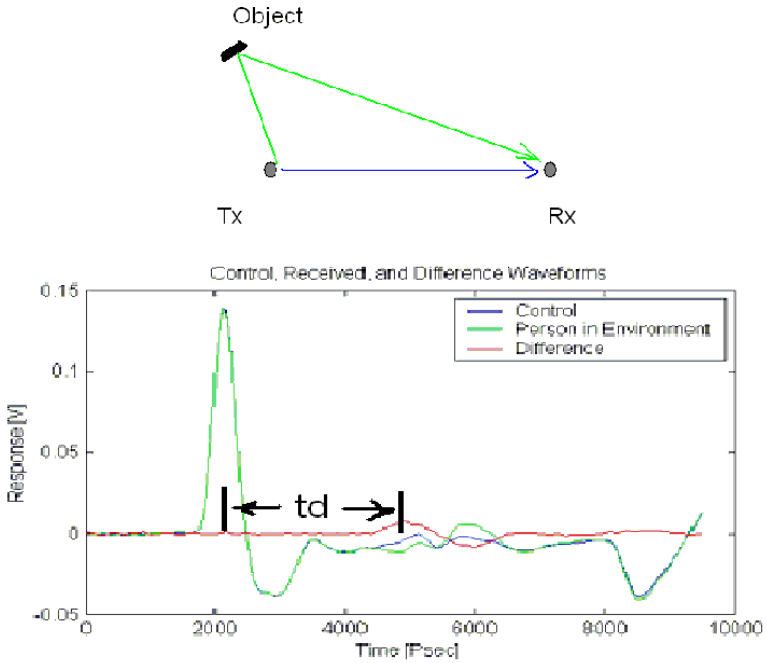


Figure 1.2: Estimation of the length of Multi-path

$d_s + t_d c$. The estimation of the TOA of the direct path signal is a well known problem in radar literature. The Cramer-Rao bound of any unbiased estimation of Td is [10]:

$$\delta_{Td}^2 \geq \frac{N_0}{\int_0^T [\frac{\partial A(t)}{\partial t}]^2 dt} \tag{1.1}$$

Where T is the observation interval, N_0 is the noise power spectral density, δ_{Td}^2 is the estimation variance of Td. Similarly the estimation variance of Tr

$$\delta_{Tr}^2 \geq \frac{N_0}{\int_0^T [\frac{\partial C(t)}{\partial t}]^2 dt} \tag{1.2}$$

Assuming the estimation of Td, Tr are independent, the Cramer-Rao bound on the estimation variance of t_d is $\delta_{Tr}^2 + \delta_{Td}^2$. Thus the CR bound on estimation variance of the multipath distance is $c^2(\delta_{Tr}^2 + \delta_{Td}^2)$. The dominant term tends to be δ_{Tr}^2 since it has a lower energy.

In this report, both the sensors and the objects are assumed to be points on a 2-D plane, and the positions of the sensors are known. And from the discussion above, the multi-path distances are observed with some estimation error. The goal is to estimate the position of the object(s). We will discuss the object tracking (position estimation) problem in different scenarios. Given different number of transmitters, receivers, objects and different prior knowledge of the motions of the objects, we are facing different challenges. For each scenario, we will derive a performance bound assuming the distance measures are corrupted by iid Gaussian noises. Then we will give an object tracking algorithm and analyze it. All algorithms are presented in a centralized computing flavor.

First in 3.2 we will study the single transmitter, single receiver case, where a motion model is a must to track the object because if the object can move with arbitrary velocity, then two different motions could give the same multipath distance measure all the time. So we assume the motion is strictly linear with constant velocity. After showing that the problem is ill-conditioned by showing that the Cramer-Rao lower bound of the estimation problem is huge, we argue that multi-sensor is needed in order to make stable estimation of the position of the object(s).

Then in 3.3 we will study a sensor network with multiple transmitters and multiple receivers, and also we will give some analysis on a multiple transmitter, single receiver network. We will argue that with more sensors, the object can be more accurately tracked in the sense of a decreasing Cramer-Rao lower bound. We will also study the asymptotic properties of the Cramer-Rao bounds when the number of sensors grows to infinity. We find that with uniformly distributed transmitters and receivers, the asymptotic Cramer-Rao bound is inversely proportional to the of total received SNR. And when the object is far away from the sensor field, the Cramer-Rao bound in Euclidean coordinates increases proportionally to the square of distance between the object and the sensors, even if the distance measurements maintain the same accuracy. Meanwhile in polar coordinates, the Cramer-Rao bound remains constant as the object moves far away. We also give a semi-linear algorithm which is order optimal when the number of sensors is big. We will also study the object tracking problem in a small sensor network, especially a two transmitter, two receiver network. We find that the tracking performance for different sensor placements are very different in a small sensor network.

Finally in 3.4, we will explore the objects tracking problem when there are more than 1 objects in a multiple transmitter, multiple receiver sensor network. We will propose a two step algorithm. The first step is inspired by the Hough Transform, where we associate the multipath measurements with the objects, then we use the semi-linear algorithm for single object tracking to estimate the positions of the objects. Our algorithm gives a satisfactory estimation performance.

1.2.2 Related Work

In multi-static radar literature, the position estimation and object tracking problem are also studied. In [5], two position estimation schemes are discussed, a one-stage scheme, and a two-stage scheme. In the one-stage scheme, all the receivers send the

entire received signal to a fusion center, and the fusion center estimates the position of the object. In the two-stage scheme, the first stage receivers estimate some geometric features (TOA, AOA,...) related to the object. Then they send the geometric features to a fusion center, which estimates the position of the object according to the geometric features. Position estimation using TOA or TDOA (time difference of Arrival) in a radar network has been studied in [5] and [2], and the accuracy is discussed in [12]. The achievable accuracy (Cramer-Rao bound) of the position estimation is also studied in [5] where it is shown that 2-stage estimation achieves the same estimation variance as 1-stage if the noises are white Gaussian, and the estimation in the first stage is unbiased and efficient.

Recently, position estimation of objects in a sensor network or in an indoor region has been studied in [38]. Position estimation in a sensor network is quite similar as the position estimation problem in multi-static Radar literature. The differences are, for sensor network, the communication capacity is much more limited [16] than multi-static radar. Thus the receivers cannot send the entire received signal to a fusion center. The receivers have to estimate some geometric feature related to the position of the object, then send the features to the fusion center to do the final estimation. In this report, the geometric features are transmitter-object-receiver multi-path distance measures. Furthermore, it is very difficult to equip a sensor network with Doppler radars or antenna arrays which can collect AOA (angle of arrival) information, so we mainly consider position estimation based on the TOA (time of arrival) information alone. The position estimation in a sensor network can be thought as a 2-stage position estimation problem using TOA information only. Multistatic radar systems are also usually sparsely deployed. So one thing that is not extensively studied in multi-static radar literature is the asymptotic behavior, i.e. the Cramer-Rao bound on the position estimation where a dense radar network exists. By contrary sensors can be densely deployed in the field, and thus the asymptotic behavior is interesting.

Chapter 2

Localization

2.1 Problem Formulation

In this section we give the mathematical model of the localization problem in both anchored localization and anchor-free localization scenarios.

On a 2-D plane, M wireless sensors with unknown positions are present, forming a set S. Also, N wireless sensors with known positions are present, forming a set namely F. Because the size of each sensor is assumed to be very small, we treat each sensor as a point. An illustration is in Fig.2.1.

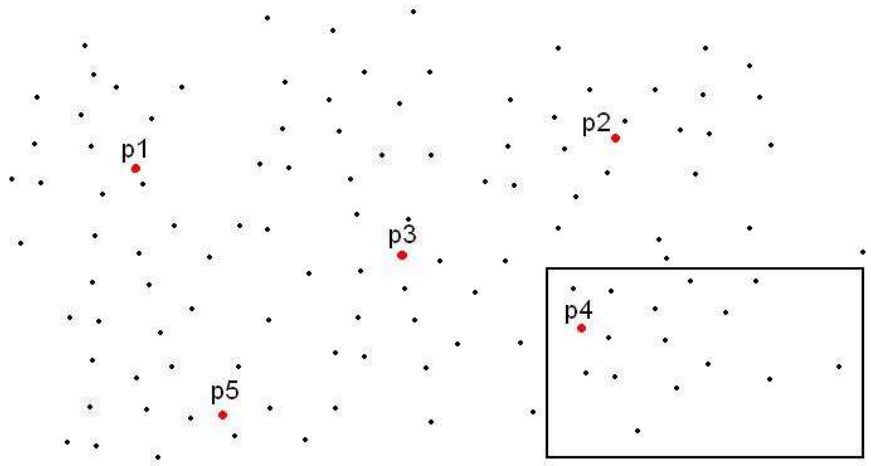


Figure 2.1: A sensor network. $F = \{p1, p2, p3, p4, p5\}, M = 5$, black points form S.

Each wireless sensor can generate limited-power signals, through which, a sensor

i can measure the distance to some nearby sensors , the set is called $adj(i)$. In general, node i can measure the distance between node j and i does not necessarily mean that node j can measure the distance between node j and i as well. However if sensor i and j can communicate with each other , then we can always assume the distance between i and j is known to both. Also, if i and j get different distance measures, we can always properly average the two measurements since we assume the noises are independent zero mean Gaussian. So we assume that $j \in adj(i)$ iff $i \in adj(j)$. Part of the sensor network in Fig.2.1 is shown in Fig.2.2.

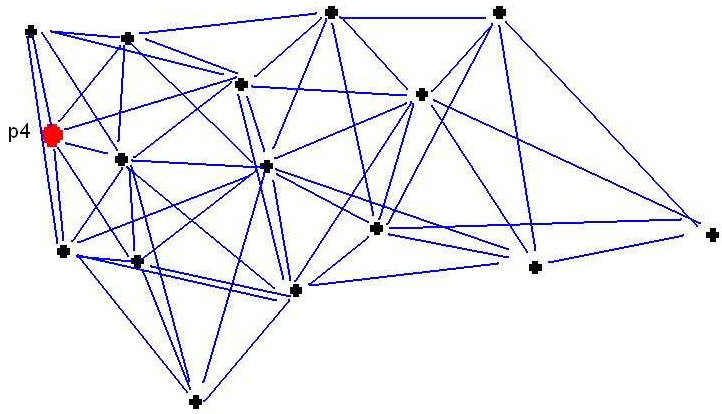


Figure 2.2: The rectangular region in Fig.2.1. Sensor i is linked to j,iff $j \in adj(i)$ with a blue line.

2.1.1 Anchored Localization Problem

If $|F| \geq 3$, i. e. there are three or more nodes with known position, then it's possible to estimate the anchored coordinate for each node. Thus we have observation D and position knowledge P_F.

$$D = \{d_{i,j} | i \in S \cup F, j \in adj(i)\} \tag{2.1}$$

$$P_F = \{(x_i, y_i)^T | i \in F\} \tag{2.2}$$

Our goal is to reconstruct the set

$$P_S = \{(x_i, y_i)^T | i \in S\} \tag{2.3}$$

(x_i, y_i) is the position of a single sensor i. $d_{i,j}$ is the measured distance between sensor i and j. $d_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} + \epsilon_{i,j}$, where $\epsilon_{i,j}$'s are modelled as independent additive Gaussian noises $\sim N(0, \sigma_{i,j}^2)$. For simplicity, we often assume $\sigma_{i,j}^2 = \sigma^2$, i. e. $\epsilon_{i,j}$'s are iid $\sim N(0, \sigma^2)$.

2.1.2 Anchor-free Localization Problem

If $|F| = 0$, no nodes have known position. Thus we only have

$$D = \{d_{i,j} | i \in S, j > i, j \in adj(i)\} \tag{2.4}$$

$$\tag{2.5}$$

Then it is impossible to estimate the anchored coordinate for any nodes. Our goal is to reconstruct the set

$$P_S = \{(x_i, y_i)^T | i \in S\} \tag{2.6}$$

With only distance information, it is only possible to estimate an anchor-free coordinate system as illustrated in Appendix A. If $P_S = \{(x_i, y_i)^T | i \in S\}$ is an estimation of the positions of the nodes, then $P'_S = \{R(\alpha)(x_i, y_i)^T + (a, b)^T | i \in S\}$ is equivalent to P_S. Because they have the same distances between any two nodes. Where

$$R(\alpha) = \begin{pmatrix} cos(\alpha) & -sin(\alpha) \\ sin(\alpha) & cos(\alpha) \end{pmatrix} \quad \text{or} \quad \begin{pmatrix} cos(\alpha) & -sin(\alpha) \\ -sin(\alpha) & -cos(\alpha) \end{pmatrix} \tag{2.7}$$

Thus the performance measure of an anchor-free localization estimation should not be $\sum (x - \hat{x})^2 + (y - \hat{y})^2$. Instead we will define the distance between equivalence classes in Section 2.2.2 and it best depicts the performance of an anchor-free coordinate estimation.

Same as the anchored localization problem the distance measures are assumed to be corrupted by iid additive Gaussian noises $\sim N(0, \sigma^2)$.

2.2 Cramer-Rao Lower Bound on Localization

In this section, we give the Cramer-Rao bound [8]on the localization problem. There are 2 difference scenarios here, **Anchored localization** and **Anchor-free localization**. The first scenario is that if there are three or more sensors with known positions, then the localization problem can be defined as a traditional parameter estimation problem. Anchor-free localization is if there are no sensors have known positions, i.e. all we know is the distance measures between sensors. Then the problem can not be defined as a traditional parameter estimation problem. We define an equivalence class estimation problem, and the anchor-free localization falls right in that category. In this section we are interested in defining the Cramer-Rao bound problem for anchor-free localization than will calculate the Cramer-Rao bound in section 2.2.2.

2.2.1 The Cramer-Rao bound on Anchored Localizaion

First we give some general results on Cramer-Rao lower bound. We first review the geometric treatment of Cramer-Rao bound introduced by Steve Smith in [36].

Theorem 2.1. *Cramer-Rao Let $f(x|\theta)$ be a family of pdfs parameterized by $\theta \in \psi$, where ψ is an n dimensional manifold. Let $l = log(f)$ be the log-likelihood function, and Fisher information matrix J, $J_{ij} = E[(\partial l/\partial\theta_i)(\partial l/\partial\theta_j)]$. Given arbitrary coordinates $\theta = (\theta_1, ..., \theta_n)^T \in R^n$ on ψ, then for any unbiased estimator $\hat{\theta}$ of θ,*

$$C \geq J^{-1} \tag{2.8}$$

Where $C = E[(\theta - \hat{\theta})(\theta - \hat{\theta})^T]$ is the covariance matrix of $\hat{\theta} - \theta$ and J is the Fisher information matrix with respect to these coordinates.

Here matrix $n \times n$ dimensional matrix $A \geq B$ means : $A - B$ is positive semi-definite.

2.2.1.1 CR bound on Anchored localization

In this section we derive the Fisher Information matrix for the anchored localization problem and numerically compute the CR bound for some particular setups of sensors.

As illustrated in 2.1.1. The position of M sensor are unknown , as $S = \{(x_i, y_i), i = 1, ...M\}$. The position of N sensors are known, as $F = \{(x_i, y_i), i = M+1, ...N+M = L\}$. The distances between sensor $i, i = 1, ...N$ and each sensor in $adj(i) \subseteq S \cup F$ are measured , but corrupted by Gaussian noises. We assume the Gaussian noises are i.i.d with zero mean and variance σ^2. So we have the observations

$$d_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} + \delta_{i,j} \tag{2.9}$$

Where $j \in adj(i)$,and $\delta_{i,j}$ is assumed to be equal to $\delta_{j,i}$, thus $d_{i,j} = d_{j,i}$. So without loss of generality, we assume $i < j$. We define the α_{ij} the angle from node i to j as following. $\alpha_{ij} \in [0, 2\pi)$ and

$$cos(\alpha_{ij}) = \frac{x_j - x_i}{\sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}}; sin(\alpha_{ij}) = \frac{y_j - y_i}{\sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}}; \tag{2.10}$$

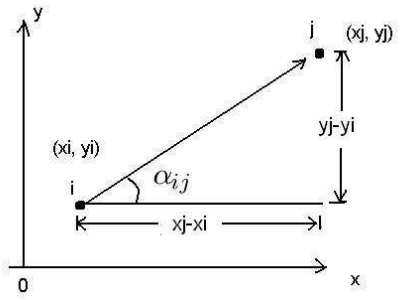


Figure 2.3: α_{ij}

The geometric interpretation of α_{ij} is illustrated in Fig.2.3. Now let the observation vector $\vec{d} = \{d_{i,j}, i < j, i \leq M, j \in adj(i)\}$. And the parameters we want to estimate

are $(x_i, y_i), i = 1, ..M$. Let x_i, y_i be the $2i-1$'th and $2i$'th parameters to be estimated respectively ,$i = 1, 2, ...M$. The Fisher information matrix is $J_{2M\times 2M}$. Then we have the following theorem.

Theorem 2.2. *Fisher Information Matrix for Anchored Localization* $\forall i = 1, ...M$

$$J_{2i-1,2i-1} = \frac{1}{\sigma^2}\sum_{j\in adj(i)} cos(\alpha_{ij})^2 \tag{2.11}$$

$$J_{2i,2i} = \frac{1}{\sigma^2}\sum_{j\in adj(i)} sin(\alpha_{ij})^2 \tag{2.12}$$

$$J_{2i-1,2i} = J_{2i,2i-1} = \frac{1}{\sigma^2}\sum_{j\in adj(i)} cos(\alpha_{ij})sin(\alpha_{ij}) \tag{2.13}$$

Above are the "diagonal" entries of the Fisher Information matrix J. For $j \neq i$, *if* $j \in adj(i)$

$$J_{2i-1,2j-1} = J_{2j-1,2i-1} = -\frac{1}{\sigma^2}cos(\alpha_{ij})^2 \tag{2.14}$$

$$J_{2i,2j} = J_{2j,2i} = -\frac{1}{\sigma^2}sin(\alpha_{ij})^2 \tag{2.15}$$

$$J_{2i-1,2j} = J_{2j,2i-1} = J_{2i,2j-1} = J_{2j-1,2i} = -\frac{1}{\sigma^2}sin(\alpha_{ij})cos(\alpha_{ij}) \tag{2.16}$$

If $j \notin adj(i)$*:* $J_{2i-1,2j-1} = J_{2j-1,2i-1} = J_{2i,2j} = J_{2j,2i} = J_{2i-1,2j} = J_{2j,2i-1} = J_{2i,2j-1} = J_{2j-1,2i} = 0$.

Proof. : Following the standard Fisher Information matrix calculation. We have the conditional pdf function:

$$p(\vec{d}|x_i, y_i, i = 1, ...M) = \prod_{i\leq M, i<j, j\in adj(i)} \frac{1}{\sqrt{2\pi\sigma^2}} exp(\frac{-(d_{ij} - \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}}{2\sigma^2}) \tag{2.17}$$

Log-likelihood function

$$ln(p(D|x_i, y_i, i = 1, ...M)) = C - \frac{1}{2\sigma^2}\sum_{i\leq M, i<j, j\in adj(i)} (d_{i,j} - \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2})^2 \tag{2.18}$$

Then the $2i-1, 2i-1$'th entry of J is

$$\begin{aligned} J_{2i-1,2i-1} &= E(\frac{\partial^2 ln(p(D|x_i, y_i, i=1,...M))}{\partial x_i^2}) \\ &= \frac{1}{\sigma^2} \sum_{j \in adj(i)} (\frac{x_j - x_i}{\sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}})^2 \\ &= \frac{1}{\sigma^2} \sum_{j \in adj(i)} cos(\alpha_{ij})^2 \end{aligned} \tag{2.19}$$

Similarly for other entries of J. □

We define $K = \sigma^2 J$, where K is independent of σ^2, we will call K the normalized Fisher information matrix. It is comparable to the GDOP (Geometric Dilution of Precision) in the radar literature [5]. As the GDOP in the radar literature, K is a dimensionless value and it only depends on the angles α_{ij}'s.

Given the expression of the Fisher Information matrix, we can evaluate the Cramer-Rao bound on the position estimation. Suppose the unbiased estimate of $(x_i, y_i), i = 1, ...N$ is $(\hat{x_i}, \hat{y_i})$. From Theorem 2.1.

$$E((\hat{x_i} - x_i)^2) \geq J^{-1}_{2i-1,2i-1} \tag{2.20}$$

$$E((\hat{y_i} - y_i)^2) \geq J^{-1}_{2i,2i} \tag{2.21}$$

Obviously , the normalized Fisher Information matrix is a function of $(x_i, y_i), i = 1, ...N + M$, written as $K((x_i, y_i), i = 1, ...N + M)$.

Corollary 2.1. *The normalized Fisher Information Matrix K is invariant under zooming and translation $K((x_i, y_i), i = 1, ...N + M) = K((ax_i, ay_i) + (c, d), i = 1, ...N + M)$,$a \neq 0, c, d \in R$.*

Proof. : Notice that the normalized Fisher Information matrix $K((x_i, y_i), i = 1, ...N + M)$ is determined by α_{ij}, and the following two point sets $(x_i, y_i), i = 1, ...N + M) = K((ax_i, ay_i) + (c, d), i = 1, ...N + M)$,$a \neq 0, c, d \in R$ yield the same $\alpha_{ij}, \forall i, j$. So $K((x_i, y_i), i = 1, ...N + M) = K((ax_i, ay_i) + (c, d), i = 1, ...N + M)$. □

2.2.1.2 A lower bound on the Cramer-Rao lower bound of anchored localization

From the previous section, we know how to compute the Cramer-Rao bound on the anchored localization problem. However, as can be seen, in order to evaluate the Cramer-Rao lower bound, we need to take the whole sensor network into account. In this section, we are going to derive a performance bound on the estimation of node l at (x_l, y_l) only based on the local geometry around it.

First we have some general results for estimation variance. Suppose $\theta \in R^n$ is the parameter to be estimated, and if the Fisher information matrix $J(\theta)$ is non-singular, thus it's positive definite. We have the following theorem.

Theorem 2.3. *A lower bound on the Cramer-Rao lower bound*
Let $\theta = (\theta_1, \theta_2, ...\theta_N) \in R^N$, $\forall M, 1 \leq M < N$, *write* $\theta^* = (\theta_{N-M+1}, ...\theta_N)$, *then for any unbiased estimator for* θ,

$$E((\theta^* - \hat{\theta^*})^T(\theta^* - \hat{\theta^*})) \geq C^{-1} \tag{2.22}$$

Where C *is an* $(N-M) \times (N-M)$ *matrix :*

$$J(\theta) = \begin{pmatrix} A & B \\ B^T & C \end{pmatrix} \tag{2.23}$$

$J(\theta)$ *is the Fisher Information matrix for* θ.

Proof. : Write the inverse of $J(\theta)$ as :

$$J(\theta)^{-1} = \begin{pmatrix} A' & B' \\ B'^T & C' \end{pmatrix} \tag{2.24}$$

$J(\theta)$ is positive definite, then

$$C' \geq C^{-1} \tag{2.25}$$

The proof is in Appendix B.

And from Theorem 2.1, $E((\theta^* - \hat{\theta^*})^T(\theta^* - \hat{\theta^*})) \geq C' \geq C^{-1}$. □.

Notice that for any subset of M parameters, we can always change the index of the parameter to make them have index $N - M + 1, ...N$.

Corollary 2.2. *A lower bound on the Cramer-Rao bound*

Write $\theta_l = (x_l, y_l)^T$ *and*

$$J_l = \frac{1}{\sigma^2} \begin{pmatrix} J(\theta)_{2l-1,2l-1} & J(\theta)_{2l-1,2l} \\ J(\theta)_{2l,2l-1} & J(\theta)_{2l,2l} \end{pmatrix} \tag{2.26}$$

Then for any unbiased estimator $\hat{\theta}$. $E((\hat{\theta}_l - \theta_l)(\hat{\theta}_l - \theta_l)^T) \geq J_l^{-1}$.

Proof. : This is directly from Theorem 2.3. □

Corollary 2.3. J_l *is only dependent on* (x_l, y_l) *and* $(x_i, y_i), i \in adj(l)$. *In other words, we can give a performance bound on the estimation of* (x_l, y_l) *using only the geometries of sensor* l*'s neighbors (sensor* j*'s, s.t.* $j \in adj(l)$*).*

Proof. : Only need to notice that the expressions on the entries of J_l in Eqn.3.104 is :

$$J_{l1,1} = \frac{1}{\sigma^2} \sum_{j \in adj(l)} cos(\alpha_{lj})^2 \tag{2.27}$$

$$J_{l2,2} = \frac{1}{\sigma^2} \sum_{j \in adj(l)} sin(\alpha_{lj})^2 \tag{2.28}$$

$$J_{l2,1} = J_{l1,2} = \frac{1}{\sigma^2} \sum_{j \in adj(l)} cos(\alpha_{lj}) sin(\alpha_{lj}) \tag{2.29}$$

So J_l only depends on $\alpha_{lj}, j \in adj(l)$. And α_{lj} only depends on (x_l, y_l) and (x_i, y_i). □

Notice that the result holds if the estimation variance on $d_{i,l}, i \in adj(l)$ are different. Let the variance be $\sigma_{i,l}^2$. Then

$$J_{l1,1} = \sum_{j \in adj(l)} \frac{1}{\sigma_{k,l}^2} cos(\alpha_{lj})^2 \tag{2.30}$$

Following the similar arguments as in the proof of Corollary 2.3, we know that $J_{l1,1}$ only depends on (x_l, y_l) and $(x_i, y_i), i \in adj(l)$, similarly for $J_{l2,1}$ $J_{l1,2}$ and $J_{l2,2}$.

Let $\sigma_{lj}^2 = \sigma^2$ again and let $adj(l) = W$, and sensors $\in adj(l)$ are $l(1), ..., l(k), ..l(W)$. With elementary trigonometry and write $\alpha_k = \alpha_{l,l(k)}$, we have :

$$J_l = \frac{1}{\sigma^2} \begin{pmatrix} \frac{W}{2} + \frac{\sum_{k=1}^{W} cos(2\alpha_k)}{2} & \frac{\sum_{k=1}^{W} sin(2\alpha_k)}{2} \\ \frac{\sum_{k=1}^{W} sin(2\alpha_k)}{2} & \frac{W}{2} - \frac{\sum_{k=1}^{W} cos(2\alpha_k)}{2} \end{pmatrix} \tag{2.31}$$

The sum of the estimation variance

$$\begin{aligned} & E((x_l - \hat{x_i})^2 + (y_l - \hat{y_i})^2) \geq {J_l^{-1}}_{11} + {J_l^{-1}}_{22} \\ & = \frac{4W\sigma^2}{W^2 - (\sum_{k=1}^{W} cos(2\alpha_k))^2 - (\sum_{k=1}^{W} sin(2\alpha_k))^2} \geq \frac{4\sigma^2}{W} \end{aligned} \tag{2.32}$$

It takes equality when $\sum_{k=1}^{W} sin(2\alpha_k) = 0, \sum_{k=1}^{W} cos(2\alpha_k) = 0$. This happens if the mass center of the unit vectors $(cos(2\alpha_k), sin(2\alpha_k))$'s is at the origin $(0, 0)$. A special case is when $\alpha_k = \frac{2k\pi}{W} + \beta$. In fact if the angles $2\alpha_k$'s are iid uniformly distributed in $[0, 2\pi)$, $E(cos(\alpha_i)cos(\alpha_j)) = E(sin(\alpha_i)sin(\alpha_j)) = \delta(i-j)$, where $\delta(0) = 1$ and $\delta(k) = 0$ if $k \neq 0$. By the law of large numbers,

$$\begin{aligned} & lim_{W\to\infty} W({J_l^{-1}}_{11} + {J_l^{-1}}_{22}) = lim_{W\to\infty} \frac{4W^2\sigma^2}{W^2 - (\sum_{k=1}^{W} cos(2\alpha_k))^2 - (\sum_{k=1}^{W} sin(2\alpha_k))^2} = \\ & lim_{W\to\infty} \frac{4\sigma^2}{1 - (\frac{1}{W}\sum_{k=1}^{W} cos(2\alpha_k))^2 - (\frac{1}{W}\sum_{k=1}^{W} sin(2\alpha_k))^2} \longrightarrow 4\sigma^2 \quad \text{a.s.} \end{aligned} \tag{2.33}$$

Eqn.2.33 shows that ${J_l^{-1}}_{11} + {J_l^{-1}}_{22}$ converges to 0 with the rate of lower bound in Eqn.2.32. Also notice that when $W = 1$, the denominator is 0, thus the CR bound is infinity.

In the above analysis, we used **one-hop** information around node i to compute a bound on the Cramer-Rao bound on the estimation of (x_i, y_i). In fact we can use **multi-hop** information to get a tighter bound on the Cramer-Rao bound. This can be easily proved by Corollary B.1 in Appendix B. For the Cramer-Rao bound of the anchored localization problem. The larger the local region we use to calculate the Cramer-Rao bound , the tighter it is.

2.2.1.3 Bound on the variance of estimation error with connectivity information

On the other hand, we can take the connectivity information into account as was done in [23],[6],[29],[28] and [32]. The connectivity information can be transferred into geometric constraints in the following way. Let R be the threshold, s.t. $\sqrt{(x_i - x_l)^2 + (y_i - y_l)^2} \leq R$ iff $i \in adj(l)$, $\sqrt{(x_i - x_l)^2 + (y_i - y_l)^2} > R$ iff $i \notin adj(l)$. R is called the visible radius.

We can treat the connectivity information as constraints. Write $C((x, y), R) = \{(u, v) | \sqrt{(u - x)^2 + (v - y)^2} \leq R\}$. If $\infty > L \geq |adj(l)| \geq 1$, then write $T_l = (\cap_{j \in adj(l)} C((x_j, y_j), R)) \cap (\cap_{j \notin adj(l), j \neq l} C((x_j, y_j), R)^c)$. The Borel set T_L is the region where (x_l, y_l) could be. It is bounded and the T_l can be covered by disc of radius R. Let ∂O be the boundary of a Borel set O, then let $N_l = \{j \neq l | \partial C((x_j, y_j), R) \cap \partial T_l \neq \emptyset\}$. It is illustrated in Fig.2.4. Notice that by the definition of N_l, $\forall j \in N_l$, $\sqrt{(x_j - x_l)^2 + (y_j - y_l)^2} \leq 3R$. We know that the estimation variance of $(x_l, y_l)$is lower bounded by the estimation problem of (x_l, y_l) given $\{(x_i, y_i), i =\in N_l \cup adj(l)\} \cup \{b_i, i \in N_l \cup adj(l)\}$. i.e, we just found a new estimation problem for (x_l, y_l) with lower estimation variance. And the new estimation problem only involves with the information of nearby sensors with distances no larger than $3R$.

The standard Cramer-Rao bound analysis does not apply to the estimation problem with connectivity information being taken into account because we impose the constraint $(x_l, y_l) \in T_l$ to the estimation problem. Thus the Cramer-Rao bound analysis no longer holds because the support set of the conditional pdf is not the whole space[8].

With large variance of distance measurement, the Cramer-Rao bound computed solely from the distance information is no longer a valid lower performance bound. Because given T_l, the estimation variance is upper bounded by $R(T_l)^2$, where $R(T_l)$ is

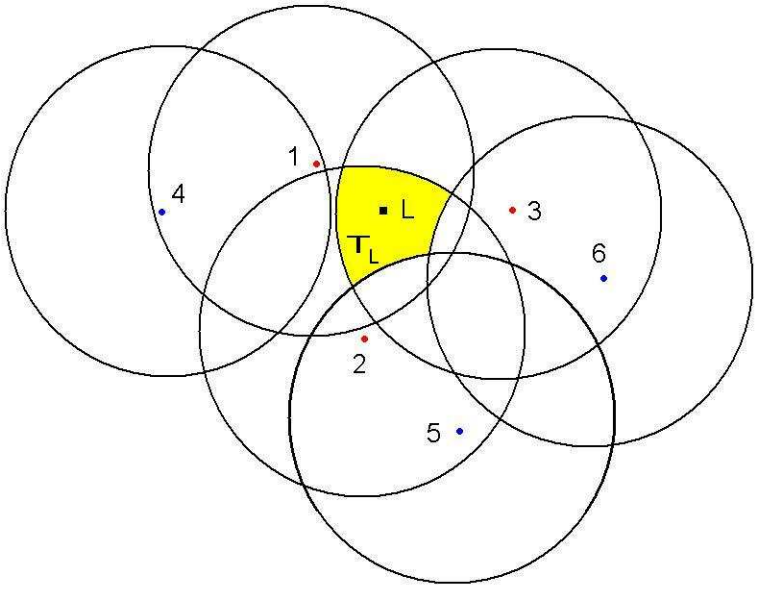


Figure 2.4: Illustration of T_L and N_L
The yellow region is T_l ,$adj(l) = \{1, 2, 3\}$, $4, 5, 6 \notin adj(l)$, while $N_l = \{2, 3, 5, 6\}$
If node l is inside $C((x_2, y_2), R) \cap C((x_3, y_3), R) \cap C((x_5, y_5), R)^c \cap C((x_6, y_6), R)^c$, then node l is destined to $\in C((x_1, y_1), R) \cap C((x_4, y_4), R)^c$

the minimum radius of a disc which covers T_l. Notice that $R(T_l)$ is solely dependent on the topology of the sensor network, it's independent with the distance measure variance σ^2. Meanwhile the Cramer-Rao bound computed solely using the distance information $\geq \frac{\sigma^2}{W}$. It can easily exceed $R(T_l)^2$ with a large σ^2.

When the variance of distance measurements is small comparing to $R(T_l)$, however, we have the following observation.

Theorem 2.4. *T_l can most likely be determined by distance measurements: Assuming $(x_l, y_l) \notin \partial T_l$, then $\forall$ non-collinear setups of $(x_i, y_i), i \in adj(l)$, $\forall \epsilon > 0$, $\exists \Omega > 0$, s. t. if $\sigma^2 < \Omega$, then we can find a region S_l solely based on $\{(x_i, y_i), d_{l,i}, i \in adj(l)\}$, s.t. $(x_l, y_l) \in S_l \subseteq T_l$ with probability $1 - \epsilon$.*

In other words, if the distance measures $d_{l,i}$'s are accurate enough, then $\forall i$, we can tell if $i \in adj(l)$ with high confidence.

Proof. : Define S_l as following:

$$S_l = \cap_{j \in adj(l)} \{(x, y) | \sqrt{(x - x_j)^2 + (y - y_j)^2} \in (d_{l,j} - D, d_{l,j} + D)\} \tag{2.34}$$

Where D is big enough to make

$$\begin{aligned} &Pr(\sqrt{(x_l - x_j)^2 + (y_l - y_j)^2} \in (d_{l,j} - D, d_{l,j} + D)) = \\ &Pr(d_{l,j} \in (\sqrt{(x_l - x_j)^2 + (y_l - y_j)^2} - D, \sqrt{(x_l - x_j)^2 + (y_l - y_j)^2} + D)) = \\ &\int_D^D \frac{e^{-\frac{z^2}{2\sigma^2}}}{\sqrt{2\pi\sigma^2}} dz \geq 1 - \frac{\epsilon}{|adj(l)|} \end{aligned} \tag{2.35}$$

With Ω goes to 0, and $\sigma^2 < \Omega$, D can be arbitrarily small, then S_l can be arbitrarily small. Notice that $(x_l, y_l) \notin \partial T_l$ thus if $(x_l, y_l) \in S_l$ and $R(S_l) < inf \sqrt{(x_l - x)^2 + (y_l - y)^2}, (x, y) \in \partial T_l$, then $S_l \subseteq T_l$. $R(S_l)$ is the radius of the minimum disc which can cover S_l. And

$$\begin{aligned} &Pr((x_l, y_l) \in S_l) = Pr((x_l, y_l) \in \cap_{j \in adj(l)} \{(x, y) | \sqrt{(x - x_j)^2 + (y - y_j)^2} \in (d_{l,j} - D, d_{l,j} + D)\} \\ &\geq 1 - |adj(l)| Pr(\sqrt{(x_l - x_j)^2 + (y_l - y_j)^2} \notin (d_{l,j} - D, d_{l,j} + D)) \geq 1 - \epsilon \end{aligned} \tag{2.36}$$

□

Thus the estimation variance behaves asymptotically like the estimation variance where connectivity is considered independent with distance. In that case, the lower estimation bound still have the same form as the Cramer-Rao lower bound.

2.2.1.4 Simulation results

In this section, we calculate the Cramer-Rao bound for anchored localization problem as in [1]. We will focus on the lower bounds on the Cramer-Rao lower bound discussed in the previous section and show some simulation results.

In our simulations, we randomly generate M points on the plane presumably with unknown position, and N points with known position. The position of the points are uniformly distributed inside region A. We calculate the normalized Cramer-Rao lower bound on the estimation variance $V_x(i), V_y(i)$ and the lower bound derived in the previous section $L_x(i), L_y(i)$ for sensor $i = 1, 2, ..., M$. The visible radius is R. In our simulation, we fix A and vary N,M and R.

(i)In the first simulation,A is the region inside unit circle $x^2 + y^2 = 1$, we suppose for all sensor pair i, j, the distance between i, j is measured, i.e. $|adj(i)| = M + N - 1$. In the following figure, we show the the normalized Cramer-Rao bound $V_x(i) + V_y(i)$, the lower bound of CRLB $L_x(i) + L_y(i)$ and $\frac{4}{|adj(i)|}$. In Fig2.5, N=10, M=100.

(ii) In the following simulations, we show the effect of the visible radius. In Fig2.6, there are 3 setups Fix the number of sensors N=10,M=100,(a) R=0.5 (b) R=0.7 and (c) R=0.9. As can be seen, the bounds decreases as the visible radius increases.

From the simulations, we can see that $\frac{4}{|adj(i)|}$ is a tight lower bound on $L_x(i) + L_y(i)$, and $L_x(i) + L_y(i)$ is a reasonable lower bound on $V_x(i) + V_y(i)$ when the average number of sensors inside the circle of radius R is reasonably big.

From corollary 2.3 and above simulation results, we can tell that the position estimation variance of (x_l, y_l) is lower bounded by the nature of local geometry $\{(x_j, y_j), j \in$

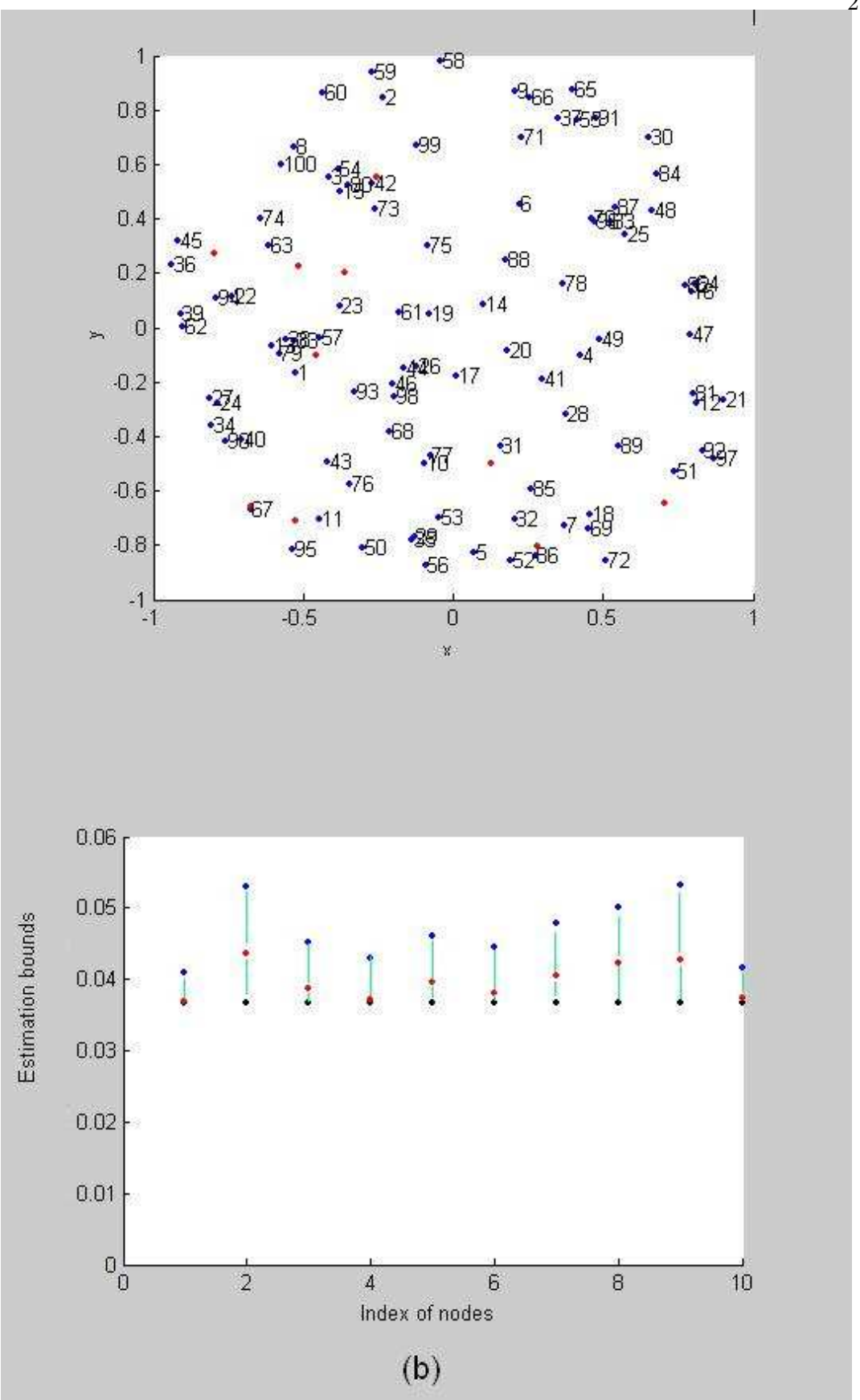


Figure 2.5: Bounds on anchored localization 1 N=10,M=100;
(a) Blue dots: unknown position, Red dots: known position (b)blue dots: $V_x(i) + V_y(i)$, red dots: $L_x(i) + L_y(i)$, black dots: $\frac{4}{|adj(i)|}$

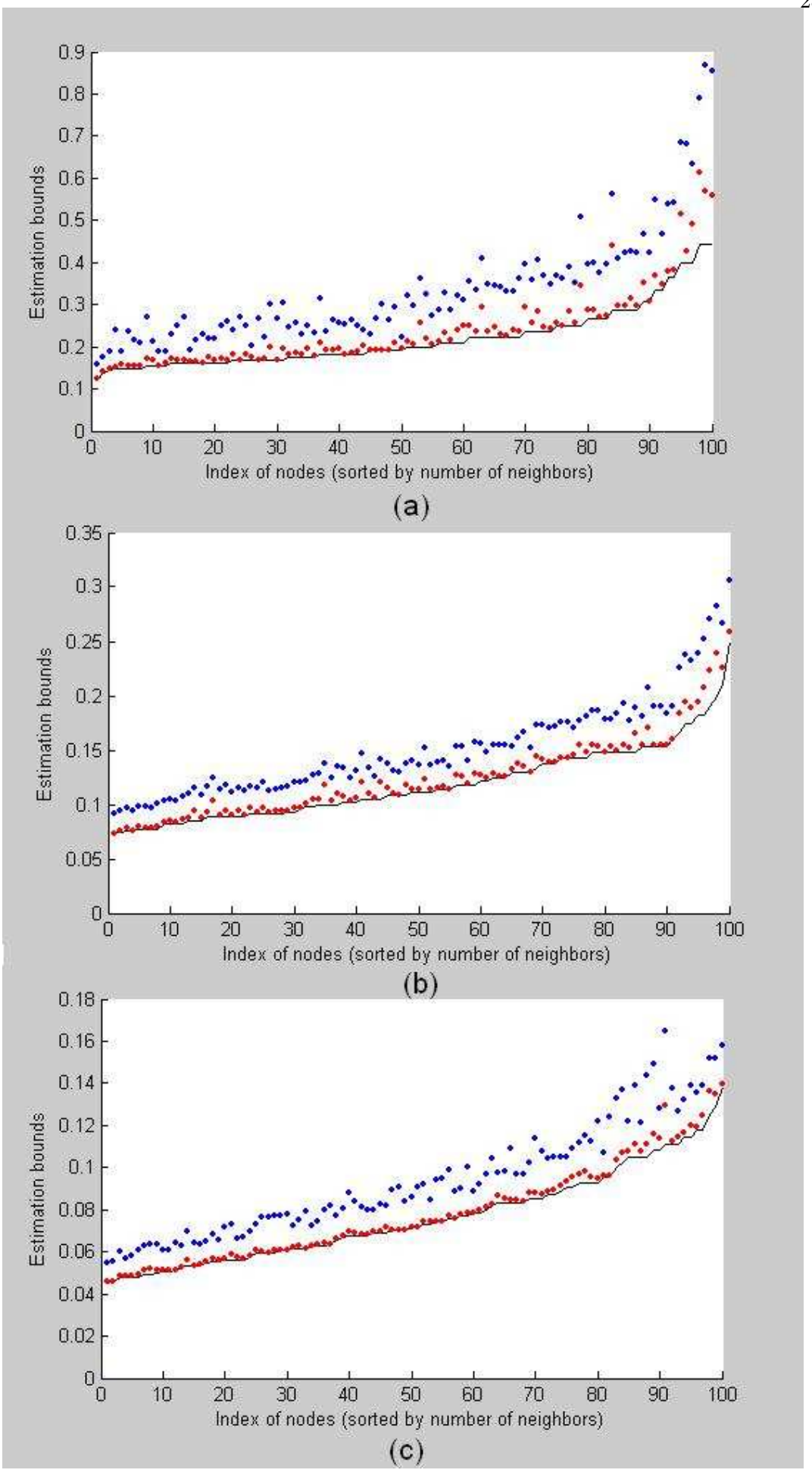


Figure 2.6: blue dots: $V_x(i) + V_y(i)$, red dots: $L_x(i) + L_y(i)$, black curve: $\frac{4}{|adj(i)|}$
(a) N=10,M=100, R=0.5 (b) N=10,M=100, R=0.7 (c) N=10,M=100, R=0.9

$adj(l)\}$and the estimation variance $\sigma^2_{j,l}$of the distance measures $d_{j,l}$. i. e. increasing the distance measurement accuracy in a faraway part of the sensor network, does not help to increase the accuracy of a position estimation.

2.2.2 The Cramer-Rao bound on Anchor-free Localization

2.2.2.1 Singularity of the Fisher Information Matrix

The standard Cramer-Rao lower bound does not exist for an estimation problems with singular Fisher Information matrix[30]. We are going to tackle a class of estimation problem with singular Fisher Information matrices. First we have the following observations on the Fisher Information matrix.

Lemma 2.1. *Rank of the Fisher information matrix: One sufficient and necessary condition for J to be nonsingular at θ is that the expectation of the square of directional derivative of l at θ is non zero for any direction $b \in R^n$.*

proof : the directional derivative of $l(x|\theta)$ at θ, along direction b is : $\tau(b) = (\partial l/\partial\theta_1, \partial l/\partial\theta_2, ..., \partial l/\partial\theta_n)b$.

$$\begin{aligned} E(\tau(b)^2) &= E(b^T(\partial l/\partial\theta_1, \partial l/\partial\theta_2, ..., \partial l/\partial\theta_n)^T(\partial l/\partial\theta_1, \partial l/\partial\theta_2, ..., \partial l/\partial\theta_n)b) \\ &= b^T Jb \end{aligned} \tag{2.37}$$

A symmetric matrix J is non-singular iff for any non-zero vector $b \in R^n$, $b^T Jb > 0$. By Eqn.2.37, we know that J is non-singular iff $E(\tau(b)^2)$ is non-zero for any $b \in R^n$. □

If there are no sensors have known positions, this could happen if we only care about the anchor-free coordinate system of a sensor network. The problem is now: there are M sensors $T_i, i = 1, ...M$ have unknown 2-D position (u_i, v_i), distance measures between sensors are $d_{T_i,T_j}, 1 \leq i < j \leq M$. Then the localization problem can *not* be treated as a parameter estimation problem for $\theta = (u_1, u_2, ...u_M, v_1, v_2, ..., v_M)$. Because the Fisher information matrix J is singular, thus the parameter estimation problem is not well defined. Here we give a brief proof of the singularity of the Fisher information matrix $J(\theta)$, let $h = (d_{T_i,T_j}), 1 \leq i < j \leq M$ be the observation vector:

Theorem 2.5. *For the anchor-free localization problem, the Fisher Information Matrix $J(\theta)$ is singular*

Proof. : The parameter vector $\theta = (u_1, u_2, ...u_M, v_1, v_2, ..., v_M)$, where (u_i, v_i) is the position for sensor T_i. The observation vector $h = (d_{T_i,T_j}), 1 \leq i < j \leq M$.Then the log-likelihood function of this estimation problem is :

$$\begin{aligned} l(h|\theta) &= constant + \frac{-1}{2\sigma^2}\sum(\sqrt{(u_i - u_j)^2 + (v_i - v_j)^2} - d_{T_i,T_j})^2 \\ &= constant + \frac{-1}{2\sigma^2}\sum(\sqrt{((u_i + ac) - (u_j + bc))^2 + ((v_i + ac) - (v_j + bc))^2} - d_{T_i,T_j})^2 \end{aligned} \tag{2.38}$$

$\forall a, b, c \in R$. Now given any parameter θ, fix a, b, the value of the log-likelihood function l along the direction $\vec{x} = (a, a, ...a, b, b, ..b)$ is constant, thus the directional derivative of l along vector $\theta + c\vec{x}, c \in R$ is zero. From Lemma 2.1, we know that the Fisher information matrix J is singular. □

2.2.2.2 equivalence class Estimation

Now we give the definition of equivalence class estimation in R^n. First we give the definition of equivalence class of parameters for a class of pdf functions.

Definition 2.1. ***Equivalent*** $\sim$***:*** *Let $y \in R^N, N = 1, 2, ...$ denote the available data vector. Assume that the data has been generated by some stochastic system with parameter vector $\theta \in R^M$ and that the density function of y given θ is $f(y|\theta)$. Two parameter vectors $\alpha, \beta \in R^M$ are called equivalent parameters iff $\forall y \in R^N, N = 1, 2, ...,$ $f(y|\alpha) = f(y|\beta)$, denote as $\alpha \sim \beta$. Obviously $\sim$ is an equivalent relation in R^M.*

Definition 2.2. ***Equivalent parameter class:*** *The equivalent parameter class for a parameter vector $\alpha \in R^N$ is defined as $\widetilde{\alpha} = \{\beta \in R^M : \alpha \sim \beta\}$. Obviously the equivalent parameter class is an equivalence class. From the property of equivalence classes, R^M is the union of some disjointed equivalent parameter classes.*

Definition 2.3. ***Parallel classes:*** *Two equivalence classes $\widetilde{\alpha}$ and $\widetilde{\beta}$ are called parallel, iff $\forall \alpha_1, \alpha_2 \in \widetilde{\alpha}$, $d(\alpha_1, \widetilde{\beta}) = d(\alpha_2, \widetilde{\beta})$.*

Where $d(\alpha, T) = min_{x \in T} d(\alpha, x), \alpha \in R^M, T \subseteq R^M$, *is the conventional definition of the distance from a point to a set,* d *is the usual Euclidean metric.*

We define $d(\widetilde{\alpha}, \widetilde{\beta}) = d(\alpha_1, \widetilde{\beta}), \alpha_1 \in \widetilde{\alpha}$ to be the distances between 2 equivalence classes $\widetilde{\alpha}, \widetilde{\beta}$ if they are parallel. If all the equivalence classes are parallel, then the underlining relation $\sim$ is called well-paralleled relation. And the distances is indeed a metric.

Lemma 2.2. $d(\widetilde{\alpha}, \widetilde{\beta})$ *is a metric in the equivalence classes space.*

proof : i) $d(\widetilde{\alpha}, \widetilde{\beta}) = d(\alpha, \widetilde{\beta}) \geq 0, \alpha \in \widetilde{\alpha}$. ii) $d(\widetilde{\alpha}, \widetilde{\beta})$ along direction b is : $\tau(b) = (\partial l/\partial\theta_1, \partial l/\partial\theta_2, ..., \partial l/\partial\theta_n)b$. □

And we have the following definition for **equivalence class estimation** and the metric for equivalence classes in an equivalence class estimation problem.

Definition 2.4. ***Equivalence class estimation:*** *Let* $y \in R^N, N = 1, 2, ...$ *denote the available data vector. Assume that the data has been generated by some stochastic system with equivalent parameter vector classes,* $\widetilde{\theta} \subseteq R^M$ *and that the density function of* y *given* $\forall \alpha \in \widetilde{\theta}$ *is* $f(y|\theta)$. *The underline relation* $\sim$ *is defined in Def.2.2. If all pairs of equivalence classes* $\widetilde{\alpha}, \widetilde{\beta}$ *are parallel as defined in Def.2.3. Then the estimation problem of* $\widetilde{\theta}$ *based on the available data vectors* y *is an equivalence class estimation problem.*

Definition 2.5. M/K ***Equivalence class estimation:*** *In an equivalence class estimation problem, if all but finite many equivalence classes* $\widetilde{\theta}_1, ..., \widetilde{\theta}_L, L < \infty$, *are* M *dimensional manifolds in* R^M. *And without loss of generality, we assume for all but finitely many equivalence classes* $\widetilde{\theta}$, $\forall \theta \in \widetilde{\theta}, \forall$ *equivalence class* $\widetilde{\beta}$, *there exists a unique* $\beta \in \widetilde{\beta}$, *s.t.* $d(\theta, \beta) = d(\widetilde{\theta}, \widetilde{\beta})$.

A simple M/K equivalence class estimation problem is given in Appendix C in which $M = 2$, $K = 1$.

In an equivalence class estimation problem, we define the **estimation variance** as : $E[d(\widetilde{\theta},\widetilde{\widehat{\theta}})^2]$, where $\widetilde{\widehat{\theta}}$ is the estimate of the equivalence class. By the definition of the distance d on the equivalence classes, we know that $\forall\theta\in\widetilde{\theta},\forall$ equivalence class $\widetilde{\beta}$ $, d(\theta,\widetilde{\beta}) = d(\widetilde{\theta},\widetilde{\beta})$. I.E. $\forall\theta\in\widetilde{\theta},\forall$ equivalence class $\widetilde{\beta}$ $,\exists\beta\in\widetilde{\beta}\subseteq R^M d(\theta,\beta) = d(\widetilde{\theta},\widetilde{\beta})$.

In an M/K estimation problem, for all but finite equivalence class $\widetilde{\theta}$ behaves like a K dimensional manifold embedded into R^M. So we can define the tangent space at $\theta\in\widetilde{\theta}$, and the normal space at θ. Denote $T(\theta)$ to be the K dimensional tangent space at θ of the manifold, generated by orthonormal vectors $\vec{v_1},...\vec{v_K}$, where each $\vec{v_i}$ is a unit vector in R^M. Denote $S(\theta)$ to be the $M-K$ dimensional normal space at θ which is orthogonal to $T(\theta)$. $S(\theta)$ is generated by $M-K$ ortohnormal vectors $\vec{w_1},...\vec{w_{M-K}}$.

Lemma 2.3. ***singular directions***

(1)$\forall x\in R^N$*,the directional derivative of* $l(x|\theta)$ *along any direction* $\vec{v}\in T(\theta)$ *is* 0*. Where* $l(x|\theta) = log(p(x|\theta))$*.*

$$(\frac{\partial l(x|\theta)}{\partial\theta_1},\frac{\partial l(x|\theta)}{\partial\theta_2},...,\frac{\partial l(x|\theta)}{\partial\theta_M})\vec{v} = 0 \tag{2.39}$$

(2) $\forall\vec{w}\in S(\theta)$*,* $\exists x\in R^N$*, the directional derivative of* $l(x|\theta)$ *along* $\vec{w}$ *is nonzero.*

$$(\frac{\partial l(x|\theta)}{\partial\theta_1},\frac{\partial l(x|\theta)}{\partial\theta_2},...,\frac{\partial l(x|\theta)}{\partial\theta_M})\vec{w} \neq 0 \tag{2.40}$$

proof: the correctness follows directly from the definition of $T(\theta)$. □

Lemma 2.4. ***Solution Space*** *For an* M/K *equivalence class estimation problem, if* $\widetilde{\theta}$ *is an equivalence class, then* $\forall\theta\in\widetilde{\theta}$*, we denote* $U(\theta) = \{\beta\in R^M | d(\theta,\widetilde{\beta}) = d(\theta,\beta)\}$*, where* $\widetilde{\beta}$ *is the equivalence class* β *belongs to.*
$U(\theta)\subseteq S(\theta)$*.*

proof: For any $\beta\in\widetilde{\beta}$, if $d(\beta,\theta) = d(\widetilde{\beta},\widetilde{\theta})$, then $\theta = argmin_{\theta\in\widetilde{\theta}} d(\theta,\beta)$. At the local region around θ, a point ρ on the K dimensional manifold can be described as $\rho = \theta + \sum_{i=1}^{K} a_i v_i$, where $a_i\in R$. Meanwhile $\beta = \theta + \sum_{i=1}^{K} b_i v_i + \sum_{j=1}^{M-K} c_j w_j$. Now

$d(\beta,\rho)=\sum_{i=1}^{K}(a_i-b_i)^2+\sum_{i=1}^{M-K}c_j^2$. It takes the minimum value at $a_i=0, i=1,..K$, it is true only if $b_i=0, i=1,..K$. □.

Now at a point $\theta\in\widetilde{\theta}$, we can do the *parameter* estimation around that point on the normal subspace $S(\theta)$. In order to calculate the lower bound of the estimation variance for the equivalence class estimation problem. We need to find the connections between the equivalence class estimation and parameter estimation on the normal bundle $S(\theta)$.

Definition 2.6. ***Unbiased equivalence class estimation*** *In an M/K equivalence class estimation problem. An estimator is called unbiased equivalence class estimator, if and only if the following conditions hold. For all but finitely many $\widetilde{\theta}$, if the estimate is $\hat{\widetilde{\theta}}$, $\forall\beta\in\widetilde{\theta}$, let $\hat{\theta}_\beta$ be the point $\in\hat{\widetilde{\theta}}$ and $d(\theta,\hat{\theta}_\beta)=d(\widetilde{\theta},\hat{\widetilde{\theta}})$. If $E(\hat{\theta}_\beta)=\theta$, then we call the estimator an unbiased equivalence class estimator.*

If we look at the parameter estimation problem around point θ on the normal subspace $S(\theta)$, $S(\theta)$ is orthogonal to the tangent subspace on θ of the K dimensional equivalence class $\widetilde{\theta}$, thus has dimension $M-K$,for any point y on $S(\theta)$, there is a unique $M-K$ dimensional vector $\beta=(\beta_1,...\beta_{M-K})$,s.t. $y-\theta=\beta_1w_1+\beta_2w_2+...+\beta_{M-K}w_{M-K}$,. Then we let $f(x|\beta)$ be a family of pdfs parameterized by β on $S(\theta)$, an $M-K$ dimensional subspace. Let $l=log(f)$ be the log-likelihood function, and Fisher information matrix $J(\theta)$, $J(\theta)_{ij}=E[(\partial l/\partial\beta_i)(\partial l/\partial\beta_j)]$.

Lemma 2.5. ***Recitation of Cramer-Rao bound on*** $S(\theta)$

For the estimation problem on $S(\theta)$, $\forall$ unbiased estimator $\hat{\theta}$ of θ,

$$C(\theta)\geq J(\theta)^{-1} \tag{2.41}$$

Where $C(\theta)=E[(\beta-\hat{\beta})(\beta-\hat{\beta})^T]$ is the covariance matrix of $\hat{\beta}-\beta$ and $J(\theta)$ is the Fisher information matrix with respect to these coordinates $w_1,..w_{M-K}$.

proof: This is a recitation of Theorem 2.1 on subspace $S(\theta)$. □

Theorem 2.6. ***Estimation variance of the*** M/K ***equivalence class estimation:*** *For an equivalence class estimation problem, with data* x*, parameter* $\widetilde{\theta}$*, for any non-biased equivalence class estimator* $\hat{\widetilde{\theta}}$*.* $\forall\theta \in \widetilde{\theta}$*,let* $J(\theta)$ *to be the Fisher information matrix defined in Lemma 2.5. Then* $E(d(\widetilde{\theta}, \hat{\widetilde{\theta}})^2) \geq Tr(J(\theta)^{-1}))$*. Where* $Tr(A)$ *is the sum of all diagonal entries of matrix* A*.*

proof: Proof by contradiction. Suppose $\exists$ unbiased estimator $T(x) = \hat{\widetilde{\theta}}$, for some $\widetilde{\theta} \subseteq R^M, \theta \in \widetilde{\theta}$, such that the estimation variance $E(d(\widetilde{\theta}, \hat{\widetilde{\theta}})^2) \leq Tr(J(\theta)^{-1}))$. Now look at the parameter estimation problem of θ on $S(\theta)$ given data x. We can always $lift$ the estimation problem to the equivalence class estimation problem without introducing any new data, noticing the structure of the equivalence classes is well defined. Then we can estimate the equivalence class $\widetilde{\theta}$, the estimation is $\hat{\widetilde{\theta}}$, then project $\hat{\widetilde{\theta}}$ back to $S(\theta)$, and let the intersection of $\hat{\widetilde{\theta}}$ and $S(\theta)$, $\hat{\theta}$to be the parameter estimation of θ on $S(\theta)$. Then the estimation variance is still lower bounded by the Cramer-Rao bound in Lemma 2.5. So $E((\theta - \hat{\theta})^T(\theta - \hat{\theta})) \geq Tr(J(\theta)^{-1})$. Meanwhile, $(\theta - \hat{\theta})^T(\theta - \hat{\theta}) = d(\widetilde{\theta}, \hat{\widetilde{\theta}})$, so we have $E(d(\widetilde{\theta}, \hat{\widetilde{\theta}})^2) \geq Tr(J(\theta)^{-1})$. Thus we have the contradiction. □

To compute $J(\theta)$, we need to find the subspace $S(\theta)$ first. However it turns out that we can compute $Tr(J(\theta)^{-1})$ without directly computing $J(\theta)$. Now we give the computation tool for the lower estimation bound for equivalence class estimation problem.

Corollary 2.4. *Computation of the lower bound of* $E(d(\widetilde{\theta}, \hat{\widetilde{\theta}})^2)$*,* $Tr(J(\theta)^{-1})$

$$Let \qquad H(\theta) = E[(\frac{\partial l}{\partial \theta_1}, \frac{\partial l}{\partial \theta_2}, ..., \frac{\partial l}{\partial \theta_M})^T(\frac{\partial l}{\partial \theta_1}, \frac{\partial l}{\partial \theta_2}, ..., \frac{\partial l}{\partial \theta_M})] \tag{2.42}$$

$Rank(H(\theta)) = M - K$*, let* $\lambda_1, \lambda_2, ..., \lambda_{M-K}$ *be the nonzero eigenvalues of* $H(\theta)$*, then*

$$Tr(J(\theta)^{-1}) = \sum_{i=1}^{M-K} \frac{1}{\lambda_i} \tag{2.43}$$

proof: J is defined in Lemma2.5.

$$J(\theta)_{ij} = E[\frac{\partial l}{\partial \beta_i}\frac{\partial l}{\partial \beta_j}] \tag{2.44}$$

Where $l = log(x|\beta)$ be a family of pdfs parameterized by β on $S(\theta)$. And $\partial l/\partial\beta_i$ is indeed the directional derivative in the original R^M along the direction $\vec{w_i}$, where $\vec{w_i} \in R^M, i = 1, 2, ..M-K$ forms an orthonormal bases for $S(\theta)$. Suppose the original R^M is parameterized by orthonormal vectors $\vec{e_1}, \vec{e_2}, ...\vec{e_M}$, where $\vec{e_i} = (\delta_{1i}, \delta_{2i}, ..., \delta_{Mi})^T$, $\delta_{ji} = 1, j = i, \delta_{ji} = 0, j \neq i$.And $\theta \in R^M$ can be written as $\theta = (\theta_1, \theta_2, ...\theta_M)^T$. Let

$$W = \begin{pmatrix} \vec{w_1} & \vec{w_2} & \dots & \vec{w_{M-K}} \end{pmatrix} = \begin{pmatrix} \vec{e_1} & \vec{e_2} & \dots & \vec{e_M} \end{pmatrix} W \tag{2.45}$$

then $W^T W = I_{M-K}$. And we have

$$\frac{\partial l}{\partial \beta_i} = (\frac{\partial l}{\partial \theta_1}, \frac{\partial l}{\partial \theta_2}, ..., \frac{\partial l}{\partial \theta_M})\vec{w_i}$$

Substitute the above equation into Eqn.2.44, we have:

$$\begin{aligned} J(\theta)_{ij} &= E[\frac{\partial l}{\partial \beta_i}\frac{\partial l}{\partial \beta_j}] = E[\vec{w_i}^T(\frac{\partial l}{\partial \theta_1}, \frac{\partial l}{\partial \theta_2}, ..., \frac{\partial l}{\partial \theta_M})^T(\frac{\partial l}{\partial \theta_1}, \frac{\partial l}{\partial \theta_2}, ..., \frac{\partial l}{\partial \theta_M})\vec{w_j}] \\ &= \vec{w_i}^T E[(\frac{\partial l}{\partial \theta_1}, \frac{\partial l}{\partial \theta_2}, ..., \frac{\partial l}{\partial \theta_M})^T(\frac{\partial l}{\partial \theta_1}, \frac{\partial l}{\partial \theta_2}, ..., \frac{\partial l}{\partial \theta_M})]\vec{w_j} \end{aligned} \tag{2.46}$$

From the definition of $H(\theta)$ we have:

$$J(\theta) = W^T H(\theta) W \tag{2.47}$$

From Lemma.2.3, we know that for any direction lying in the tangent space $T(\theta)$ of the K dimensional equivalence class manifold,$\forall x \in R^N$ the directional derivative of $l(x|\theta)$is 0, i.e. $\forall \vec{v} \in T(\theta)$

$$(\frac{\partial l(x|\theta)}{\partial \theta_1}, \frac{\partial l(x|\theta)}{\partial \theta_2}, ..., \frac{\partial l(x|\theta)}{\partial \theta_M})\vec{v} = 0 \tag{2.48}$$

So, for the orthonormal vectors $\vec{v_1}, ..., \vec{v_K}$ which generated $T(\theta)$, we have

$$\begin{aligned} \vec{v_i}^T H(\theta)\vec{v_i} &= \vec{v_i}^T E[(\frac{\partial l}{\partial \theta_1}, \frac{\partial l}{\partial \theta_2}, ..., \frac{\partial l}{\partial \theta_M})^T(\frac{\partial l}{\partial \theta_1}, \frac{\partial l}{\partial \theta_2}, ..., \frac{\partial l}{\partial \theta_M})]\vec{v_i} \\ &= E[\vec{v_i}^T(\frac{\partial l}{\partial \theta_1}, \frac{\partial l}{\partial \theta_2}, ..., \frac{\partial l}{\partial \theta_M})^T(\frac{\partial l}{\partial \theta_1}, \frac{\partial l}{\partial \theta_2}, ..., \frac{\partial l}{\partial \theta_M})\vec{v_i}] = 0 \end{aligned} \tag{2.49}$$

Meanwhile, with the assumption that $p(x|\theta)$ is non-zero, following Lemma 2.3,for the orthonormal vectors $\vec{w_1}, ..., \vec{w_{M-K}}$ which generate $S(\theta)$,we have

$$\begin{aligned} \vec{w_i}^T H(\theta)\vec{w_i} &= \vec{w_i}^T E[(\frac{\partial l}{\partial \theta_1}, \frac{\partial l}{\partial \theta_2}, ..., \frac{\partial l}{\partial \theta_M})^T (\frac{\partial l}{\partial \theta_1}, \frac{\partial l}{\partial \theta_2}, ..., \frac{\partial l}{\partial \theta_M})]\vec{w_i} \\ &= E[\vec{w_i}^T (\frac{\partial l}{\partial \theta_1}, \frac{\partial l}{\partial \theta_2}, ..., \frac{\partial l}{\partial \theta_M})^T (\frac{\partial l}{\partial \theta_1}, \frac{\partial l}{\partial \theta_2}, ..., \frac{\partial l}{\partial \theta_M})\vec{w_i}] > 0 \end{aligned} \quad (2.50)$$

The above shows that $Rank(H(\theta)) = M - K$. Notice that $v_1, ..v_K, w_1, ...w_{M-K}$ form an orthonormal space for R^M, we know that v_is are eigenvectors of $H(\theta)$ with 0 eigenvalues, meanwhile those eigenvectors of $H(\theta)$ corresponding to non-zero eigenvalue span the same subspace as $w_1, ..., w_{M-K}$, i.e., denote the non-zero eigenvectors to be $\vec{\gamma_1}, \vec{\gamma_2}, ..., \vec{\gamma_{M-K}}$ corresponding to eigenvalues $\lambda_1, ..., \lambda_{M-K}$ respectively, then $\exists A$, an $(M-K) \times (M-K)$ unitary matrix, s.t. $A^T A = I_{M-K}$, and

$$W = (\vec{w_1}, ..., \vec{w_{M-K}}) = (\vec{\gamma_1}, \vec{\gamma_2}, ..., \vec{\gamma_{M-K}})A = \Gamma A \quad (2.51)$$

So

$$\begin{aligned} J(\theta) &= W^T H(\theta) W = A^T \Gamma^T H(\theta) \Gamma A = A^T diag(\lambda_1, \lambda_2, ..., \lambda_{M-K}) A \\ J(\theta)^{-1} &= A^T diag(\lambda_1^{-1}, \lambda_2^{-1}, ..., \lambda_{M-K}^{-1}) A \end{aligned} \quad (2.52)$$

Noticing that $Tr(AB) = Tr(BA)$, we finally have:

$$\begin{aligned} Tr(J(\theta)^{-1}) &= Tr(A^T diag(\lambda_1^{-1}, \lambda_2^{-1}, ..., \lambda_{M-K}^{-1}) A) \\ &= Tr(AA^T diag(\lambda_1^{-1}, \lambda_2^{-1}, ..., \lambda_{M-K}^{-1})) = \sum_{i=1}^{M-K} \frac{1}{\lambda_i} \end{aligned} \quad (2.53)$$

□

2.2.2.3 Equivalence Class Estimation of the Anchor-free Localization Problem

In the first section, we showed that if no sensors have known position, the localization problem cannot be described as a parameter estimation problem. However, it can be well defined in the framework of the equivalence class estimation problem.

(1) Equivalence classes

For an M points localization problem, the data $z \in R^N$ are modelled as the inter-sensor distances corrupted by iid Gaussian noises $\sim N(0, \sigma^2)$. i.e. $z_i = \sqrt{(x_{i1} - x_{i2})^2 + (y_{i1} - y_{i2})^2} + \delta_i$. The equivalence class for $\theta = (x_1, y_1, x_2, y_2, ...x_N, y_N) \in R^{2N}$ is $\widetilde{\theta} = \{(u_1, v_1, u_2, v_2, ...u_N, v_N) \in R^{2N} | (u_i, v_i)^T = R(\alpha)(x_i, y_i)^T + (t_x, t_y)^T\} \subseteq R^{2N}$, where

$$R(\alpha) = \begin{pmatrix} cos(\alpha) & -sin(\alpha) \\ sin(\alpha) & cos(\alpha) \end{pmatrix} \tag{2.54}$$

is the rotation matrix, and $(t_x, t_y)^T \in R^2$ is the translation vector. Here we ignore the fact that the reflection of any of the points in the following way falls into the same equivalence class, $\beta = (u_1, v_1, u_2, v_2, ...u_N, v_N) \in \widetilde{\theta}$ let $\beta^* = (u_1, -v_1, u_2, -v_2, ...u_N, -v_N)$, then $\forall z \in R^N, p(z|\beta) = p(z|\beta^*)$. In general $\widetilde{\theta} \bigcap \widetilde{\theta^*} = \emptyset$, unless all the points are collinear.

We notice that there is no subspace in which we can do parameter estimation. This claim is proved by a simple counter-example in Appendix D. i.e. the equivalence class estimation model is the only proper way to describe the problem.

(2) Calculation of $H(\theta)$.

Now at $\forall \theta \in R^{2M}$, we calculate $H(\theta)$, and a lower bound on the variance of the equivalence class estimation is $\sum_{i=1}^{2M-3} \frac{1}{\lambda_i}$, where $\lambda_i, i = 1, 2, ..., 2M-3$ are the non-zero eigenvalues of $H(\theta)$.

For point i, denote $c(i) = \{j | d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} + \delta_{ij}$ is measured $\}$. Then, $\forall j \in c(i)$

$$H(\theta)_{2i-1,2j-1} = E[\frac{\partial l}{\partial x_i}\frac{\partial l}{\partial x_j}] = -\frac{1}{\sigma^2}cos(\alpha_{ij})^2 \tag{2.55}$$

$$H(\theta)_{2i,2j} = E[\frac{\partial l}{\partial y_i}\frac{\partial l}{\partial y_j}] = -\frac{1}{\sigma^2}sin(\alpha_{ij})^2 \tag{2.56}$$

$$H(\theta)_{2i-1,2j} = E[\frac{\partial l}{\partial x_i}\frac{\partial l}{\partial y_j}] = -\frac{1}{\sigma^2}cos(\alpha_{ij})sin(\alpha_{ij}) \tag{2.57}$$

$$H(\theta)_{2i,2j-1} = E[\frac{\partial l}{\partial y_i}\frac{\partial l}{\partial x_j}] = -\frac{1}{\sigma^2}cos(\alpha_{ij})sin(\alpha_{ij}) \tag{2.58}$$

Where $\alpha_{ij} \in [0, 2\pi)$, and $cos(\alpha_{ij}) = \frac{x_i - x_j}{\sqrt{(x_i-x_j)^2+(y_i-y_j)^2}}$, $sin(\alpha_{ij}) = \frac{y_i - y_j}{\sqrt{(x_i-x_j)^2+(y_i-y_j)^2}}$.

And $\forall i = 1, 2, ...M$

$$H(\theta)_{2i-1,2i-1} = E[(\frac{\partial l}{\partial x_i})^2] = \frac{1}{\sigma^2} \sum_{j \in c(i)} cos(\alpha_{ij})^2 \tag{2.59}$$

$$H(\theta)_{2i,2i} = E[(\frac{\partial l}{\partial y_i})^2] = \frac{1}{\sigma^2} \sum_{j \in c(i)} sin(\alpha_{ij})^2 \tag{2.60}$$

$$H(\theta)_{2i,2i-1} = H(\theta)_{2i-1,2i} = E[\frac{\partial l}{\partial x_i}\frac{\partial l}{\partial y_i}] = \sum_{j \in c(i)} cos(\alpha_{ij})sin(\alpha_{ij}) \tag{2.61}$$

Theorem 2.7. *CR bound is invariant under zooming*

Proof. : $H(\theta) = H(a\theta), \forall a \in R$. This follows the expression of the matrix $H(\theta)$, thus the Cramer-Rao lower bound for the localization problem $V(\theta) = \sum_{i=1}^{2M-3} \frac{1}{\lambda_i}$, where λ_i's are non-zero eigenvalues of $H(\theta)$ is invariant under zooming $V(\theta) = V(a\theta)$. □

2.2.2.4 Simulation Results

In our simulation, we randomly generate M points on the plane in a region A and we calculate the normalized lower bound on the estimation variance $V(\theta) = \sigma^2 \sum_{i=1}^{2M-3} \frac{1}{\lambda_i}$, where λ_i's are non-zero eigenvalues of $H(\theta)$, it is worth to point out that $V(\theta)$ is invariant $\forall \theta \in \widetilde{\theta}$ by the definition of equivalence class estimation.

(i)In the first simulation, A is the region inside the unit circle $x^2 + y^2 = 1$, we suppose for all sensor pair i, j, the distance between i, j is measured. In the following figure, we show the average $V(\theta)$ over 50 independent experiments for different M. As can be seen, $V(\theta)$ remains constant as M increases. $V(\theta)$ indicates the total estimation variance of the positions of the sensors, thus the average estimation variance decreases as $\frac{1}{M}$.

(ii) In the following simulation, we show the effect of the shape of the region of the sensor networks on the normalized estimation variance $\sigma^2 V(\theta)$. We have M sensors uniformly distributed in a region A. Here we assume all the pairwise distances are measured. Our simulations are average of 50 independent experiments.

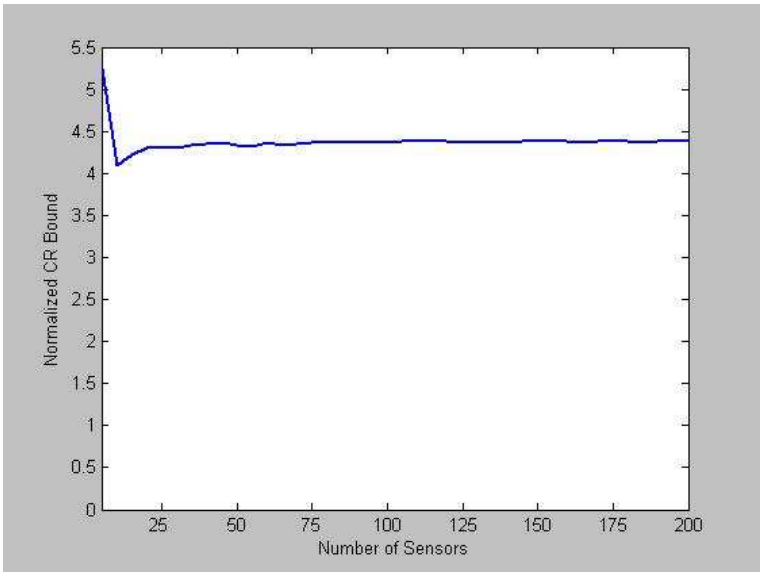


Figure 2.7: The normalized Cramer-Rao lower bound for localization VS number of nodes.
Nodes are uniformly distributed inside the unit circle.

In Fig.2.8 (a), A is a rectangular region with dimension $L_1 \times L_2, L_1 \geq L_2$, as we pointed above, the only thing matters is the ration $R = \frac{L_1}{L_2}$, and it turns out that the normalized CR bound increases as R increases. In Fig.2.8 (b), A is an annulus region with outer radius $R_1 = 1$, inner radius $R_2 = r$ and it turns out that the normalized CR bound increases as r decreases.

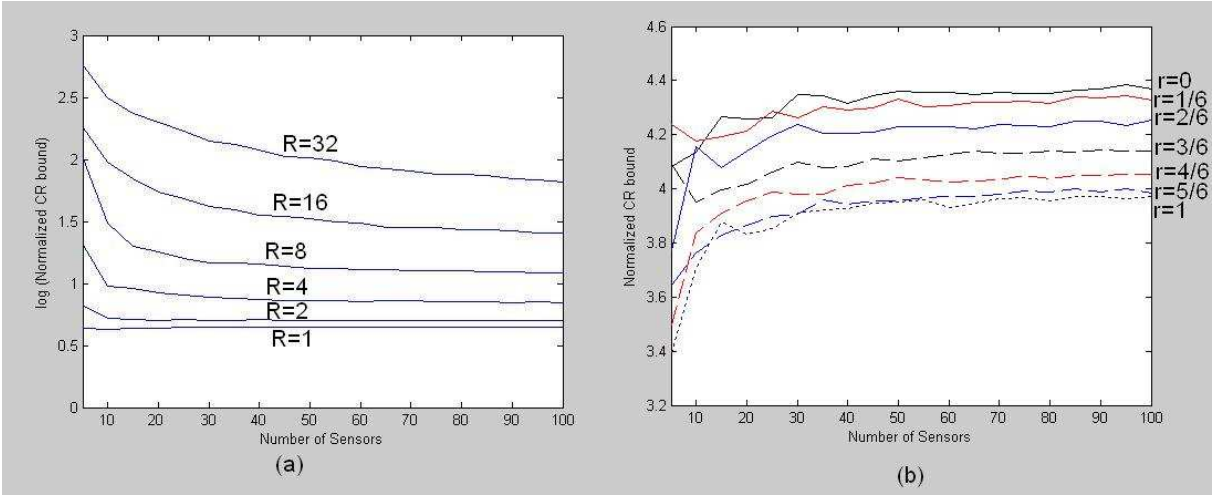


Figure 2.8: The normalized Cramer-Rao lower bound for localization VS number of nodes.
(a) Rectangular region ($R = \frac{L_1}{L_2}$)
(b)Annulus region ($R_1 = 1, R_2 = r$)

Before giving the results of some more simulations, we revisit the definition of visible radius, and revisit Corollary 2.4.

Definition 2.7. *Visible Radius R:*

In a sensor network, if for each sensor i, $\forall$ sensor j, d_{ij} is measured iff $d_{ij} < R$,R is the visible radius.

From Corollary 2.4, we know that, the total Cramer-Rao bound on estimation variance is the same for a sensor network with positions θ with visible radius R and a sensor network with positions $a\theta$ with visible radius aR. **The critical parameter is more likely to be the average number of visible sensors.**

(iii)In the next simulation, A is the region inside the unit circle $x^2 + y^2 = 1$. We have the number of sensors $M = 200, 150, 100$ respectively. In the following figure, we show an average $V(\theta)$ of 10 independent experiments for different visible radius R. As can be seen, the CR lower bound for equivalence class estimation only depends on R, i.e. the average fraction F of the sensors visible to each sensor, F is defined as the ratio of average visible sensors over the total number of sensors.

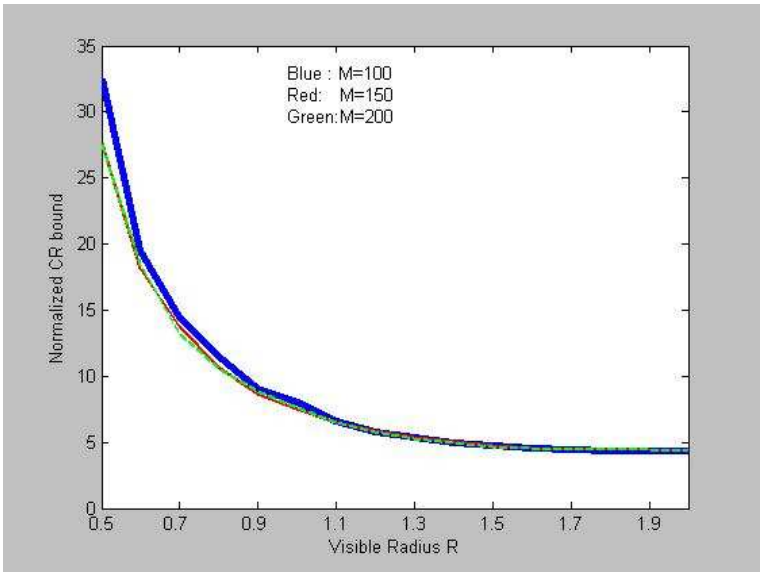


Figure 2.9: The normalized Cramer-Rao lower bound for localization VS visible radius R

Based on the observation in Fig.2.10, we conjecture that $\frac{\sigma^2 V(\theta)}{M} \sim \frac{1}{FM}$. i.e. the normalized estimation variance per sensor is inversely proportional to the average visible sensors for each sensor. Fig.2.10 illustrates the above conjecture.

(iv) The following simulation shows the relation between the equivalence class estimation variance and the size of the sensor network given constant sensor density and constant visible radius. For each sensor i, only those d_{ij} are measured iff $d_{ij} < R$, R is the visible radius. We fix $R = 2$, and the density of the sensor network is fixed as $\frac{20}{\pi}$ sensors per unit area, and increase the region of the sensor network. In our simulation, the region A is a circular region with radius $D = 1, 2, ..., 10$. In Fig.2.11, we show

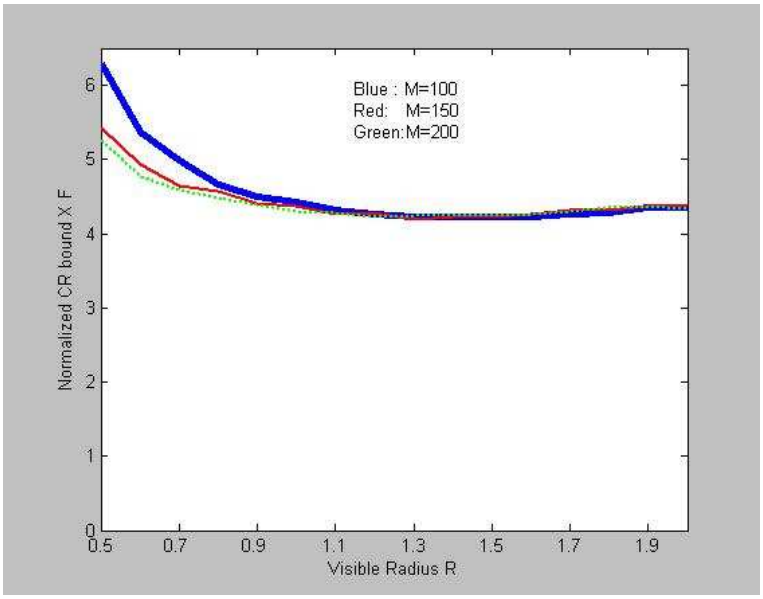


Figure 2.10: The normalized Cramer-Rao lower bound for localization $\times$ fraction F

an average estimation variance of 50 independent experiments for different radius D. Fig.2.11(a) is the total estimation variance bound, (b) is the average estimation variance bound.

As can be seen, the average estimation variance decreases as the size of network increases, but it converges to a constant. This supposes that it is good enough to estimate the position of the sensors in a nearby sensor network with radius $R^* \sim 3R$, when the average visible sensors are around 80.

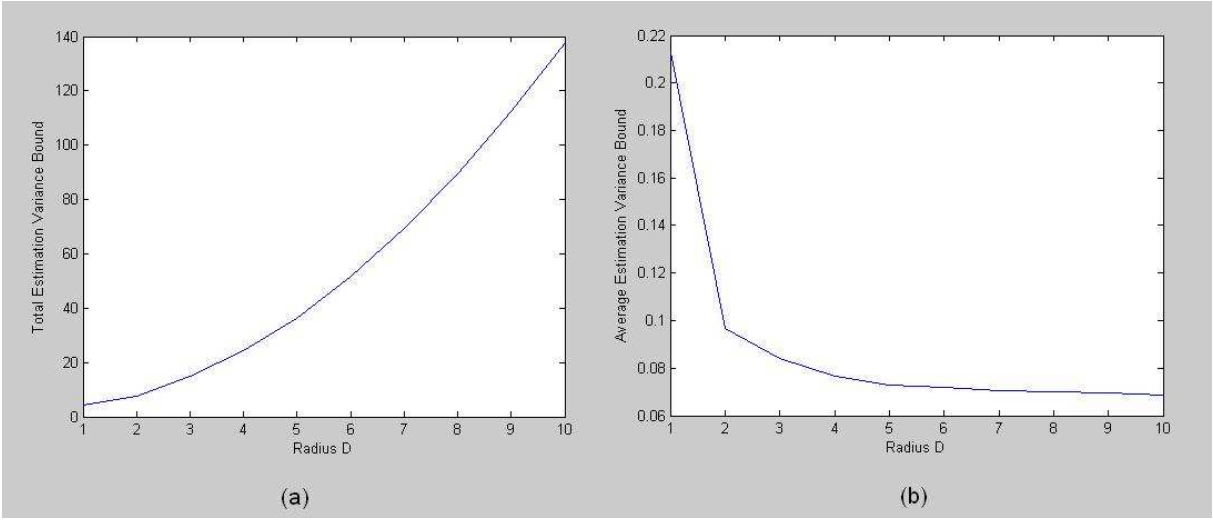


Figure 2.11: Average estimation variance VS size of sensor network, with fixed visible region and sensor density

2.3 2-Step Anchored Localization Scheme

In this section we propose a 2-step anchored localization scheme. We are dealing with anchored localization as stated in the previous section. F is the set of anchors. S is the set of nodes without known positions. The goal is to reconstruct the set

$$P_S = \{(x_i, y_i)^T | i \in S\} \tag{2.62}$$

Given

$$D = \{d_{i,j} | i \in S \cup F, j \in adj(i)\} \tag{2.63}$$

$$P_F = \{(x_i, y_i)^T | i \in F\} \tag{2.64}$$

Where $d_{i,j} = d'_{i,j} + \delta_{i,j}$, $\delta_{i,j}$'s are assumed to be iid Gaussian $\sim N(0, \sigma^2)$ and $d'_{i,j} = \sqrt{(x'_i - x'_j)^2 + (y'_i - y'_j)^2}$ is the true distance between sensor i and j. It is easy to give an optimization problem restatement of the problem based on the ML rule given the measurements of the distances $d_{i,j}$ with variance additive Gaussian noises. As following:

$$argmin_{\{(x_i,y_i)|i\in S\}} \sum_{i \in S \cup F} \sum_{j \in adj(i)} (\hat{d}_{i,j} - d_{i,j})^2 \tag{2.65}$$

$$\text{Where } \hat{d}_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{2.66}$$

Here we also assume that all those additive Gaussian noises added to the measurement of distances have the same variance. However, it is not easy to solve the above optimization problem in real time. Because the optimization problem here is not convex as shown in Appendix E, there could be many local optimal solutions to the problem. So we come up with an approximate solution to the problem without directly solving the optimization problem.

The first step of the scheme is to estimate an anchor-free coordinate system based solely on $D_S = \{d_{i,j} | i, j \in S, j \in adj(i)\}$. We do not distinguish any 2 coordinate systems such that one is a rigid transformed of another. In 2D, a rigid transformation is a combination of any rotations, translations and reflections.

Definition 2.8 (Anchor-free coordinate system). *An anchor-free coordinate system of a point set S is an equivalent class in $R^{2|S|}$: $P = \{(x_i, y_i)|i \in S\}$, we say P, Q are equivalent if and only if $(x_i, y_i)_p^T = A(x_i, y_i)_q^T + B, \forall i$, where $AA^T = I_2, B \in R^2$.*

The reason why they are equivalent is that the distances between any 2 points P_i, P_j are the same in P, Q. Thus based on the ML rule they have the same likelihood given the assumption that noises are gaussian random variable with same variances.

In the first stage, any position information of set F is not used. We estimate the anchor-free coordinate system solely based on the distances. In the second stage, we combine F with the anchor-free coordinate system and estimate the anchored coordinates of S. The solution is unique in general. And as can be seen later in Section 2.3.1 and Section 2.3.2, there is no iterations in the algorithm. We expect the algorithm to give an answer in real time. Our algorithm does not guarantee any optimality as the size of the space is huge and the problem is not convex.

2.3.1 Step 1 : Anchor-free Coordinates Estimation

Problem restatement :

Given a point set T, and the distances of some of the adjacent point pairs $D_T = \{d_{i,j}|i, j \in T, j \in adj(i)\}$, we want to assign a 2D coordinate $(x_i, y_i)$to each point i such that this coordinate system best matches the distance information we have in D_T. Again, a least square based estimation problem can be stated,

$$argmin_{\{(x_i,y_i)|i\in T\}} \sum_{i\in T} \sum_{j\in adj(i)} (\hat{d}_{i,j} - d_{i,j})^2 \tag{2.67}$$

$$\text{Where } \hat{d}_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{2.68}$$

The solution to this optimization problem is obviously non-unique, because the solution is an anchor-free coordinate systems which is an equivalent class in $R^{2|T|}$. Due

to the nature of the optimization problem (no-convexity as shown in Appendix E, and multiple local optimal), we will set up a more geometrical-oriented description of the problem. And a solution will be given.

We will propose a 2 step scheme. In the first stage, we will define a *cell*, and focus on the coordinates estimation in a single cell. In the second stage, we will glue all those small cells together to a complete anchor-free coordinate system.

2.3.1.1 Coordinates Estimation within a Cell

In this stage, we will first estimate the anchor-free coordinate system for a subset of $S \cup F$, with some properties called *cell*. The scheme here is based on ML estimation. In [4], they used a very similar scheme.

Definition 2.9 (cell). *A cell C is a subset of point set T, such that $\forall i, j \in C, i \in adj(j)$, or a clique in the graph if we treat the point as vertex, and vertex i, j are linked by an edge if $d_{i,j} \in D_T$. That is, we have D_T, such that $d_{i,j} \in D_T, \forall i, j \in C$. Now within a cell C, we want to minimize E.*

As can be seen, a cell is indeed a clique in the graph if two points (vertices) are linked by an edge if the distance is known.

$$E = argmin_{\{(x_i,y_i)|i\in C\}} \sum_{i,j\in C} (\hat{d}(i,j) - d_{i,j})^2 \tag{2.69}$$

$$\text{Where } \hat{d}_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{2.70}$$

First, let us make some obvious geometrical observations.

1) $|C|$=1, then all $(x_1, y_1) \in R^2$ are equivalent. Without loss of generality $p_1^T = (x_1, y_1) = (0, 0)$.

2) $|C| = 2$,then $D_T = \{d_{12}\}$, any 2 points in R^2 with distance d_{12} minimize E, without loss of generality, $p_1^T = (x_1, y_1) = (0, 0), p_2^T = (x_2, y_2) = (d_{12}, 0)$.

Now if we have $|C| > 2$, we can treat C as the union of 2 base points with $|C| - 2$ associate points. Let us fix the base and add other points one by one. Thus the accuracy of the distance between the 2 base points is crucial to get the accurate anchor-free coordinate system. Due to the additive noise to the distance measurement, we have to come back to get a more accurate distance estimate later.

Given another point P_3 with distance d_{32}, d_{31} to those 2 base points.

Case 1: the general case with triangule inequality satisfied, E achieves 0. And there are 2 solutions which are symmetrical around the x axis. The solutions are the 2 intersections of the 2 circles with origin at p_1, p_2, radius d_{31}, d_{32} respectively :

$$(d_{31}cos(\alpha), \pm d_{31}sin(\alpha)) \tag{2.71}$$

$$\text{Where } \alpha = cos^{-1}(\frac{d_{12}^2 + d_{31}^2 - d_{32}^2}{2d_{12}d_{31}}) \tag{2.72}$$

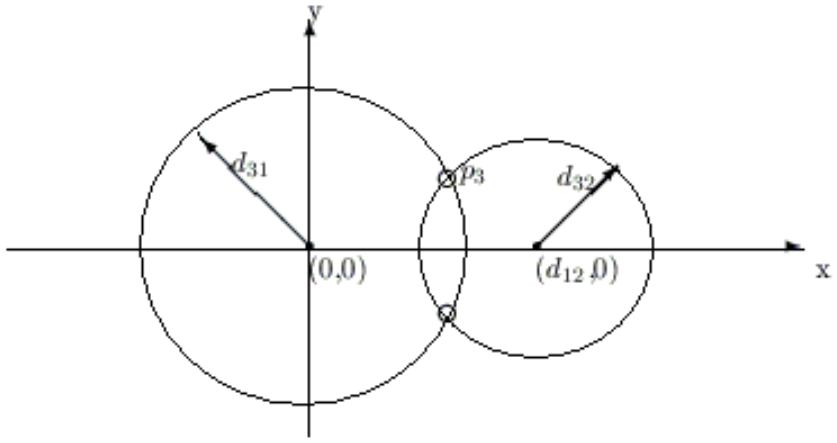
Without loss of generality, $p_3 = (d_{31}cos(\alpha), d_{31}sin(\alpha))$. It is illustrated in Fig. 2.12(a).

Case 2: when triangule inequality is not satisfied. This only happens when some distance measurement is deviated from the true value very badly. Then we either have $d_{31} + d_{32} < d_{12}$ or $|d_{31} - d_{32}| > d_{12}$. In either case, E cannot achieve 0, unlike the first case, the solution is unique and is on the x-axis.

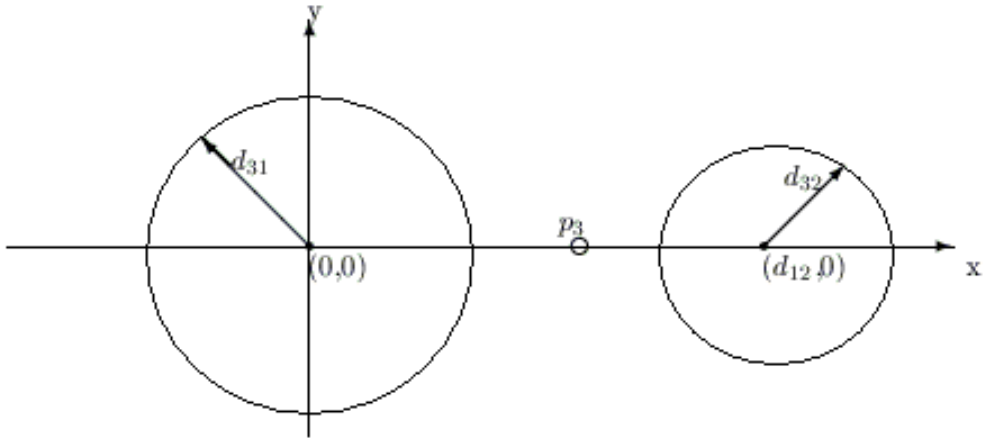
If $d_{31} + d_{32} < d_{12}$, to minimize E we have $p_3 = ((d_{12} + d_{31} - d_{32})/2, 0)$. It is illustrated in Fig. 2.12(b).

If $d_{31} - d_{32} > d_{12}$, to minimize E we have $p_3 = ((d_{12} + d_{31} + d_{32})/2, 0)$. It is illustrated in Fig. 2.12(c). Similarly, if $d_{32} - d_{31} > d_{12}$,$p_3 = (-(d_{12} + d_{31} + d_{32})/2, 0)$.

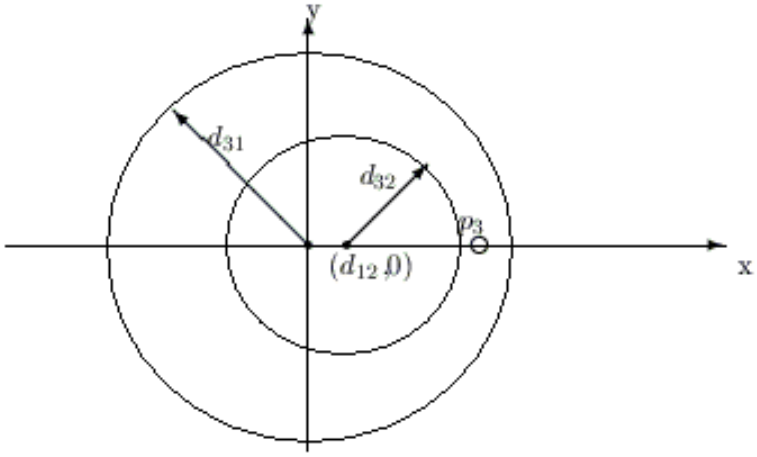
Now given another point p_4, with distance d_{41}, d_{42}, d_{43}, to each of the previous coordinate determined points. Let us give an estimation of the position of p_4. First let us assume p_1, p_2, p_3 are of general setup which means the triangule inequality is satisfied. Then the position of p_4 can be estimated in the following way. At first, let us do the estimation for p_4 using the method stated for p_3. In degenerated case, where p_4



(a) General case



(b) Degenerated case 1



(c) Degenerated case 2

Figure 2.12: Estimation of anchor-free coordinates

falls on the x axis, the best estimation is exactly the same as what we did for p_3. In general, there are still 2 solutions for (x_4, y_4) just as that for (x_3, y_3). But it turns out that the whole picture is no longer symmetric around x-axis because p_3 is on one side of the x-axis. (x_4, y_4) is determined by the following method :

$$(x, y) = (d_{41}cos(\alpha), d_{41}sin(\alpha))$$

$$(x, y') = (d_{41}cos(\alpha), -d_{41}sin(\alpha))$$

$$\text{Where } \alpha = cos^{-1}(\frac{d_{12}^2 + d_{41}^2 - d_{42}^2}{2d_{12}d_{41}})$$

The sign of y coordinate is determined by the following criteria which is also based on the additive Gaussian noise assumption.

$$\begin{aligned} (x_4, y_4) &= (x, y), \qquad (2.73) \\ if &\quad (\|(x, y) - (x_3, y_3)\|_2 - d_{43})^2 < (\|(x, y') - (x_3, y_3)\|_2 - d_{43})^2 \\ (x_4, y_4) &= (x, y'), \text{ otherwise.} \end{aligned}$$

(2.74)

Following the above scheme, we can estimate the positions of points one by one and finally form the set $S_C = \{(x_i, y_i) | \forall i \in C, \}$. At the beginning we mentioned that the distance between the 2 base points d_{12} is crucial for the accuracy. What we can do is to revisit the optimization problem in Eqn.2.69 with d_{12} as the only independent variable. This is an 1 dimensional problem which can be approximately solved despite that the problem has multiple optimums. In Appendix F we will give an example of the multiple local minimums.

2.3.1.2 Refinement of the Coordinates Estimation within a Cell

In the previous section, we get an anchor-free coordinate system for a cell, by the simple triangulation idea. As pointed out, the distance measure between the 2 base points d_{12} is crucial for the accuracy. So we have the simple refinement idea. We randomly pick the sensor-triples K times and get K anchor-free coordinate systems $S_{C1}, ...S_{CK}$. And we also calculate the average of all those anchor-free coordinate systems to get a new anchor-free coordinate system S_{C*}. Then pick the best one from $S_{C1}, ...S_{CK}, S_{C*}$ by means of minimizing the evaluation in Eqn.2.67. In general, the ML estimation is biased as showed in Appendix G.

For the same set of points, given 2 anchor-free coordinate systems A, B with weight w_A, w_B respectively. An anchor-free coordinate system has weight w if it is an average of w S_{Ci}'s. Suppose we have $P_1, ...P_W$, W points in the cell. And in A, the coordinates are $(x_1, y_1), ...(x_W, y_W)$, in B the coordinates are $(u_1, v_1), ...(u_W, v_W)$. Then we first find the rigid transform based on MSE method. The rigid transform $(u, v)^T = R(\alpha)I_r^\gamma(x, y)^T + T$. WHere $R(\alpha)I_r^\gamma, T$ are defined as following.

$$I_r = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, R(\alpha) = \begin{pmatrix} cos(\alpha) & -sin(\alpha) \\ sin(\alpha) & cos(\alpha) \end{pmatrix}, T = \begin{pmatrix} t_x \\ t_y \end{pmatrix} \quad (2.75)$$

And γ takes value in {0,1}, where 1 indicates a reflection of the coordinate system, 0 indicates no reflection. To minimize the MSE we have the following:

$$argmin_{\gamma,\alpha,T} \sum_{i \in S_C} \|R(\alpha)I_r^\gamma p^L 1_i + T - p^L 2_i\|_2^2 \quad (2.76)$$

Due the binary property of γ, the above optimization problem can be stated as following :

$$(\alpha_1, T_1) = argmin_{\alpha,T} \sum_{i \in S_C} \|R(\alpha)(x_i, y_i)^T + T - (u_i, v_i)^T\|_2^2 \quad (2.77)$$

$$(\alpha_2, T_2) = argmin_{\alpha,T} \sum_{i \in S_C} \|R(\alpha)I_r(x_i, y_i)^T + T - (u_i, v_i)^T\|_2^2 \quad (2.78)$$

We pick the better (α, T) of the 2. The solution of Eqn. 2.76 is :

$$(\gamma, \alpha, T) = (0, \alpha_1, T_1), \tag{2.79}$$

$$\text{if } \sum_{i \in S_C} \|R(\alpha_1)(x_i, y_i)^T + T_1 - (u_i, v_i)^T\|_2^2 \leq \sum_{i \in S_C} \|R(\alpha_2) I_r (x_i, y_i)^T + T_2 - (u_i, v_i)^T\|_2^2$$

$$(\gamma, \alpha, T) = (1, \alpha_2, T_2), \text{otherwise.} \tag{2.80}$$

The solution of optimization problems in Eqn. 2.77 and Eqn. 2.78 are unconstrained optimization problems. We include the solution in Appendix H.

Now we have the optimal (in the MSE sense) rigid transform $R(\alpha), I_r^\gamma, T$. The new combined anchor-free coordinate system of A, B with weights w_A, w_B respectively is:

$$(f_i, g_i)^T = \frac{w_A}{w_A + w_B}(R(\alpha) I_r^\gamma (x_i, y_i)^T + T) + \frac{w_B}{w_A + w_B}(u_i, v_i)^T \tag{2.81}$$

Using the average scheme above, we can average all K anchor-free coordinate systems $S_{C1}, ..S_{CK}$ to form S_{C*}. □

2.3.1.3 Gluing of Cells

In this stage, all the cells on the 2D space will be glued together. Every single cell will be glued into the whole anchor-free coordinate system through one of its *adjacent cells.* Gluing anchor-free coordinate systems is also studied in [4] and [39]. But their scheme is by no means optimal [4] or even resulting to non-Euclidean transform [39]. We will give a scheme based on the MSE of 2D positions.

Definition 2.10 (super cells). *A subset S_C of $S \cup F$ is said to be a super cell, if an anchor-free coordinate system is defined for every point in it.*

Definition 2.11 (adjacent super cells). *A pair of super cells S_{C1}, S_{C2} is said to be adjacent if and only if there are at least 3 non-collinear points in $S_{C1} \cup S_{C2}$.*

We will now describe how to glue 2 *adjacent super cells* based on the the minimum squared error method. In the noiseless case, if 2 super cells S_{C1}, S_{C2} are adjacent with anchor-free coordinate system L_{C1}, L_{C2} respectively, we can uniquely determine the coordinates of the points in S_{C1} in the coordinate system L_{C2} and vice versa. This is because 3 non-collinear points with coordinates in 2 different coordinate system can uniquely determine the rigid transformation (any combination of translation, rotation and reflection) between the 2 coordinate systems.

Now we have 2 adjacent super cells S_{C1}, S_{C2} with anchor-free coordinate system L_{C1}, L_{C2} respectively. Let $S_C = S_{C1} \cup S_{C2}$, $P_{L1} = \{p_i^{L2} = (x_i, y_i)$,the coordinate of point i $\in L_{C1} | i \in S_C\}$, and $P_{L2} = p_i^{L2} = (x_i, y_i)$, the coordinate of point i $\in L_{C1} | i \in S_C\}$.

The 2 super cells can be glued together through a rigid transform $R(\alpha) I_r^{\gamma}, T$ which transforms L_{C1} to L_{C2}. Where $R(\alpha), I_r^{\gamma}, T$ are defined in Eqn.2.75/

And γ takes value in {0,1}, where 1 indicates a reflection of the coordinate system, 0 indicates no reflection.

Due to the inaccuracy of the estimation of the anchor-free coordinates in each super cells, we expect the positions of the points in F to be deviated from the true value by a 2D random variable. To minimize the total square error, we have the following optimization problems:

$$argmin_{\gamma,\alpha,T} \sum_{i \in S_C} \| R(\alpha) I_r^{\gamma} p^L 1_i + T - p^L 2_i \|_2^2 \tag{2.82}$$

Due the binary property of γ, the above optimization problem can be stated as following :

$$(\alpha_1, T_1) = argmin_{\alpha,T} \sum_{i \in S_C} \| R(\alpha) p_i^{L1} + T - p_i^{L2} \|_2^2 \tag{2.83}$$

$$(\alpha_2, T_2) = argmin_{\alpha,T} \sum_{i \in S_C} \| R(\alpha) I_r p_i^{L1} + T - p_i^{L2} \|_2^2 \tag{2.84}$$

We pick the better (α, T) of the 2. The solution of Eqn. 2.82 is :

$$\begin{aligned}(\gamma, \alpha, T) &= (0, \alpha_1, T_1), \text{ if } \sum_{i \in S_C} \|R(\alpha_1)p_i^{L1} + T_1 - p_i^{L2}\|_2^2 \leq \\ \sum_{i \in S_C} \|R(\alpha_2)I_r p_i^{L1} + T_2 - p_i^{L2}\|_2^2 \\ (\gamma, \alpha, T) &= (1, \alpha_2, T_2), \text{otherwise.} \end{aligned} \tag{2.85}$$

The optimization problems in Eqn. 2.83 and Eqn. 2.84 are unconstrained optimization problems which are well investigated in computer vision context where it is called registration. We include the solution in Appendix H.

2.3.2 Step 2 : Anchored Coordinates Estimation

From the 1st step of the algorithm, we obtained a rigid anchor-free coordinate system P_L of the point set $S \cup F$, $P_L = \{p_i^L | p_i^L = (x_i, y_i)^T, i \in S \cup F\}$. The 2nd step is about how to embed this coordinate system into the point set F in which every point has known position. Since any rigid transformation is a combination of rotation, translation and reflection. This is very much the same as what we did for gluing 2 adjacent super cells. Again,we can parameterize the problem as:

The positions of F P_F can be embedded through the solid transform $I_r^\gamma R(\alpha), T$. They are defined in Eqn.(H.2).

Using the same argument as gluing the super cells, we have the following optimization problems:

$$argmin_{\gamma,\alpha,T} \sum_{i \in F} \|R(\alpha)I_r^\gamma p_i^L + T - p_i\|_2^2 \tag{2.86}$$

Due the binary property of γ, the above optimization problem can be stated as following :

$$(\alpha_1, T_1) = argmin_{\alpha,T} \sum_{i \in F} \|R(\alpha) p_i^L + T - p_i\|_2^2 \tag{2.87}$$

$$(\alpha_2, T_2) = argmin_{\alpha,T} \sum_{i \in F} \|R(\alpha) I_r p_i^L + T - p_i\|_2^2 \tag{2.88}$$

The solution of Eqn. 2.86 is :

$$\begin{aligned} (\gamma, \alpha, T) &= (0, \alpha_1, T_1), \text{ if } \sum_{i \in F} \|R(\alpha_1) p_i^L + T_1 - p_i\|_2^2 \leq \\ &\sum_{i \in F} \|R(\alpha_2) I_r p_i^L + T_2 - p_i\|_2^2 \\ (\gamma, \alpha, T) &= (1, \alpha_2, T_2), \text{otherwise.} \end{aligned} \tag{2.89}$$

The optimization problem here is exactly the same as what we have in gluing super cells. We will include the solution in the Appendix H.

2.3.3 Summary and further discussion

Given the 2 step method above, we summarize our algorithm as following: a point set F of cardinality larger than 3 with known 2D positions and a point set S with unknown positions. In the first step we can determine the anchor-free coordinate system for $S \cup F$ based on the measured distances between some of the point pairs. In step two, we can embed the 2D positions of set F into the anchor-free coordinate system and reconstruct all the 2D positions of points in S. The first step can be done in a distributed way described. The whole scheme is described as following:

Step 1.1: sensor i estimates the distance between i and $j, j \in adj(i)$, $d_{i,j}$.

Step 1.2: sensor i sends all those $d_{i,j}, j \in adj(j)$ to all those sensors k, $k \in adj(i)$.

Step 1.3: sensor i estimate the anchor-free coordinate system using all the distances measures $d_{i,j}, j \in adj(i)$ and $d_{j,k}, j \in adj(i), k \in adj(i) \bigcap adj(j)$ based on the first step of the localization scheme.

Step 2.1: sensor i sends the anchor-free coordinate system to the fusion center, total $|adj(i)| + 1$ $2D$ positions.

Step 2.2: the fusion center gives an anchored coordinate system estimation by gluing all the anchor-free coordinate systems and the points with known position.

In fact, we do not need to estimate the positions of all those points in S together. Instead we can partition the set S and F into K subsets:

$$S = \cup_{i=1}^{K} S_i, F = \cup_{i=1}^{K} F_i, \tag{2.90}$$

Such that for each i, we can still do the 2 step estimation solely on S_i and F_i, $|F_i| \geq 3$. And it is not necessary for $S_i's$ or $F_i's$ to be disjointed, they could be overlapped. If a point P is in the intersection of K_P sets $S_{P_i}, i = 1, ..., K_P$, there are multiple position estimations for P, $(x_{P_i}, y_{P_i}), i = 1, ..., K_P$. The final position estimation for P is the average of all those estimations. The accuracy can be benefit from this kind of partitions, if the size of the sensor network tends to be too big. That is because when gluing a big network together, the error expands through the whole network, thus the accuracy gets worse.

We summarize the 2 step localization algorithm in the following flowchart in Fig.2.13.

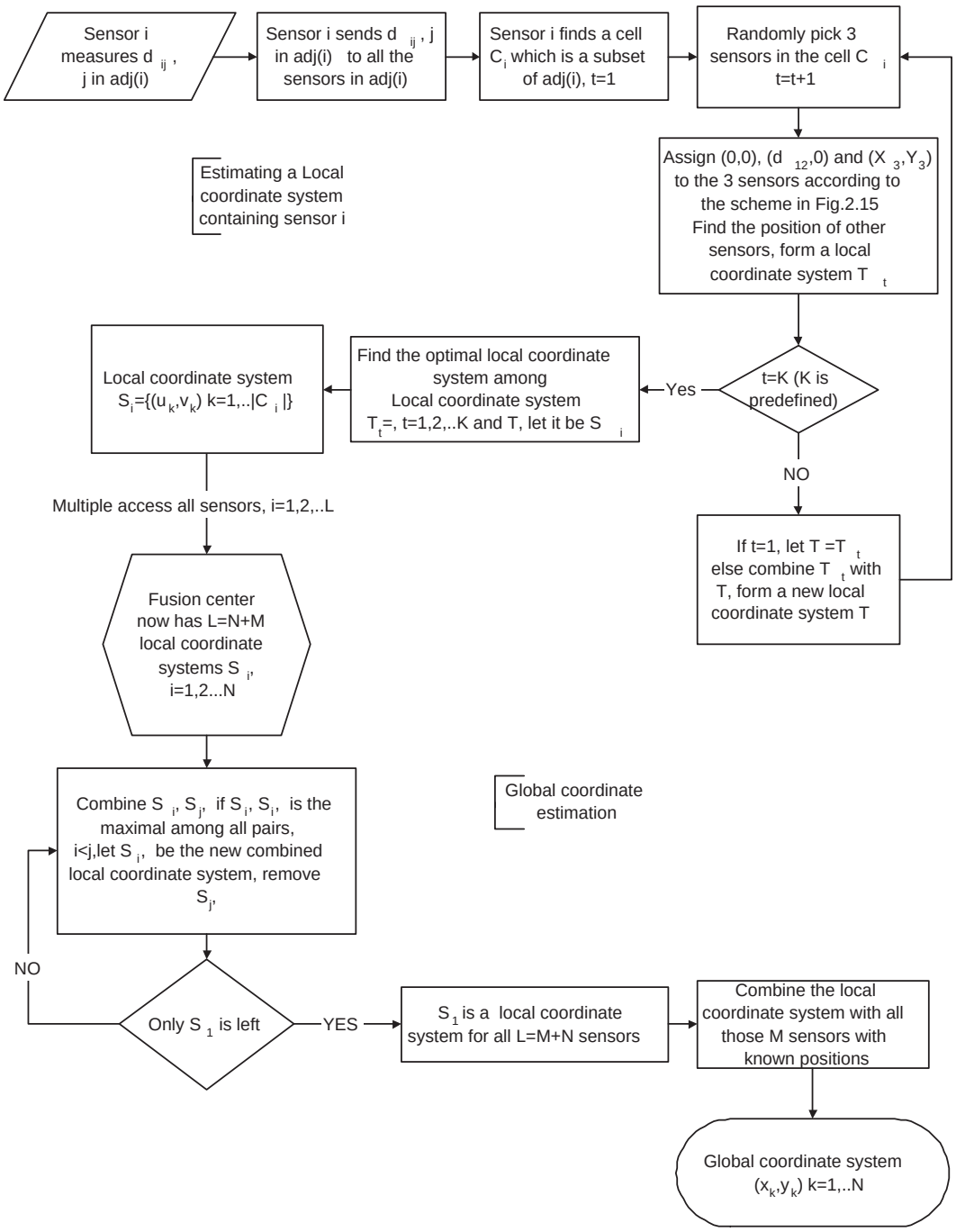


Figure 2.13: 2-step anchored localization scheme

2.3.4 Simulation Results

In this section, we are going to run our anchored localization algorithm for different setups of sensor networks. Like the simulation we did for the Cramer-Rao bound, there are several intrinsic parameters. The variance of the additive Gaussian Noises σ^2, the shape of the sensor network A, the visible radius R, the number of sensors with unknown position M, number of anchors N etc.

In our simulations, we randomly generate M nodes on the plane presumably with unknown position, and N anchors with known position. The position of the points are uniformly distributed inside the region A. We calculate the Cramer-Rao lower bound on the estimation variance $V_x(i), V_y(i)$ and the squared estimation error $(D_x(i), D_y(i)) = ((x_i - \hat{x_i})^2, (y_i - \hat{y_i})^2)$ for nodes $i = 1, 2, ..., M$. In all simulations, the number of tries for each anchor-free coordinate system estimation is $L = 30$.

(i) In the first simulation, we suppose that the distance between i, j is measured for all sensor pairs i, j. A is the region inside the unit circle. In the following figure, we show the Cramer- Rao bound $V_x(i)+V_y(i)$ and squared estimation error $(D_x(i), D_y(i)) = ((x_i - \hat{x_i})^2, (y_i - \hat{y_i})^2)$. In Fig2.14, N=3, M=10. In Fig.2.15, N=4, M=20. In both simulations, $\sigma^2 = 2.5 \times 10^{-3}$.

(ii) In the second simulation,we show the effect of the visible radius. A is the region inside the unit circle. In Fig2.16(a), N=10,M=100, R=0.5. In Fig.2.15, N=10,M=400, R=0.25. In both simulations, $\sigma^2 = 2.5 \times 10^{-3}$.

As can be seen, when the network gets big comparing with the visible radius R, the performance of the algorithm tends to be worse due to the propagation of the errors. Thus first partitioning the entire network and then estimating the positions in each partition is preferable.

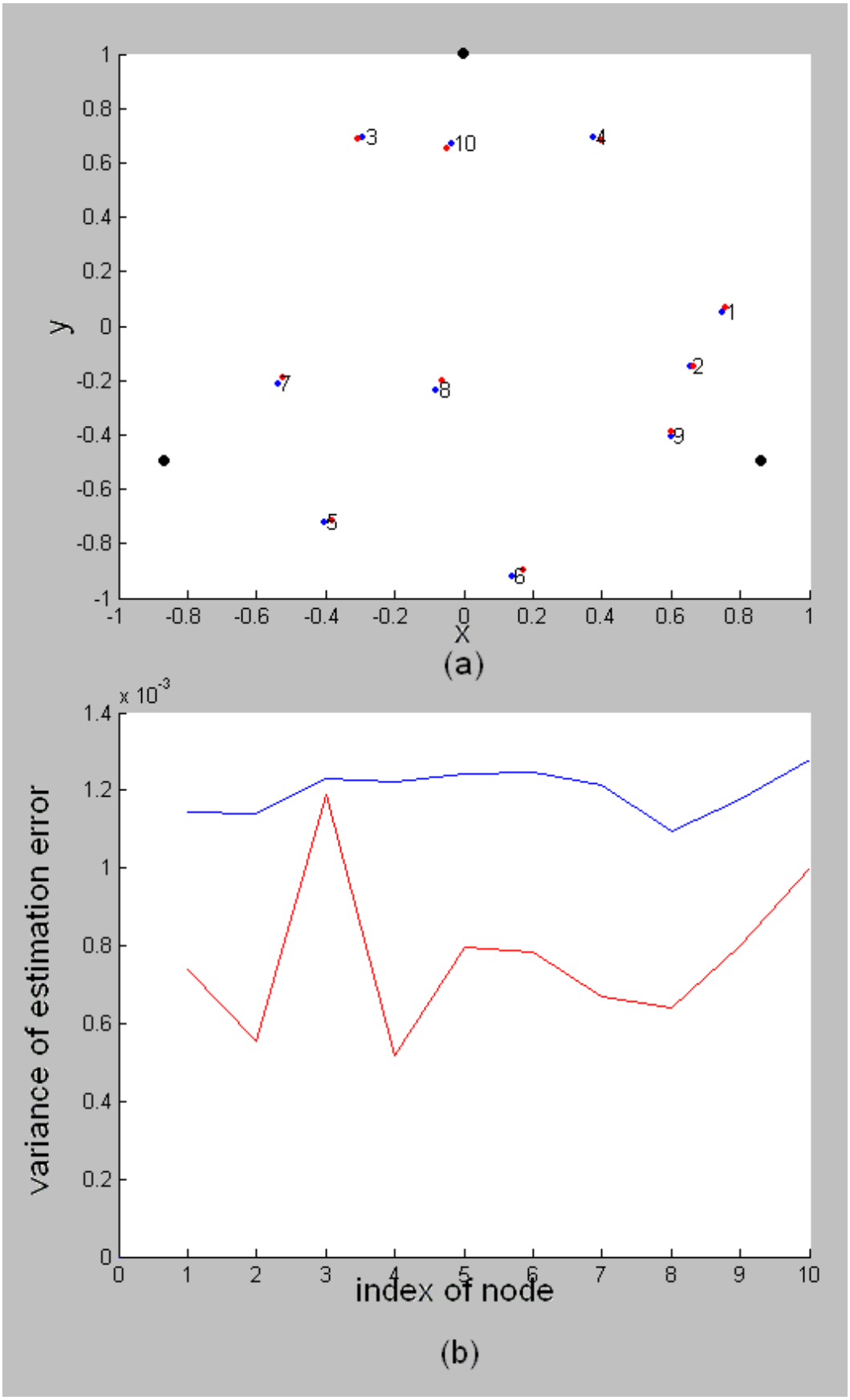


Figure 2.14: N=3,M=10;(a) Black dots: known position, Blue dots: unknown position, red dots: estimation of the position.(b) red: Cramer-Rao bound: $V_x(i) + V_y(i)$, blue: squared error $D_x(i) + D_y(i)$)

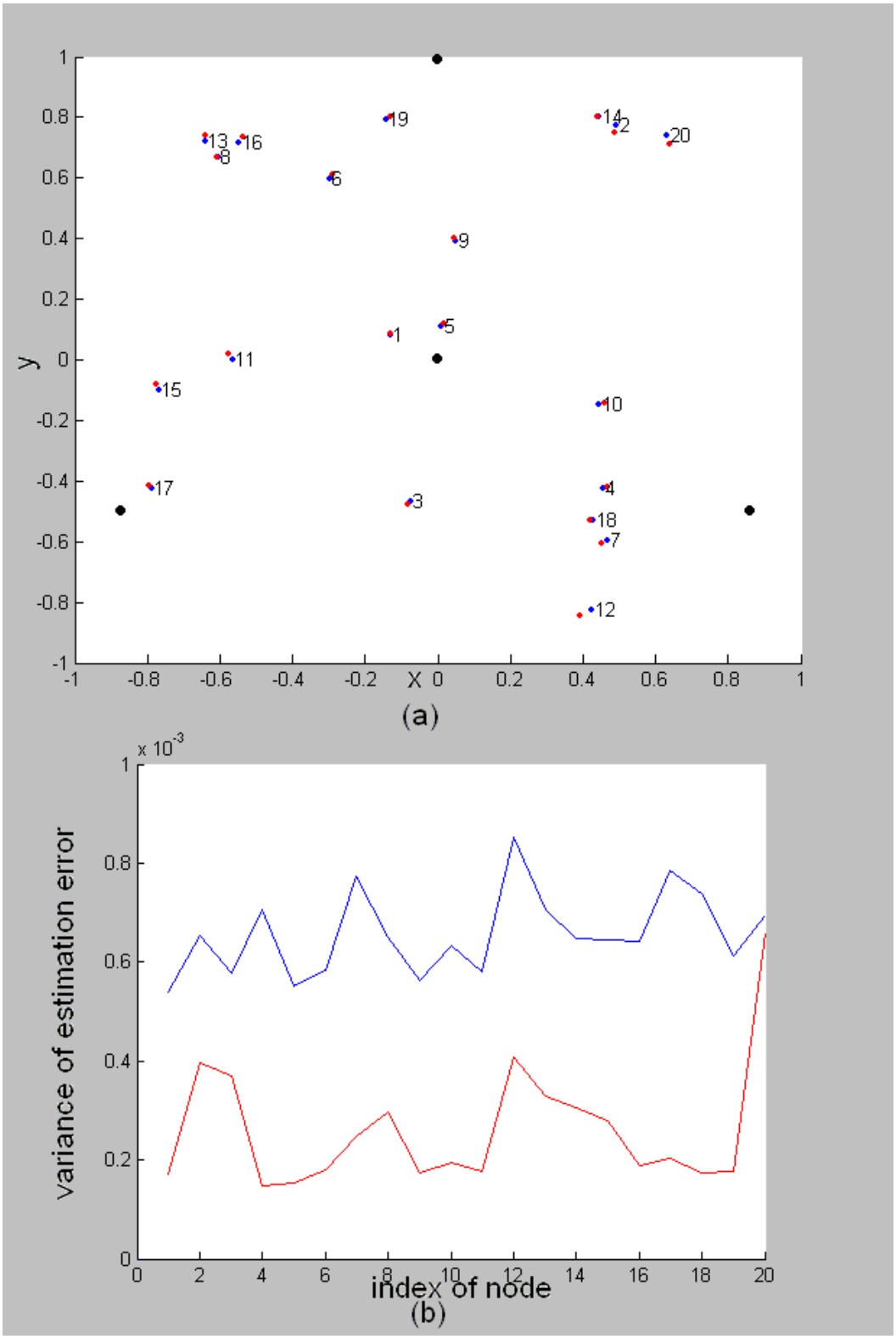


Figure 2.15: N=4,M=20;(a) Black dots: known position, Blue dots: unknown position, red dots: estimation of the position. (b) red: Cramer-Rao bound: $V_x(i) + V_y(i)$, blue: squared error $D_x(i) + D_y(i)$)

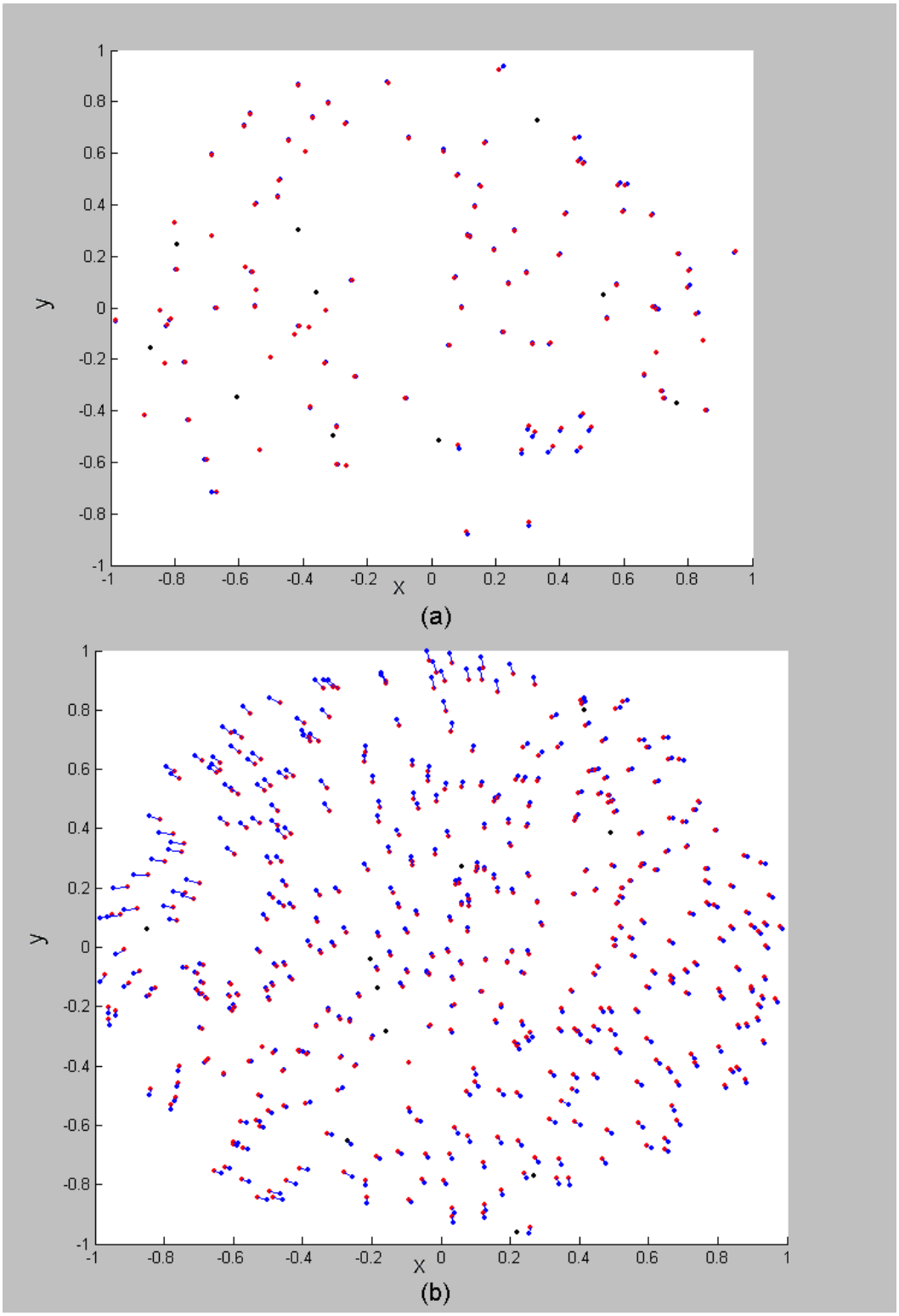


Figure 2.16: Black dots: known position, Blue dots: unknown position, red dots: estimation of the position. (a)N=10,M=100 (b)N=10,M=400

2.4 Conclusions and Future Work

In this chapter, we studied the localization problem and gave a method to compute the Cramer-Rao bound in terms of the geometry of the sensor network. For the anchored localization problem, we showed that the Cramer-Rao bound is invariant under zooming and translation. Also we derived a lower bound on the CR bound that can be computed using only local geometry. We observe that it converges to Cramer-Rao bound if the local area is expanded. This result shows that we can use local geometry to predict the accuracy of the position estimation. For the anchor-free localization problem, due to the singularity of the Fisher Information matrix, we defined the equivalence class estimation problem. Inside this framework we gave a Cramer- Rao-like performance bound. We observed that the per sensor Cramer- Rao-like bound is only dependent on the average number of neighbors. This result again implies that local information can well predict the accuracy of the position estimation.

We also presented a 2-step anchored localization scheme. In this scheme, we first distributedly determine anchor-free coordinate systems in the sensor network Then we combine anchor-free coordinate systems through the optimal (MSE sense) rigid transform.

In our report, we computed the Cramer-Rao bound on the localization problem. But the localization sensitivities to individual observations is still unclear. It might be very helpful to the localization if one can identify the bottleneck of the problem. i.e. how to figure out which distance measure could help to increase the localization accuracy the most. With the knowledge of the bottleneck, we can allocate the power in a smart way through which we can achieve the best localization accuracy.

For the anchor-free coordinate estimation, we showed that the optimization problem is non-convex. We believe that randomized optimization algorithms (genetic algorithm, stimulate annealing etc.) might achieve better performance. When combining

multiple anchor-free coordinate systems together, the estimation error in an anchor-free coordinate system can propagate. How the error propagates is important but still open. Our anchored localization scheme is realized in a semi-distributed way. A distributed localization algorithm requires less communication thus is more preferable. How to reduce the communication requirement of the localization scheme is an interesting problem.

Chapter 3

Tracking Objects by a Sensor Network

In this chapter we will study the object tracking by a sensor network problem. We assume that the positions of the sensors are known, and the multipath distance measurements are collected to estimate the position of the object(s). We will analyze the problem in different scenarios based on the number of the transmitters, the number of the receivers and the number of objects. We will derive the Cramer-Rao lower bound for each scenario and propose estimation schemes for object tracking.

3.1 Tracking Single Object in a Single Transmitter, Single Receiver Sensor Network (STSR)

In a single transmitter, single receiver sensor (STSR) network, if only the multi-path length d (the distance from the transmitter Tx to the object plus the distance from the object to the receiver Rx) is available, it is impossible to estimate the position of the object. This is illustrated in Fig.3.1. The only thing we can say about the position of the object is that it is on the ellipse which is formed by points p, s.t. $d_2(p, p_{Tx})+d_2(p, p_{Rx}) = d$, where d_2 is the Euclidean distance, p_{Tx}, p_{Rx} are the positions of transmitter and receiver respectively. This kind of problem can be thought as a bi-static radar system in multi-static literature [5]. And the Doppler effect or angular information is often used together with the TOA (time of arrival) in the position estimation.

However, it is possible to estimate the position of the object with some presumed

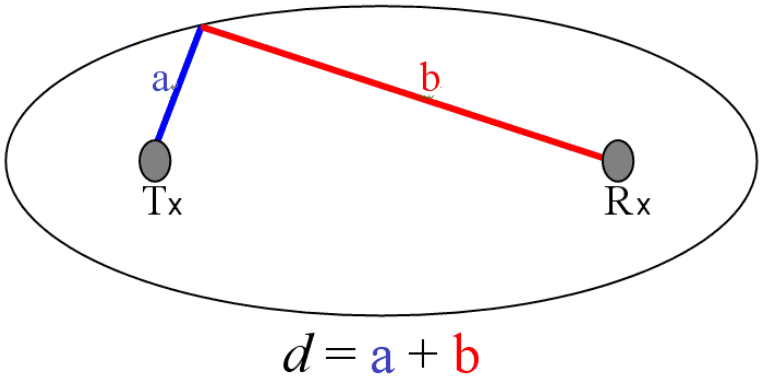


Figure 3.1: $a = d_2(p, p_{Tx}), b = d_2(p, p_{Rx})$

motion model of the object. The motion model could be constant velocity or constant acceleration (Appendix I). In our report, we assume that the object moves with a constant velocity. To determine the position of an object with a constant velocity in a time interval $[a, b]$, only the starting position at time 0 and velocity are needed, or equivalently the starting position and the ending position of the object. We will use the latter in our report, i.e. we want to estimate the starting position and the ending position of the object from the multi-path length measurements, given the constant velocity assumption. The setup is illustrated in Fig. 3.2.

We will first prove that a linear algorithm can estimate the exact motion of the object when the multi-path distance measurements are noiseless. Then we will discuss the stability of the algorithm.

3.1.1 A Simple Estimation Scheme

If the multi-path length measurements are noise free and the object moves strictly at a constant velocity, we can determine the starting position and ending position of the object perfectly from only 6 measurements with equal time interval in between consecutive measurements. We state as the following theorem.

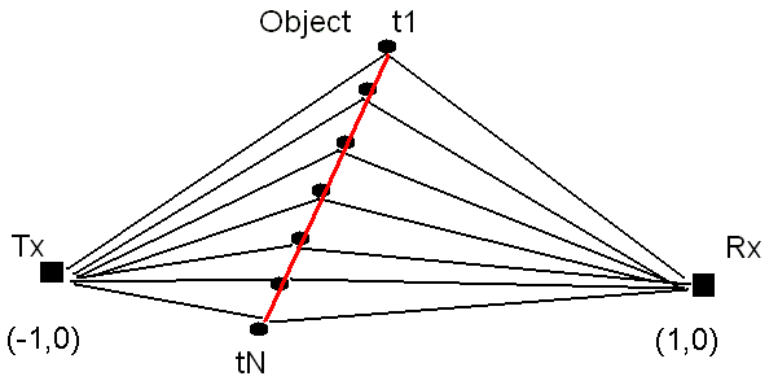


Figure 3.2: The object is moves with a constant velocity

Theorem 3.1. *STSR object tracking In a single receiver, single transmitter network, the positions of the sensors are known. Suppose that the object moves with a constant velocity, then the motion of the object can in general be tracked by N, $N \geq 6$ multipath distance measures.*

Proof. : Choose coordinate so that the transmitter is at $(-1, 0)$ and receiver is at $(1, 0)$. At time t, the multi-path distance is d. Then we have the following equation for the position of the object (x, y):

$$\sqrt{(x+1)^2 + y^2} + \sqrt{(x-1)^2 + y^2} = d \tag{3.1}$$

As shown in Fig.1.2, (x, y) is on the ellipse with the following ellipse equation:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 \tag{3.2}$$

Where $a = \frac{d}{2}, b = \sqrt{(\frac{d}{2})^2 - 1}$. With the constant velocity assumption, we assume that multi-path length measurements are at time $0, T, 2T, ..NT$, where T is the time difference between 2 consecutive measures. Then we have the following linear map:

$$(x_k, y_k) = \frac{N-k}{N}(x_0, y_0) + \frac{k}{N}(x_N, y_N) \tag{3.3}$$

Where (x_k, y_k) is the position of the object at time kT, in particular (x_0, y_0) and (x_N, y_N) are the starting and ending position respectively (given $N + 1$ measurements). With multi-path length d_k, the object with coordinate at time kT is on the ellipse:

$$\frac{(\frac{N-k}{N}x_0 + \frac{k}{N}x_N)^2}{a_k^2} + \frac{(\frac{N-k}{N}y_0 + \frac{k}{N}y_N)^2}{b_k^2} = 1 \quad (3.4)$$

$k = 0, 1, ...N$ We have the following linear equation system:

$$\begin{aligned} & {b_k}^2((1-\frac{k}{N})^2{x_0}^2 + 2(1-\frac{k}{N})\frac{k}{N}x_0x_N + (\frac{k}{N})^2{x_N}^2) \\ + \quad & {a_k}^2((1-\frac{k}{N})^2{y_0}^2 + 2(1-\frac{k}{N})\frac{k}{N}y_0y_N + (\frac{k}{N})^2{y_N}^2) = {a_k}^2{b_k}^2 \end{aligned} \quad (3.5)$$

$k = 0, 1, 2, .., N$

$$\text{Let } v = (x_0{}^2, x_0x_N, x_N{}^2, y_0{}^2, y_0y_N, y_N{}^2)^T \quad (3.6)$$

$$g = (a_0^2b_0^2, ...a_N^2b_N^2)^T \quad (3.7)$$

and an $(N+1) \times 6$ matrix A_N, s.t. the $k-th$ row (k=0,1,2,..,N) of A_N is

$$({b_k}^2(1-\frac{k}{N})^2, 2{b_k}^2(1-\frac{k}{N})\frac{k}{N}, {b_k}^2(\frac{k}{N})^2, {a_k}^2(1-\frac{k}{N})^2, 2{a_k}^2(1-\frac{k}{N})\frac{k}{N}, {a_k}^2(\frac{k}{N})^2) \quad (3.8)$$

Then we have:

$$A_N v = g \quad (3.9)$$

Then if A_N is of rank 6, we have

$$v = (A_N^T A_N)^{-1} A_N^T g \quad (3.10)$$

□

We note that there are three cases when the matrix A_N is not of full rank (6) even if $N \geq 5$. These degenerate cases happen when the target moves along one of the x or y axes. The degenerate cases are shown in Appendix J. In general as long as $N + 1 \geq 6$, we can solve the above linear equation. And from Eqn.3.6, we can solve the

unknowns $x_0, y_0 x_N, y_N$. Notice that there are 4 solutions of $x_0, y_0 x_N, y_N$ for a given 6-tuple $({x_0}^2, x_0 x_N, {x_N}^2, {y_0}^2, y_0 y_N, {y_N}^2)$. And this is the nature of the single transmitter, single receiver tracking problem by noticing that (x_0, y_0, x_N, y_N), $(-x_0, y_0, -x_N, y_N)$, $(x_0, -y_0, x_N, -y_N)$ and $(-x_0, -y_0, -x_N, -y_N)$ all give the same multipath distance measure at any time.

3.1.1.1 Stability Analysis

Suppose that the multi-path length measurements d_k's are corrupted by an additive noise δ_k, assumed to be iid noise $\sim N(0, \sigma^2)$. $d_k = d_k^{'} + \delta_k$, where $d_k^{'}$ is the true multi-path length. Notice that in Eqn.3.9, d_k appears in both A_N and g which makes the matrix $A^T A$ not noise free. In practice the sensitivity to noise of the matrix A_N must be taken into account. A measure of the sensitivity to the noise is given by the following quantity that is also known as condition number for matrices.

$$\chi(A_N) = \|A_N\|_2 \|(A_N^T A_N)^{-1} A_N^T\|_2 \tag{3.11}$$

This roughly tells how close the matrix A_N is to a singular matrix. The larger the $\chi(A_N)$, the closer the matrix A is to a singular matrix. We have numerically computed $\chi(A_N)$ for different A_N's corresponding to different 5-tuples $(x_0, x_N, y_0, y_N, N+1)$. It turns out that $\chi(A_N)$ converges to a finite real value when N goes to infinity as shown in Appendix K, write that limit $\chi(A)$. Unfortunately, the limit is generally very large which means that the linear equation system is very sensitive to the noises. The fact that the limit $\chi(A)$ is also very big shows that no matter how many measurements are taken, the simple linear scheme can not give a robust estimation of the motion. We also found that $\chi(A_N)$ does not change significantly with N when $N \gg 1$. And $\chi(A_N)$ is very sensitive to the 4-tuple (x_0, x_N, y_0, y_N). Fig. 3.3 shows contour plots of $log(\chi(A))$ as a function of the position of the object, for $N \gg 1$, i.e. the limit (which occurs at around $N = 20$), and when the target moves 1 towards the origin $(0, 0)$, and 2

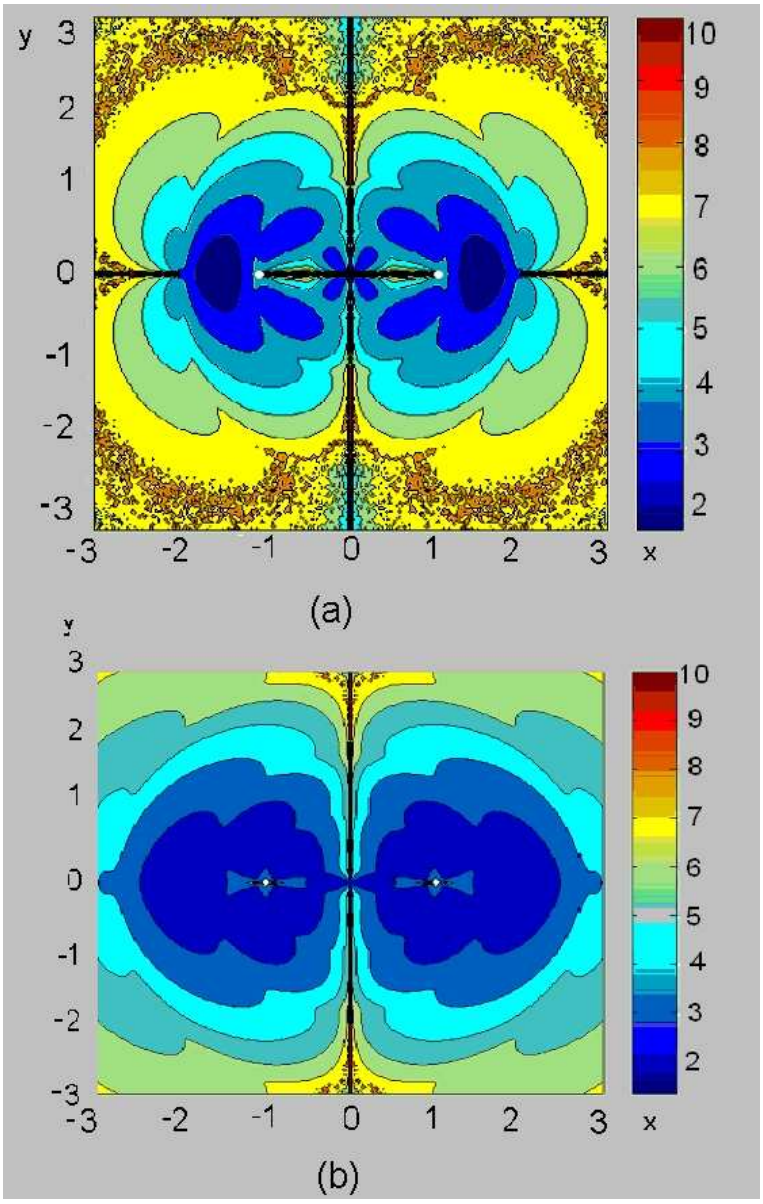


Figure 3.3: The value on $(x, y) \in [-3, 3] \times [-3, 3]$ indicates $log_{10}(\chi(A))$, (a) is of the 4-tuple $(x, y, x(1 - \frac{1}{\sqrt{x^2+y^2}}), y(1 - \frac{1}{\sqrt{x^2+y^2}}))$. i.e. the object moves 1 towards the origin $(0, 0)$. (b) is of the 4-tuple $(x, y, x(1 - \frac{2}{\sqrt{x^2+y^2}}), y(1 - \frac{2}{\sqrt{x^2+y^2}}))$. i.e. the object moves 2 towards the origin $(0, 0)$. The 2 white points are transmitter and receiver at $(-1, 0), (1, 0)$.. they are 2 away from each other. The black part indicates that A is absolutely singular.

respectively, as before the transmitter and the receiver are at $(-1,0),(1,0)$ respectively.

$\chi(A)$ is very large in both cases, a little noise in the multi-path measurements will result a huge error in the solution in Eqn.3.9. Thus the simple algorithm does not work well in practice.

3.1.2 Cramer-Rao bound analysis

In this section, we derive the theoretical estimation variance bound on the single transmitter single receiver, single object tracking. Again we have the multi-path length measurements corrupted by iid zero mean gaussian noises. Suppose that the measuring starts at time 0 when the object is at (x,y), ends at time 1 when the object is at (u,v). Then at time t the object is at $(sx+tu, sy+tv)$, where $s=1-t$. Then we have the following observations :

$$d_t = \sqrt{(sx+tu-1)^2+(sy+tv)^2} + \sqrt{(sx+tu+1)^2+(sy+tv)^2} + \delta_t \tag{3.12}$$

Where δ_t's are iid gaussian noise $\sim N(0,\sigma^2)$. If we have N measurements with equal time interval between consecutive measurements, then $t=0, \frac{1}{N-1}, \frac{2}{N}, ..., \frac{N-2}{N-1}, 1$.

Theorem 3.2. *Fisher Information Matrix of STSR object tracking For a specific motion which starts from (x,y), ends at (u,v). If there are N multipath distance measurements as in Eqn.3.12. Let x,y,u,v be the 1st, 2nd, 3rd and 4th parameter to be estimated respectively. Let $J_{4\times4}$ be the Fisher Information Matrix for x,y,u,v.*

$$lim_{N\to\infty}\frac{\sigma^2 J}{N} = K(x,y,u,v) \tag{3.13}$$

Where $K(x,y,u,v)$ is a function of (x,y,u,v).

Proof. : Let the observation vector $\overrightarrow{d} = (d_{t_1}, ...d_{t_N})$. Then the probability density

function is :

$$p(\overrightarrow{d}\,|x,y,u,v) = \prod_t \frac{1}{\sqrt{2\pi\sigma^2}} exp(\frac{-(d_t - \sqrt{(sx+tu-1)^2+(sy+tv)^2} - \sqrt{(sx+tu+1)^2+(sy+tv)^2})^2}{2\sigma^2}) \tag{3.14}$$

Log likelihood function:

$$\begin{aligned} ln(p(\overrightarrow{d}\,|x,y,u,v)) &= \frac{N}{2}ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_t (d_t - \sqrt{(sx+tu-1)^2+(sy+tv)^2} \\ &\quad -\sqrt{(sx+tu+1)^2+(sy+tv)^2})^2 \end{aligned} \tag{3.15}$$

The Fisher information matrix $J_{4\times4}$ is as following.

$$\begin{aligned} j_{11} &= E(\frac{\partial^2 ln(p(\overrightarrow{d}\,|x,y,u,v))}{\partial x^2}) \\ &= \frac{1}{\sigma^2}\sum_t (\frac{s(sx+tu-1)}{\sqrt{(sx+tu-1)^2+(sy+tv)^2}} + \frac{s(sx+tu+1)}{\sqrt{(sx+tu+1)^2+(sy+tv)^2}})^2 \\ &= \frac{1}{\sigma^2}\sum_t (cos(\alpha_t)+cos(\beta_t))^2 s^2 \end{aligned} \tag{3.16}$$

Similarly we have the expression for J_{ii}:

$$J_{22} = \frac{1}{\sigma^2}\sum_t (sin(\alpha_t)+sin(\beta_t))^2 s^2 \tag{3.17}$$

$$J_{33} = \frac{1}{\sigma^2}\sum_t (cos(\alpha_t)+cos(\beta_t))^2 t^2 \tag{3.18}$$

$$J_{44} = \frac{1}{\sigma^2}\sum_t (sin(\alpha_t)+sin(\beta_t))^2 t^2 \tag{3.19}$$

and J_{ij}'s:

$$J_{12} = J_{21} = \frac{1}{\sigma^2}\sum_t (sin(\alpha_t) + sin(\beta_t))(cos(\alpha_t) + cos(\beta_t))s^2 \tag{3.20}$$

$$J_{13} = J_{31} = \frac{1}{\sigma^2}\sum_t (cos(\alpha_t) + cos(\beta_t))^2 st \tag{3.21}$$

$$J_{14} = J_{41} = \frac{1}{\sigma^2}\sum_t (sin(\alpha_t) + sin(\beta_t))(cos(\alpha_t) + cos(\beta_t))st \tag{3.22}$$

$$J_{23} = J_{32} = \frac{1}{\sigma^2}\sum_t (sin(\alpha_t) + sin(\beta_t))(cos(\alpha_t) + cos(\beta_t))st \tag{3.23}$$

$$J_{24} = J_{42} = \frac{1}{\sigma^2}\sum_t (sin(\alpha_t) + sin(\beta_t))^2 st \tag{3.24}$$

$$J_{34} = J_{43} = \frac{1}{\sigma^2}\sum_t (sin(\alpha_t) + sin(\beta_t))(cos(\alpha_t) + cos(\beta_t))t^2 \tag{3.25}$$

Where $\alpha_t, \beta_t \in [0, 2\pi)$ and

$$\begin{aligned} cos(\alpha_t) &= \frac{sx + tu - 1}{\sqrt{(sx + tu - 1)^2 + (sy + tv)^2}}; sin(\alpha_t) = \frac{sy + tv}{\sqrt{(sx + tu - 1)^2 + (sy + tv)^2}}; \\ cos(\beta_t) &= \frac{sx + tu + 1}{\sqrt{(sx + tu + 1)^2 + (sy + tv)^2}}; sin(\beta_t) = \frac{sy + tv}{\sqrt{(sx + tu + 1)^2 + (sy + tv)^2}}; \end{aligned} \tag{3.26}$$

The geometric interpretation of α_t and β_t is illustrated in Fig 3.9.

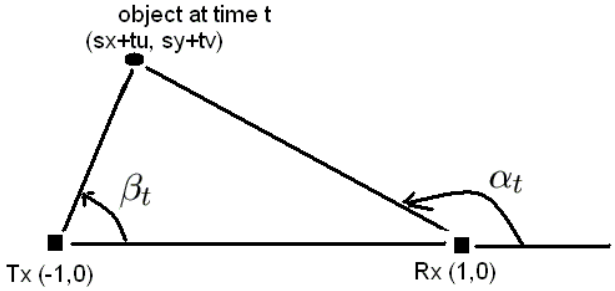


Figure 3.4: α_t and β_t

If N is sufficiently large, notice that $s^2(cos(\alpha_t)+cos(\beta_t))^2$ is a bounded continuous

function of t, then we have the following result:

$$\begin{aligned} J_{11} &= \frac{1}{\sigma^2}\sum_t (cos(\alpha_t)+cos(\beta_t))^2 s^2 \\ &= \frac{N}{\sigma^2}\sum_t \frac{1}{N}(cos(\alpha_t)+cos(\beta_t))^2 s^2 \\ &\simeq \frac{N}{\sigma^2}\int_0^1 (cos(\alpha_t)+cos(\beta_t))^2 s^2 dt \end{aligned} \tag{3.27}$$

Similarly for J_{ij}'s. Thus the Fisher information matrix J have the following form when N is sufficiently large (when the sum converges to the integral).

$$lim_{N\to\infty}\frac{\sigma^2 J}{N} = K(x,y,u,v) \tag{3.28}$$

□

Where K is a constant matrix. We call it normalized Fisher Information matrix. It is independent of σ^2 and N.It is solely determined by the 4-tuple (x,y,u,v). The normalized Fisher Information matrix depicts the intrinsic estimation bound for the 4-tuple (x,y,u,v).

$$\begin{aligned} K_{11} &= \int_0^1 (cos(\alpha_t)+cos(\beta_t))^2 s^2 dt \\ K_{12} = K_{21} &= \int_0^1 (sin(\alpha_t)+sin(\beta_t))(cos(\alpha_t)+cos(\beta_t)) s^2 dt \end{aligned} \tag{3.29}$$

And so on for $K_{13}, ...$ Now we can calculate the Cramer-Rao bound on the estimation.

$$\begin{aligned} V(x) = \frac{\sigma^2}{N}{K^{-1}}_{11}; V(y) = \frac{\sigma^2}{N}{K^{-1}}_{22}; \\ V(u) = \frac{\sigma^2}{N}{K^{-1}}_{33}; V(v) = \frac{\sigma^2}{N}{K^{-1}}_{44}; \end{aligned} \tag{3.30}$$

We can numerically compute $K(x,y,u,v)$ and have an idea of how many multi-path length measurements are needed to achieve a certain estimation variance ξ^2 for all of x,y,u,v.

$$\boxed{N \geq \frac{\sigma^2}{\xi^2} max_i K_{ii}^{-1}} \tag{3.31}$$

In Fig.3.1.2 we show some numerical results for $max_i K_{ii}^{-1}$.

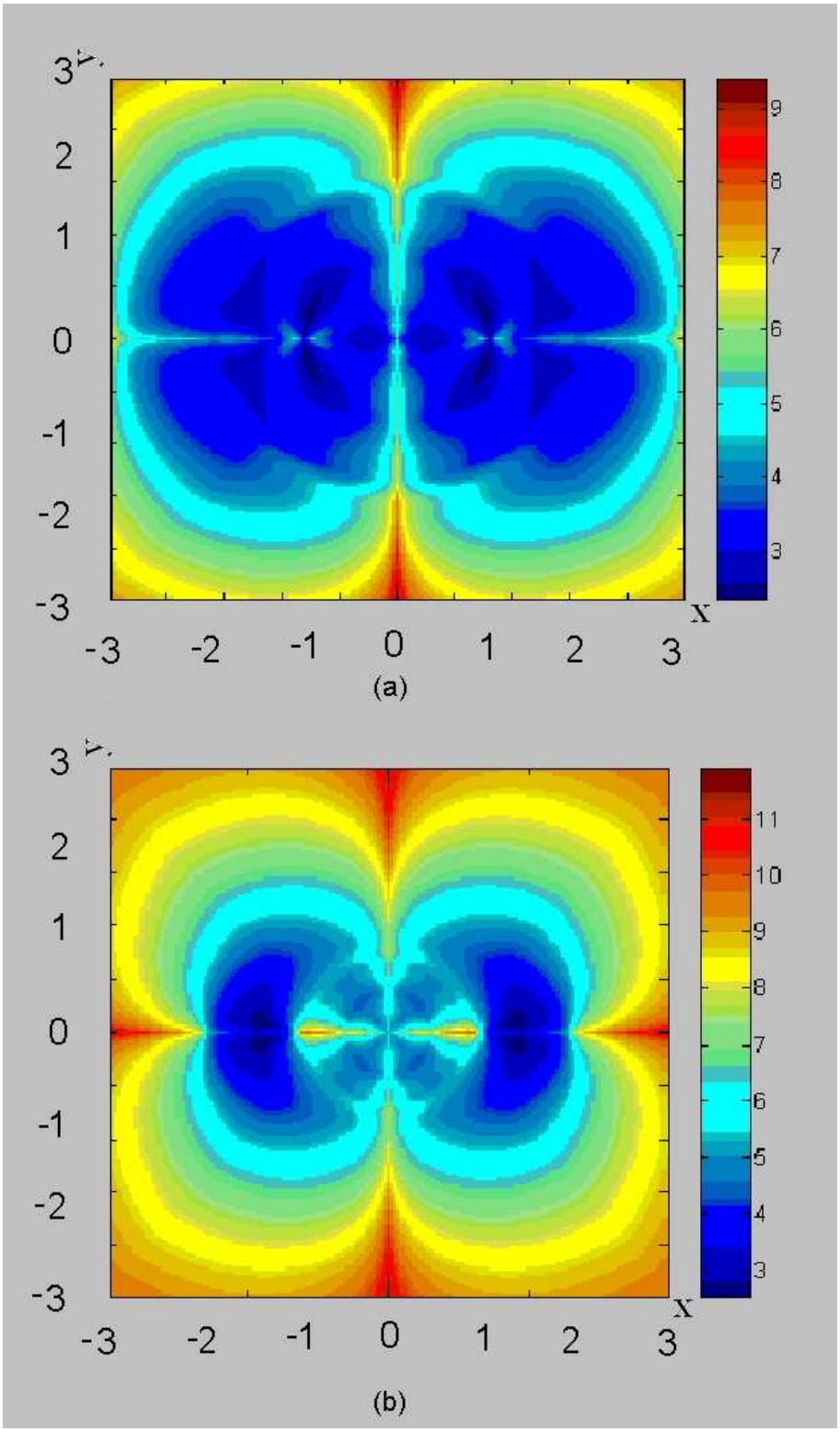


Figure 3.5: The value on (x, y) indicates $log_{10}(max_i {K^{-1}}_{ii})$
(a) is of the 4-tuple $(x, y, x(1-\frac{2}{\sqrt{x^2+y^2}}), y(1-\frac{2}{\sqrt{x^2+y^2}}))$. i.e. the object moves 2 towards the origin $(0, 0)$.
(b) is of the 4-tuple $(x, y, x(1-\frac{1}{\sqrt{x^2+y^2}}), y(1-\frac{1}{\sqrt{x^2+y^2}}))$. i.e. the object moves 1 towards the origin $(0, 0)$. Transmitter and receiver at $(-1, 0), (1, 0)$.

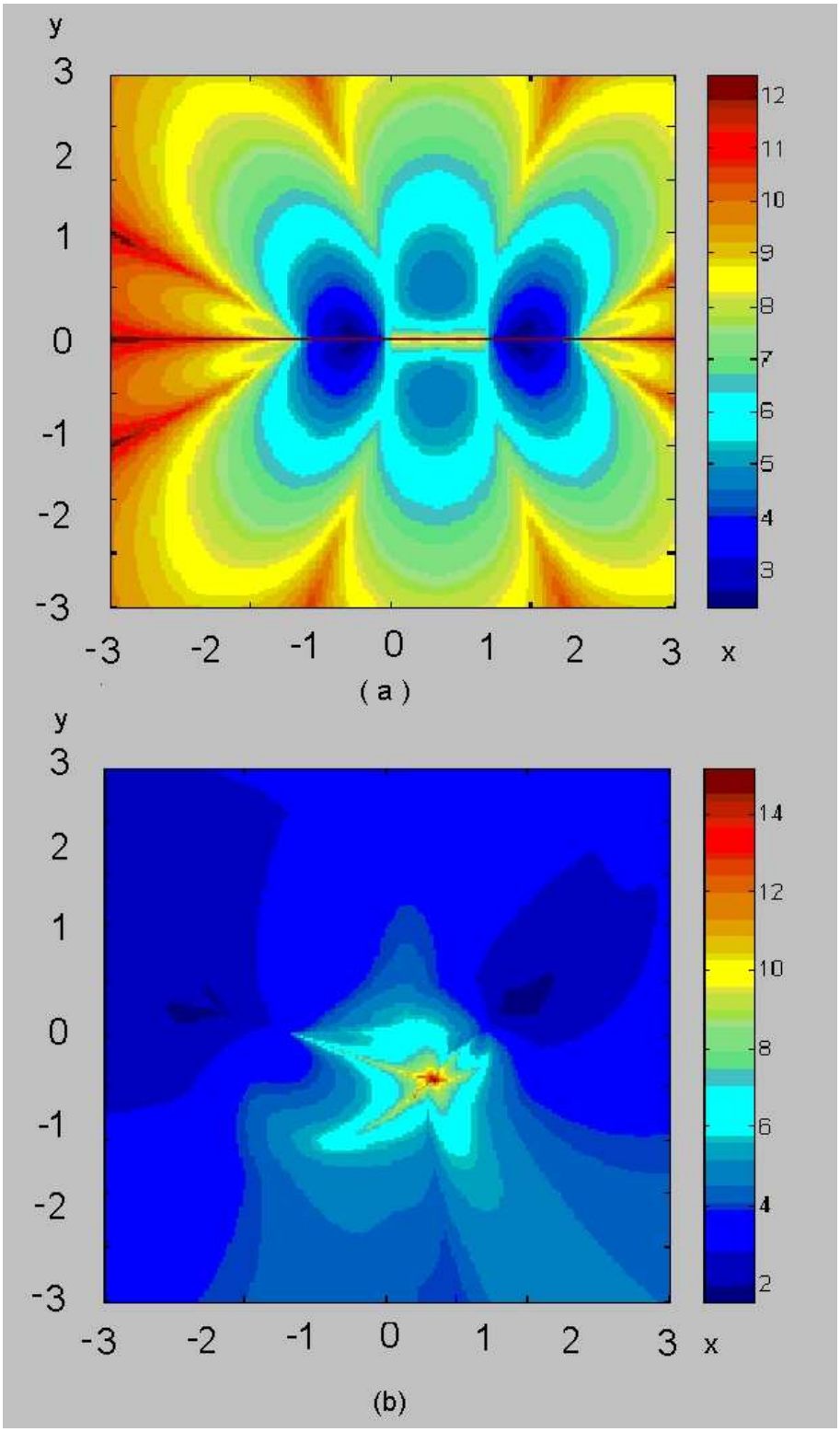


Figure 3.6: The value on (x, y) indicates $log_{10}(max_i {K^{-1}}_{ii})$, (a) is of the 4-tuple $(x, y, x-1, y)$. i.e. the object moves 1 leftward along the x-axis. (b) is of the 4-tuple $(x, y, 0.5, -0.5)$. i.e. the motion of the object ends at $(0.5, -0.5)$. Transmitter and receiver at $(-1, 0), (1, 0)$.

3.1.3 A Consistent Estimation Scheme

Suppose that the additive gaussian noises added to the multi-path measurements have fixed variance δ^2. Unlike the naive linear algorithm which cannot accurately estimate the motion even with infinite many distance measurements, we propose an estimation algorithm, s.t. the estimation converges to the true value as the number of multi-path length measurements goes to infinity. Notice that although we fixed the noise variance on each measure, the overall SNR increases to infinity since we have more measurements. We define SNR as the ratio of the total signal power over noise variance. $SNR = \frac{NP}{\sigma^2}$, where N is the number of measurements, P is the power used per measurement, σ^2 is the noise variance.

The algorithm has two steps. In the first step, we average the multi-path length measurements, which will give an estimation of a single multi-path length with very low estimation variance by the law of large numbers. More specifically, without loss of generality, we assume the starting time is 0, ending time is 1. We divide the time interval $[0, 1]$ into n^3 sub intervals with the same size $1/n^3$. In the beginning of each time interval, we have a multipath distance measure $d_j, j = 1, ..n^3$. For convenience, let $n = 7m, m \in N$. Then we will average multipath distances around $T_i = i/7, i = 1, 2, ...6$, let

$$l_i = \frac{1}{n+1} \sum_{j=\frac{n^3 i}{7}-\frac{n}{2}}^{j=\frac{n^3 i}{7}-\frac{n}{2}} d_j \tag{3.32}$$

We will prove that l_i converges to the true multipath distance at time $i/7$ with probability 1 when n goes to infinity.

In the second step we use these very accurate multipath distances to form the matrix A and vector g by the same way as in the previous section. Then we argue that the matrix A and the vector g will be arbitrarily *close* to the true value given enough measurements. Then we use Eqn.3.10 to solve for the unknowns x_s, y_s, x_e, y_e,

where (x_s, y_s) is the starting location, and (x_e, y_e) is the ending location of the object. Following flowchart shows how the scheme works.

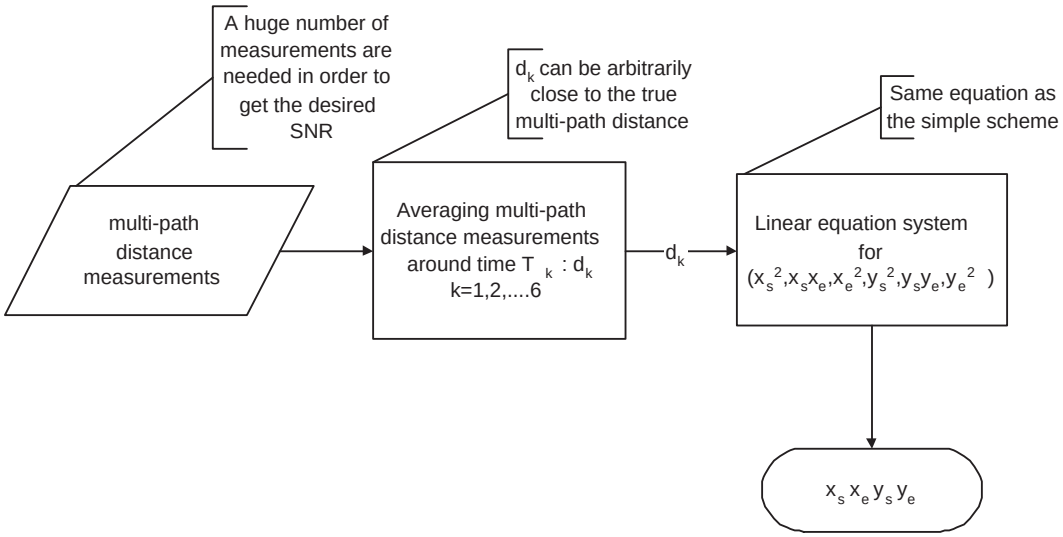


Figure 3.7: Flowchart of a consistent scheme

Step 1: distance averaging

In the first step, we average a bunch of multi-path distance measurements around 6 different times to get good approximations of the multi-path distances around 6 times.

For the convenience, we write $M = n^2, N = n$. Then there are $L = MN$ multi-path distance measurements:

$$l_1^1, l_1^2, ..., l_1^M, l_2^1, l_2^2, ..., l_2^M, ..., l_N^1, l_N^2 ... l_N^M \tag{3.33}$$

$$Where\ l_k^j = \sqrt{(x_k^j - 1)^2 + {y_k^j}^2} + \sqrt{(x_k^j + 1)^2 + {y_k^j}^2} + \delta_k^j = h_k^j + \delta_k^j \tag{3.34}$$

$$And\ let\ L_T = \sqrt{(x_1^1 - x_N^M)^2 + (y_1^1 - y_N^M)^2}. \tag{3.35}$$

$\delta_k^{j\prime}s$ are iid Gaussian random noises with variance δ^2. L_T is the total travelling path length of the object. And h_k^j is the true multi-path lengths. Our goal is to estimate h_k^j from a group of l_k^j's.

Now let $d_k = \frac{1}{M}\sum_{j=1}^{M} l_k^j$ be the average of M noisy multipath distance measures.

Theorem 3.3. *$lim_{N\to\infty,M\to\infty} d_k = h_k^c$, where $c = \frac{M+1}{2}$, h_k^c is the true $kN + c$'th multipath distance.*

Proof. : We have:

$$d_k = \frac{\sum_{j=1}^{M} h_k^j}{M} + \frac{\sum_{j=1}^{M} \delta_k^j}{M} = l_k + \sigma_k \tag{3.36}$$

Where $\sigma_k's$ are iid Gaussian random noises with variance $\frac{\delta^2}{M}$.

For convenience, we assume that M is an odd number. Let $c = (M+1)/2$, then l_k is an approximation of h_k^c. Using Taylor expansion, we have :

$$\begin{aligned} h_k^{c+q} &= h_k^c + q\triangle x\frac{\partial h_k^c}{\partial x} + q\triangle y\frac{\partial h_k^c}{\partial y} + (q\triangle x)^2\frac{\partial^2 h_k^c}{2\partial x^2} + (q\triangle y)^2\frac{\partial^2 h_k^c}{2\partial y^2} \\ &+ (q\triangle x)(q\triangle y)\frac{\partial^2 h_k^c}{\partial x\partial y} + o((\frac{L_T}{N})^2) \end{aligned} \tag{3.37}$$

Where $x = x_k^c, y = y_k^c$, $q = -\frac{M-1}{2}, -\frac{M-1}{2}+1, ..., \frac{M-1}{2}-1, -\frac{M-1}{2}$, and $\triangle x = \frac{L_T cos(\alpha)}{MN}$, $\triangle y = \frac{L_T sin(\alpha)}{MN}$, α is the angle between the $x-axis$ and the travelling path of the object. Obviously $\triangle x, \triangle y$ are upper-bounded by $\frac{L_T}{NM}$ and $q\triangle x, q\triangle y$ are upper-bounded by $\frac{L_T}{N}$.

From Eqn.(3.37), we have :

$$l_k = \frac{1}{M}\sum_{j=1}^{M} h_k^j = \frac{1}{M}\sum_{q=-\frac{M-1}{2}}^{\frac{M-1}{2}} h_k^{c+q} \tag{3.38}$$

$$\begin{aligned} &= h_k^c + \frac{1}{M}\sum_{q=-\frac{M-1}{2}}^{\frac{M-1}{2}}((q\triangle x)^2\frac{\partial^2 h_k^c}{2\partial x^2} + (q\triangle y)^2\frac{\partial^2 h_k^c}{2\partial y^2} \\ &\quad +(q\triangle x)(q\triangle y)\frac{\partial^2 h_k^c}{\partial x\partial y}) \\ &= h_k^c + \triangle_k \end{aligned} \tag{3.39}$$

Again let $x_k^c = x; y_k^c = y;$ then we have $h_k^c = \sqrt{(x-1)^2+y^2} + \sqrt{(x+1)^2+y^2}$

$$\frac{\partial^2 h_k^c}{\partial y\partial y} = \frac{1}{\sqrt{(x-1)^2+y^2}} + \frac{y^2}{\sqrt{(x-1)^2+y^2}^3} + \frac{1}{\sqrt{(x+1)^2+y^2}} + \frac{y^2}{\sqrt{(x+1)^2+y^2}^3}$$

$$|\frac{\partial^2 h_k^c}{\partial y\partial y}| \le \frac{2}{\epsilon} + 2 = P_\epsilon \tag{3.40}$$

$$\text{Where } \epsilon = min(\sqrt{(x-1)^2+y^2}, \sqrt{(x+1)^2+y^2}) \tag{3.41}$$

Similarly we have $|\frac{\partial^2 h_k^c}{\partial x \partial y}| \leq \frac{1}{\epsilon} + 1, |\frac{\partial^2 h_k^c}{\partial x^2}| \leq \frac{2}{\epsilon} + 2$.

Suppose that the travelling path does not pass through the transmitter or receiver, more strictly speaking, the minimum distance from the travelling path to the transmitter and receiver is lower bounded by some positive real value e, then from Eqn.(3.40), we know that $|\frac{\partial^2 h_k^c}{\partial x \partial y}|, |\frac{\partial^2 h_k^c}{\partial y^2}|, and\ |\frac{\partial^2 h_k^c}{\partial x^2}|$ are upper-bounded by P_e. Thus we have :

$$\begin{aligned}
|\triangle_k| &\leq \frac{P_e}{M} \sum_{q=-\frac{M-1}{2}}^{\frac{M-1}{2}} (\frac{(q\triangle x)^2}{2} + \frac{(q\triangle y)^2}{2} + |(q\triangle x)(q\triangle y)|) \\
&= \frac{P_e}{M}(\frac{(\triangle x)^2}{2} + \frac{(\triangle y)^2}{2} + |\triangle x \triangle y|) \sum_{q=-\frac{M-1}{2}}^{\frac{M-1}{2}} q^2 \\
&= \frac{P_e}{M}(\frac{(\triangle x)^2}{2} + \frac{(\triangle y)^2}{2} + |\triangle x \triangle y|)(\frac{M}{2} - 1)(\frac{M}{2} + 1)\frac{M}{3} \\
&\leq \frac{P_e}{24M}((\triangle x)^2 + (\triangle y)^2 + 2|\triangle x \triangle y|)M^3 \\
&\leq \frac{P_e}{24M}((\frac{L_T}{MN})^2 + (\frac{L_T}{MN})^2 + 2(\frac{L_T}{MN})^2)M^3 \\
&= \frac{P_e L_T^2}{6N^2} \\
&= \frac{Q}{N^2} \qquad (3.42)
\end{aligned}$$

Where $Q = \frac{P_e L_T^2}{6}$, now we have $d_k = h_k^c + \sigma_k + \triangle_k$, where $|\triangle_k| \leq \frac{Q}{N^2}, \sigma_k \sim N(0, \frac{\delta^2}{M})$. In the above analysis d_k and h_k^j are all functions of M and N. We write them as $d_k(M, N)$ and $h_k^j(M, N)$. We let $n = N, n^2 = M$ because we can arbitrarily pick N, M. Using the standard Chebychev inequality, it is easy to verify that the sequence $d_k)$ converges to h_k^c with probability 1, and $c = \frac{n^2-1}{2}$. □

Step 2: Solving the unknowns

Now we pick 6 *target points*,$p_1, ...p_6$, where p_i is at (u_i, w_i).

$$(u_i, w_i) = \frac{i}{7}(x_e, y_e) + \frac{7-i}{7}(x_s, y_s) \qquad (3.43)$$

Let r_i be the multi-path length of the i'th measurement,then $r_i = \sqrt{(u_i - 1)^2 + w_i^2} + \sqrt{(u_i + 1)^2 + w_i^2}$. Now Eqn.3.9 has the following form.

$$Av = g \qquad (3.44)$$

Where $v = ({x_s}^2, x_s x_e, {x_e}^2, {y_s}^2, y_s y_e, {y_e}^2)^T$, $g = (a_1^2 b_1^2, ...a_6^2 b_6^2)^T$ and a 6×6 matrix A, s.t the $k-th$ row (k=1,2,..,6) of it is
$({b_k}^2(1-\frac{k}{7})^2, 2{b_k}^2(1-\frac{k}{7})\frac{k}{7}, {b_k}^2(\frac{k}{7})^2, {a_k}^2(1-\frac{k}{7})^2, 2{a_k}^2(1-\frac{k}{7})\frac{k}{7}, {a_k}^2(\frac{k}{7})^2)$, and $a_k = \frac{r_k}{2}, b_k = \sqrt{(\frac{r_k}{2})^2 - 1}$.

As we know from step 1, we can estimate (with error converges to 0) those r_k's by averaging a bunch of multi-path length measurements.

Now Eqn.3.9 have the following form :

$$A_L v = g_L \tag{3.45}$$

Where $v = ({x_s}^2, x_s x_e, {x_e}^2, {y_s}^2, y_s y_e, {y_e}^2)^T$, $g_L = (a_{n1}^2 b_{n1}^2, ...a_{n6}^2 b_{n6}^2)^T$ and a 6×6 matrix A_L, s.t the $k-th$ row (k=1,2,..,6) of it is
$({b_{n_k}}^2(1-\frac{n_k}{L})^2, 2{b_{n_k}}^2(1-\frac{n_k}{L})\frac{n_k}{L}, {b_{n_k}}^2(\frac{n_k}{L})^2, {a_{n_k}}^2(1-\frac{n_k}{L})^2, 2{a_{n_k}}^2(1-\frac{n_k}{L})\frac{n_k}{L}, {a_{n_k}}^2(\frac{n_k}{L})^2)$, where $n_k = \rho(\frac{kL}{7})$ and $a_{n_k} = \frac{d_{n_k}(n^2,n)}{2} b_{n_k} = \sqrt{(\frac{d_{n_k}(n^2,n)}{2})^2 - 1}$, where $\rho(x)$ is the closest integer to x and $L = n^3$.

From step 1, we know that $d_{n_k}(n^2, n)$ converges to r_k with probability 1. So we have a sequence of linear equations $A_L v = g_L$, where A_L, g_L are defined by $d_{nk}(n^2, n), k = 1, 2, ..., 6$, thus A_L, g_L converges to A, g with probability 1. A is nonsingular, so $A_L^{-1} g_L$ converges to $A^{-1} g$ with probability 1. We only use $6n^2$ measurements out of n^3 total measurements. □

We just proved that no matter how large the variance of a single multi-path length measurements is, we still can accurately estimate the motion of the object given very large amount of measurements. This is only of theoretical interest. Practically the number of multi-path distance measurements needs to be huge in order to have an accurate estimation of the position, because the matrix A is near singular. Furthermore, the motion model fails if the object moves randomly or it is at rest. In Appendix L, we will show that if the object is allowed to move with non-constant velocity, it is possible that 2 different motions give the same multi-path distance all the time in a

single transmitter, single receiver network. That's why it is impossible for a single transmitter, single receiver network to track the object when the object does not move with a constant velocity.

3.2 Single Object Tracking in a Multiple-sensor Sensor Network (MSSO)

From the previous section, we know that even the object moves strictly with a constant velocity, it is still not practical to estimate the position of the object if only 1 transmitter and 1 receiver are presented in the sensor network. So we move on to a sensor network with multiple transmitter, multiple receivers and no motion model is presumed for the objects. We will first derive the Cramer-Rao bound and then give a 2-step estimation scheme on the estimation of the object position in the multiple-sensor scenario.

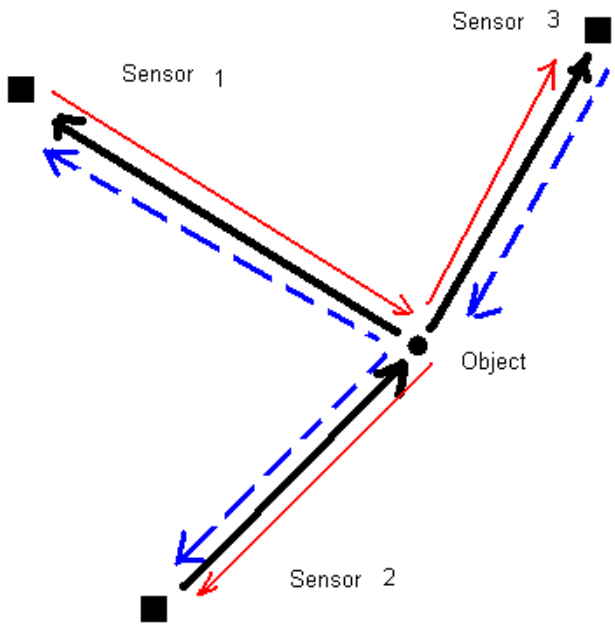


Figure 3.8: Multiple-Sensors, Single Object

As shown in Fig.3.8 we have N sensors with known positions and 1 object with

unknown position in the filed, where each sensor can both send and receive signals. There are $N(N-1)$ multi-path distance measurements if the sensors can not receive their own signal (in fact, this assumption is a minor one which does not affect the algorithm or the Cramer-Rao bound analysis). From the Cramer-Rao bound analysis, it can be shown that the variance of the estimation converges to 0 with order $O(1/N^2)$ and that is also the convergence rate of the estimation variance that our algorithm achieves. Here we assume all the multi-path distance measurements are taken simultaneously, i.e. there is not motion of the object. It is reasonable since the motion of the objects is typically slow comparing to the time differences $sup_{i,j}\ inf_{n,m}|T_i(n) - T_j(m)|$, where $T_i(n)$ is the n th signal sent out by sensor i.

3.2.1 Cramer-Rao Bound

In this section we derive the lower bound on the position estimation variance of the multiple-sensor, single object system. Here we are going to derive the Cramer-Rao Bound for a general system with N trasmitters and M receivers.

Suppose that there is a sensor network with N transmitters with known positions $(x_i, y_i), i = 1, ...N$, M receivers with known positions $(x_j^{'}, y_j^{'}), j = 1, ...M$. And the position of the object is (x, y) which is unknown. The observation vector $\overrightarrow{l} = (l_{11}, l_{12}, ...l_{ij}, ..., l_{NM})$, vector of all the multi-path distances. Assume the observations are corrupted by iid Gaussian noises $\varepsilon \sim N(0, \sigma^2)$, then

$$l_{ij} = \sqrt{(x - x_i)^2 + (y - y_i)^2} + \sqrt{(x - x_j^{'})^2 + (y - y_j^{'})^2} + \varepsilon_{ij} \tag{3.46}$$

The estimation of (x, y) is based on the knowledge of the positions of the sensors and the observation vector $\overrightarrow{l}$. Then we have the probability density function as following:

$$p(\overrightarrow{l}|x, y) = \prod_{1 \leqslant i \leqslant N, 1 \leqslant j \leqslant M} \frac{1}{\sqrt{2\pi\sigma^2}} exp(\frac{-(l_{ij} - \sqrt{(x - x_i)^2 + (y - y_i)^2} - \sqrt{(x - x_j')^2 + (y - y_j')^2})^2}{(2\sigma^2)}) \tag{3.47}$$

The Fisher information matrix $J_{2\times 2}$ is as following.

$$\begin{aligned} ln(p(\overrightarrow{l}|x,y)) &= \frac{MN}{2}ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i,j}(l_{ij} - \\ &\quad \sqrt{(x-x_i)^2+(y-y_i)^2} - \sqrt{(x-x_j^{'})^2+(y-y_j^{'})^2})^2 \qquad (3.48) \\ J_{11} &= E(\frac{\partial^2 ln(p(\overrightarrow{l}|x,y))}{\partial x^2}) \\ &= \frac{1}{\sigma^2}\sum_{i,j}(\frac{x-x_i}{\sqrt{(x-x_i)^2+(y-y_i)^2}} + \frac{x-x_j^{'}}{\sqrt{(x-x_j^{'})^2+(y-y_j^{'})^2}})^2 \\ &= \frac{1}{\sigma^2}\sum_{i,j}(cos(\alpha_i)+cos(\alpha_j^{'}))^2 \qquad (3.49) \end{aligned}$$

Similarly :

$$J_{22} = \frac{1}{\sigma^2}\sum_{i,j}(sin(\alpha_i)+sin(\alpha_j^{'}))^2 \qquad (3.50)$$

$$J_{21} = J_{12} = \frac{1}{\sigma^2}\sum_{i,j}(cos(\alpha_i)+cos(\alpha_j^{'}))(sin(\alpha_i)+sin(\alpha_j^{'})) \qquad (3.51)$$

We define normalized Fisher Information Matrix $K = \sigma^2 J$, where K is independent of σ^2, K is the intrinsic measure of how hard it is to estimate the position of an object at point (x, y).

Where $\alpha, \alpha^{'} \in [0, 2\pi)$ and

$$cos(\alpha) = \frac{x-x_i}{\sqrt{(x-x_i)^2+(y-y_i)^2}}; sin(\alpha) = \frac{y-y_i}{\sqrt{(x-x_i)^2+(y-y_i)^2}};$$
$$cos(\alpha^{'}) = \frac{x-x_i^{'}}{\sqrt{(x-x_i^{'})^2+(y-y_i^{'})^2}}; sin(\alpha^{'}) = \frac{y-y_i^{'}}{\sqrt{(x-x_i^{'})^2+(y-y_i^{'})^2}};$$

The geometric interpretation of α is illustrated in Fig.3.9.

Now we can calculate the Cramer-Rao bound on the position estimation. For a 2×2 Fisher information matrix $J_{2\times 2}$, we have the Cramer-Rao bound on the estimation of x, y as following.

$$V(x) = \frac{J_{22}}{J_{11}J_{22}-J_{12}^2}; V(y) = \frac{J_{11}}{J_{11}J_{22}-J_{12}^2} \qquad (3.52)$$

The Fisher information matrix J is now a random matrix. The randomness comes from the position of the sensors. i.e. for different set up of the sensor network *and* the

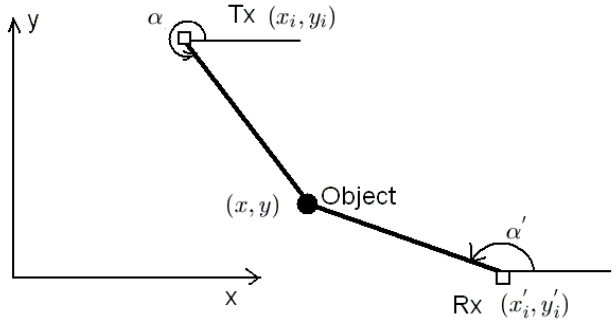


Figure 3.9: α

position of the object, the Cramer-Rao bound could be different. We assume α_i and $\alpha_j^{'}$ are iid uniformly distributed in $[0, 2\pi)$. This is reasonable when there are a large amount of sensors uniformly distributed in the sensor field. Notice that with that assumption, $V(x)$ and $V(y)$ have the same statistical properties by the nature of symmetry. α's are iid uniformly distributed, then we have the following results:

Theorem 3.4. *Asymptotic Cramer-Rao bound for MSSO If the transmitters and receivers are iid angularly uniformly distributed around the object at (x, y). Then*

$$lim_{N\to\infty,M\to\infty}(V(x)+V(y))NM = 2\sigma^2 \quad \textit{in probability} \tag{3.53}$$

Proof. :

$$E(cos(\alpha_i)cos(\alpha_j)) = E(sin(\alpha_i)sin(\alpha_j)) = \frac{\delta(i-j)}{2} \tag{3.54}$$

$$E(cos(\alpha_i)sin(\alpha_j)) = 0 \tag{3.55}$$

$$E(cos(\alpha_i)^4) = E(sin(\alpha_i)^4) = \frac{3}{8} \tag{3.56}$$

$$E(cos(\alpha_i)^2cos(\alpha_j)^2) = E(cos(\alpha_i)^2sin(\alpha_j)^2) = \frac{1}{4},\ \text{i} \neq j \tag{3.57}$$

$$E(cos(\alpha_i)^2sin(\alpha_i)^2) = \frac{1}{8} \tag{3.58}$$

Where $\delta(k) = 1$, if $k = 0$,$\delta(k) = 0$, if $k \neq 0$. Now we have the following results for the entries of the Fisher information matrix:

$$
\begin{aligned}
E(J_{11}) &= E(J_{22}) = E(\frac{1}{\sigma^2}\sum_{i,j}(cos(\alpha_i)+cos(\alpha_j^{'}))^2) \\
&= \frac{1}{\sigma^2}(\sum_{i,j}E(cos(\alpha_i)^2)+\sum_{i,j}E(cos(\alpha_j^{'})^2)+\sum_{i,j}E(cos(\alpha_i)cos(\alpha_j^{'}))) \\
&= \frac{1}{\sigma^2}(\frac{NM}{2}+\frac{NM}{2}+0) = \frac{NM}{\sigma^2} \qquad (3.59) \\
E(J_{12}) = E(J_{21}) &= \frac{1}{\sigma^2}E(\sum_{i,j}(cos(\alpha_i)+cos(\alpha_j^{'}))(sin(\alpha_i)+sin(\alpha_j^{'}))) = 0 \qquad (3.60) \\
E(J_{11}J_{22}) &= \frac{1}{\sigma^4}E(\sum_{i,j}(cos(\alpha_i)+cos(\alpha_j^{'}))^2\sum_{i,j}(sin(\alpha_i)+sin(\alpha_j^{'}))^2) \\
&= \frac{1}{\sigma^4}(N^2M^2-\frac{NM^2+N^2M}{8}) \qquad (3.61) \\
E(J_{12}^2) &= \frac{1}{\sigma^4}E((\sum_{i,j}(cos(\alpha_i)+cos(\alpha_j^{'}))(sin(\alpha_i)+sin(\alpha_j^{'})))^2) \\
&= \frac{1}{\sigma^4}(\frac{3N^2M+3NM^2}{8}+\frac{NM}{4}) \qquad (3.62) \\
E(J_{11}J_{22}-J_{12}^2) &= \frac{1}{\sigma^4}(N^2M^2-\frac{NM^2+N^2M}{2}-\frac{NM}{4}) \qquad (3.63)
\end{aligned}
$$

Now substitute the above results into Eqn.3.52, noticing that N, M is large so the variance goes to zero and apply the law of large numbers. We have the following result :

$$
\begin{aligned}
&lim_{N\to\infty,M\to\infty}\frac{J_{11}}{NM} = \frac{1}{\sigma^2} \text{ in probability} \\
&lim_{N\to\infty,M\to\infty}\frac{J_{11}J_{22}-{J_{12}}^2}{N^2M^2} = \frac{1}{\sigma^4} \text{ in probability}
\end{aligned} \qquad (3.64)
$$

So we have

$$
lim_{N\to\infty,M\to\infty}NMV(x) = lim_{N\to\infty,M\to\infty}NM\frac{J_{22}}{J_{11}J_{22}-{J_{12}}^2} = \sigma^2 \text{ in probability} \qquad (3.65)
$$

□

We claim that the Cramer-Rao bound on the position estimation (x, y) is $V(x) = V(y) \sim \frac{\sigma^2}{NM}$. The variance of the position estimation is

$$
\boxed{V(x)+V(y) = \frac{2\sigma^2}{NM}} \qquad (3.66)
$$

In Fig.3.10, we show the theoretical Cramer-Rao bounds vs. simulation results (Monte-Carlo method) in logarithm terms. In our simulation we simulate the Cramer-Rao bound based on realized sensor positions. We fix the object at $[0, 0]$, N transmitters and N receivers are uniformly distributed in $[-1, 1] \times [1, 1]$, for each N we compute $V(x) + V(y)$ for 100 times and get the average.

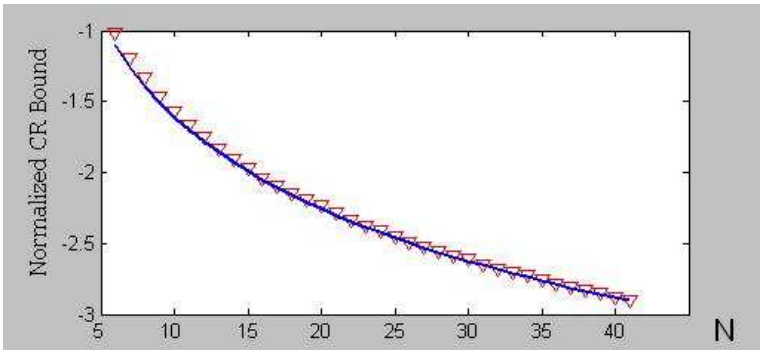


Figure 3.10: Blue curve is the theoretical CR bound in logarithm term :$log10(\frac{2}{N^2})$, red triangles are simulation results

3.2.2 A Position Estimation Scheme

In this section, we assume that all the sensors can both transmit and receive signals. We propose a least square based algorithm as following which achieves the optimal convergence rate $O(1/N^2)$. The algorithm has 2 steps. In the first step, we extract the distances $d_i, 1 \leq i \leq N$, between the object and the $i-th$ sensor from the multi-path lengths l_{jk}, where $1 \leq j \neq k \leq N$. In the second step, we estimate the position (x, y) of the object from the distance estimations $d_i's$. The first step is a linear estimator, but as a whole the estimation is *not* a linear estimation from the observations.

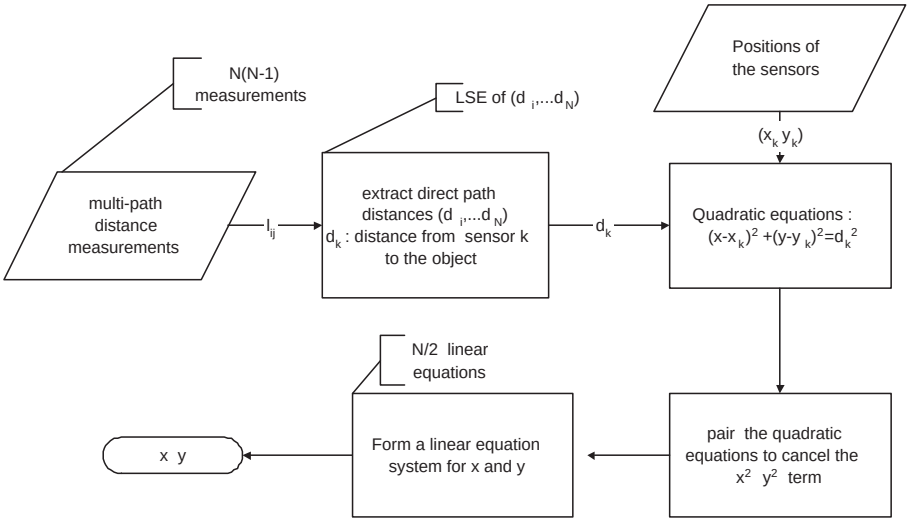


Figure 3.11: Flowchart of MSSO position estimation

3.2.2.1 Extraction of d_i's

In the first stage of the estimation scheme, we estimate the distances from each sensor to the object from the multi-path distances measurements. Notice that $l_{ij} = d_i + d_j + \varepsilon_{ij}$, where l_{ij}'s are the observed multi-path length and ε_{ij}'s are iid Gaussian random variables $\sim N(0, \sigma^2)$, for all $1 \leq j \neq k \leq N$. Thus we have the following equation system:

$$A\vec{d} + \vec{\varepsilon} = \begin{pmatrix} 1\,1\,0\,0 & \dots & 0\,0 \\ 1\,1\,0\,0 & \dots & 0\,0 \\ 1\,0\,1\,0 & \dots & 0\,0 \\ 1\,0\,1\,0 & \dots & 0\,0 \\ \vdots & \ddots & \vdots \\ 0\,0\,0\,0 & \dots & 1\,1 \\ 0\,0\,0\,0 & \dots & 1\,1 \end{pmatrix} \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_N \end{pmatrix} + \begin{pmatrix} \varepsilon_{12} \\ \varepsilon_{21} \\ \varepsilon_{13} \\ \varepsilon_{31} \\ \vdots \\ \varepsilon_{N-1N} \\ \varepsilon_{NN-1} \end{pmatrix} = \begin{pmatrix} l_{12} \\ l_{21} \\ l_{13} \\ l_{31} \\ \vdots \\ l_{N-1N} \\ l_{NN-1} \end{pmatrix} = \vec{l} \tag{3.67}$$

Where A is an $N(N-1)/2 \times N$ matrix consisted of 0s and 1s. Now the least square estimation of $\vec{d}$ is

$$\hat{\vec{d}} = (A^T A)^{-1} A^T \vec{l} = (A^T A)^{-1} A^T (A\vec{d} + \vec{\varepsilon}) = \vec{d} + (A^T A)^{-1} A^T \vec{\varepsilon} = \vec{d} + \vec{v} \tag{3.68}$$

Where $\vec{v}$ is a Gaussian random vector with zero mean, autocorrelation matrix $\sigma^2 (A^T A)^{-1}$.

$$A^T A = 2 \begin{pmatrix} N-1 & 1 & \dots & 1 \\ 1 & N-1 & \dots & 1 \\ \vdots & & \ddots & \vdots \\ 1 & & \dots & N-1 \end{pmatrix} \tag{3.69}$$

$$(A^T A)^{-1} = 0.5 \begin{pmatrix} a & b & \dots & b \\ b & a & \dots & b \\ \vdots & & \ddots & \vdots \\ b & & \dots & a \end{pmatrix} \tag{3.70}$$

Where $a = \frac{2N-3}{(N-2)(2N-2)} \sim \frac{1}{N}$, $b = \frac{1}{(N-2)(2N-2)} \sim \frac{1}{2N^2}$. That is: $E(v_k^2) \sim \frac{\sigma^2}{2N}, E(v_k v_j) \sim \frac{\sigma^2}{4N^2}, j \neq k$.

3.2.2.2 Estimation of (x, y)

With the linear least square estimation of the distances from each sensor to the object, we have the following estimation of (x, y). For convenience, we assume that N is even. Then we have $N/2$ following equation pairs:

$$\begin{aligned} (x - x_{2k-1})^2 + (y - y_{2k-1})^2 &= \hat{d}_{2k-1}^2 \\ (x - x_{2k})^2 + (y - y_{2k})^2 &= \hat{d}_{2k}^2 \\ k &= 1, 2, ..., N/2 \end{aligned} \tag{3.71}$$

By subtracting the 2k-1'th equation from the 2k'th equation we get the following linear equation for x, y.

$$\begin{aligned} 2(x_{2k-1} - x_{2k})x + 2(y_{2k-1} - y_{2k})y &= \hat{d}_{2k}^2 - \hat{d}_{2k-1}^2 + y_{2k-1}^2 - y_{2k}^2 + x_{2k-1}^2 - y_{2k}^2 \\ k &= 1, 2, ..., N/2 \end{aligned} \tag{3.72}$$

So now we have a linear equation system for x, y:

$$B \begin{pmatrix} x \\ y \end{pmatrix} = \vec{c} + \vec{w} \tag{3.73}$$

Where B is an $(N/2) \times 2$ matrix, the kth row of B is $(x_{2k-1} - x_{2k}, y_{2k-1} - y_{2k})$. The kth element of $\vec{c}$ is $d_{2k}^2 - d_{2k-1}^2 + y_{2k-1}^2 - y_{2k}^2 + x_{2k-1}^2 - y_{2k}^2$. And the kth element of the noise vector $\vec{w}$ is $v_{2k}^2 - v_{2k-1}^2 + 2d_{2k}v_{2k} - 2d_{2k-1}v_{2k-1}$. The least square estimation of x, y is

$$(\hat{x}, \hat{y})^T = (B^T B)^{-1} B^T (\vec{c} + \vec{v}) = (x, y)^T + (B^T B)^{-1} B^T \vec{w} = (x, y)^T + (u_1, u_2)^T \tag{3.74}$$

3.2.2.3 Estimation Mean and Variance

First, $\vec{w}$ is zero mean, so $\vec{u} = (u_1, u_2)^T$ is zero mean as well, so the estimator is unbiased.

Secondly, we are interested in the estimation variance of the estimator.

Theorem 3.5. *Autocorrelation of $\vec{u}$ If there are N sensors uniformly distributed inside a circle centered at the object. then*

$$E(\vec{u}\vec{u}^T) = \frac{28\sigma^2}{2N^2} I_{2\times 2} \tag{3.75}$$

Proof. : First we will look at the the autocorrelation matrix of $\vec{u}$. The autocorrelation matrix of $\vec{u}$ is

$$(B^T B)^{-1} B^T E(\vec{w}\vec{w}^T) B (B^T B)^{-1} = \frac{(\frac{B^T B}{N})^{-1} B^T E(\vec{w}\vec{w}^T) B (\frac{B^T B}{N})^{-1}}{N^2} \tag{3.76}$$

Without loss of generality we assume that the positions of the sensors are iid random vectors with finite variance on the 2-D plan. Then $\frac{B^T B}{N}$ converges to

$$\begin{pmatrix} \frac{E((x_i - x_j)^2)}{2} & 0 \\ 0 & \frac{E((y_i - y_j)^2)}{2} \end{pmatrix} = \frac{E((x_i - x_j)^2)}{2} I_{2\times 2} \tag{3.77}$$

$E(\vec{w}\vec{w}^T)$ is an N/2 by N/2 matrix. Notice that the expectation of the product of odd many zero mean Gaussian random variables is zero no matter if they are independent or not. The $k, k\ th$ element is

$$\begin{aligned} & E(4d_{2k-1}^2 v_{2k-1}^2 + 4d_{2k}^2 v_{2k}^2 - 8d_{2k-1}d_{2k}v_{2k-1}v_{2k} + v_{2k-1}^4 + v_{2k}^4 - 2v_{2k-1}^2 v_{2k}^2) \\ & \approx E(4d_{2k-1}^2 v_{2k-1}^2 + 4d_{2k}^2 v_{2k}^2 - 8d_{2k-1}d_{2k}v_{2k-1}v_{2k}) \\ & \approx 2\sigma^2 (\frac{d_{2k-1}^2}{N} + \frac{d_{2k}^2}{N} - \frac{d_{2k-1}d_{2k}}{N^2}) \end{aligned} \tag{3.78}$$

Meanwhile the $i, j\ th$ $(i \neq j)$ element of that autocorrelation matrix is

$$\begin{aligned} & E(4d_{2i-1}d_{2j-1}v_{2i-1}v_{2j-1} + 4d_{2i}d_{2j}v_{2i}v_{2j} - 4d_{2i-1}d_{2j}v_{2i-1}v_{2j} - 4d_{2i}d_{2j-1}v_{2i}v_{2j-1}) \\ & \approx \frac{\sigma^2 (d_{2i} - d_{2i-1})(d_{2j} - d_{2j-1})}{N^2} \end{aligned} \tag{3.79}$$

$$
\begin{aligned}
&B^T E(\vec{w}\vec{w}^T)B \\
&\approx \frac{2\sigma^2}{N}\begin{pmatrix} \sum_{k=1}^{N/2}(x_{2k}-x_{2k-1})^2(d_{2k-1}^2+d_{2k}^2) & 0 \\ 0 & \sum_{k=1}^{N/2}(y_{2k}-y_{2k-1})^2(d_{2k-1}^2+d_{2k}^2) \end{pmatrix} \\
&\approx \sigma^2 E((x_i-x_j)^2(d_i^2+d_j^2))I_{2\times 2} \qquad (3.80)
\end{aligned}
$$

Combining with Eqn.3.77.The autocorrelation matrix of $\vec{u}$ is

$$\frac{4\sigma^2 E((x_i-x_j)^2(d_i^2+d_j^2))}{N^2 E((x_i-x_j)^2)^2} I2\times 2 \qquad (3.81)$$

Suppose transmitters all have some power limit, then all of the transmitters which contributes to a multi-path distance measurement are within a circle,radius R, center at the object. If we assume that the sensors are uniformly distributed, then we can calculate $\frac{E((x_i-x_j)^2(d_i^2+d_j^2))}{E((x_i-x_j)^2)^2}$. Without loss of generality, we assume the object is at $(0,0)$, then (x_i, y_i) is uniformly distributed in the region $\{(x,y) : x^2+y^2 \le R^2\}$. Then

$$E((x_i-y_i)^2) = E(x_i^2+y_i^2) = \frac{\int_0^{2\pi}\int_0^R r^2 r dr d\theta}{\pi R^2} = \frac{R^2}{2} \qquad (3.82)$$

$$
\begin{aligned}
&E((x_i-x_j)^2(d_i^2+d_j^2)) = E((x_i-x_j)^2(x_i^2+y_i^2+x_j^2+y_j^2)) = E((x_i^2+y_i^2)^2) \\
&+E(x_i^2+y_i^2)^2 = \frac{\int_0^{2\pi}\int_0^R r^4 r dr d\theta}{\pi R^2} + (\frac{\int_0^{2\pi}\int_0^R r^2 r dr d\theta}{\pi R^2})^2 = \frac{R^4}{3}+\frac{R^4}{4} = \frac{7R^4}{12} \qquad (3.83)
\end{aligned}
$$

Now we have the autocorrelation matrix of $\vec{u}$ as following.

$$\boxed{\frac{28\sigma^2}{3N^2} I_{2\times 2}} \qquad (3.84)$$

□

3.2.2.4 Simulation Results for MSSO

In our simulation, each sensor can both send and receive signals. Thus we have $N(N-1)$ multi-path length measurements given N sensors in the sensor network. The result is shown in Fig.3.12. In our simulation, the position of the sensors and the objects are uniformly distributed in a 1×1 region. The multi-path length measurements

are corrupted with iid Gaussian noise with distribution $\sim N(0, 10^{-4})$ so the standard deviation is 0.01. The bar plot in Fig.3.12 is $\sqrt{\frac{\sum_{i=1}^{W}(x_i-x_i^e)^2+(y_i-y_i^e)^2}{2W}}$, an average of the variance of estimation errors over $W = 100$ random simulations. Where (x_i, y_i) is the position of the object in the i'th simulation, (x_i^e, y_i^e) is the estimation of the position using our algorithm. As can be seen, the square root of the estimation variances decays to 0 at rate $\frac{1}{N}$ which is what we predicated. The solid curve is $\sqrt{\frac{28\sigma^2}{3N^2}}$.

As shown in Fig.3.13, the mean error of the estimations is rather small comparing to the square root of the variance of the multi-path distance errors.

3.2.2.5 Discussions on Multiple-Sensor, Single Object Tracking

The proposed scheme achieve the estimation variance $\sim O(N^{-2})$ given that every pair of sensor can communicate with each other, where N is the number of sensors. Comparing to the more general Cramer-Rao bound we derived $V(x) = V(y) \sim O(\frac{1}{NM})$, where N is the number of transmitters, M is the number of receivers. If each sensor can both transmit and receive signals as assumed in our scheme. Then $M = N$, thus the Cramer-Rao bound is $V(x) = V(y) \sim O(\frac{1}{N^2})$. So our algorithm is order optimal. I.e the estimation variance and the Cramer-Rao bound approach to 0 at the same rate asymptotically as the number of sensors goes to infinity.

The above analysis are based on the assumption that every sensor transmits at the constant power level no matter how many sensors are in the field. A natural question to ask is what if the total transmission power remains constant instead of increasing proportionally with the number of sensors? What if the total received power remains constant? With the assumption that the variance of the AWGN added to the estimation of multiple-path lengths is proportional to the inverse of the power of the transmitter, we can answer those questions.

When the total transmission power is constant, the power of each single transmission is scaled down by N. Then the variance of the AWGN added to the measurement

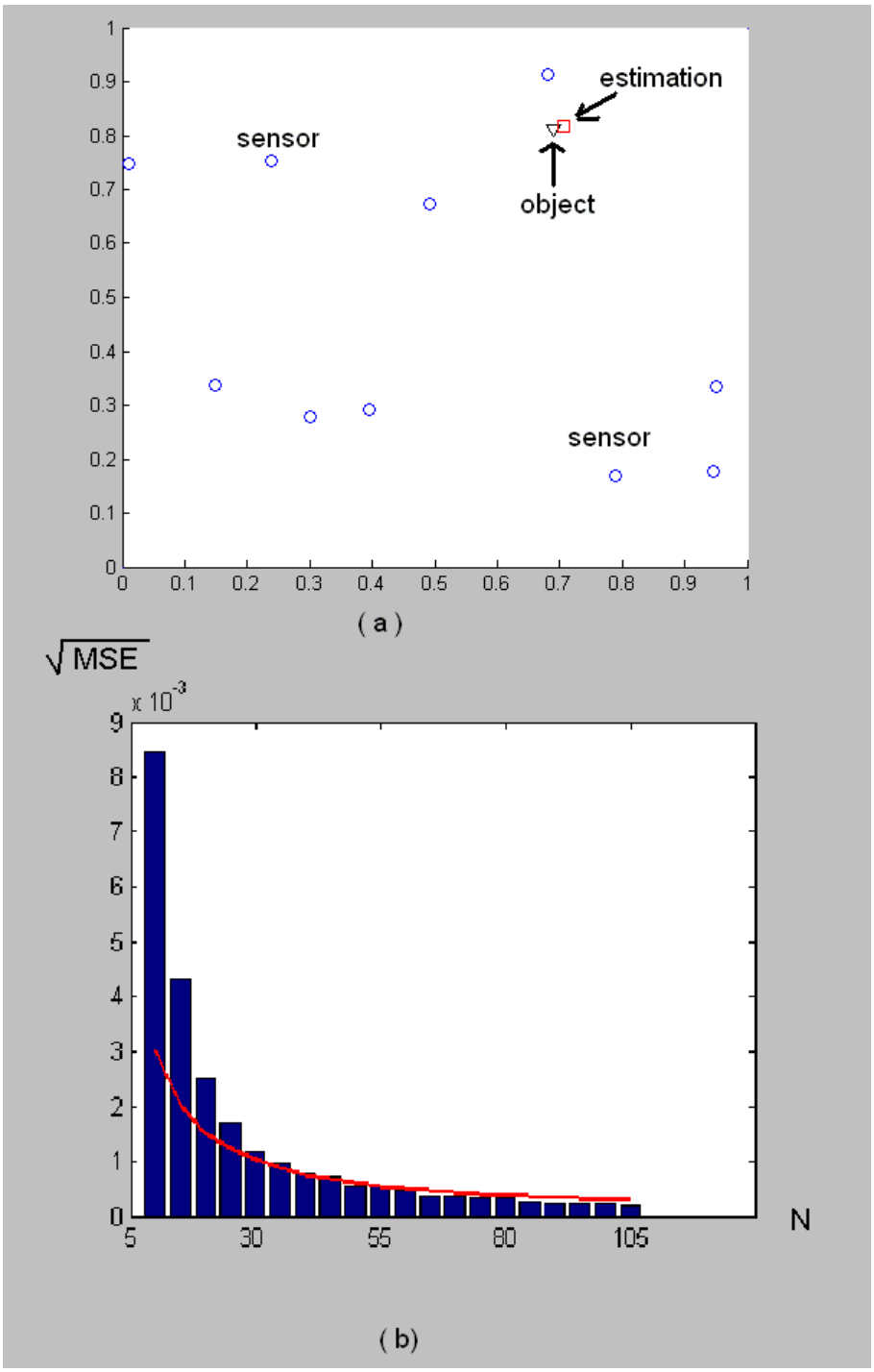


Figure 3.12: (a) is the setup of the simulations, the position of the sensors and the objects are uniformly distributed in a 1×1 region. The red square is the estimation of the object. Here we have 10 sensors.
(b) is the square root of the mean error square VS. the number of sensors N. The bar plot is the error, while the solid line is the $\sqrt{\frac{28\sigma^2}{3N^2}}$ curve.

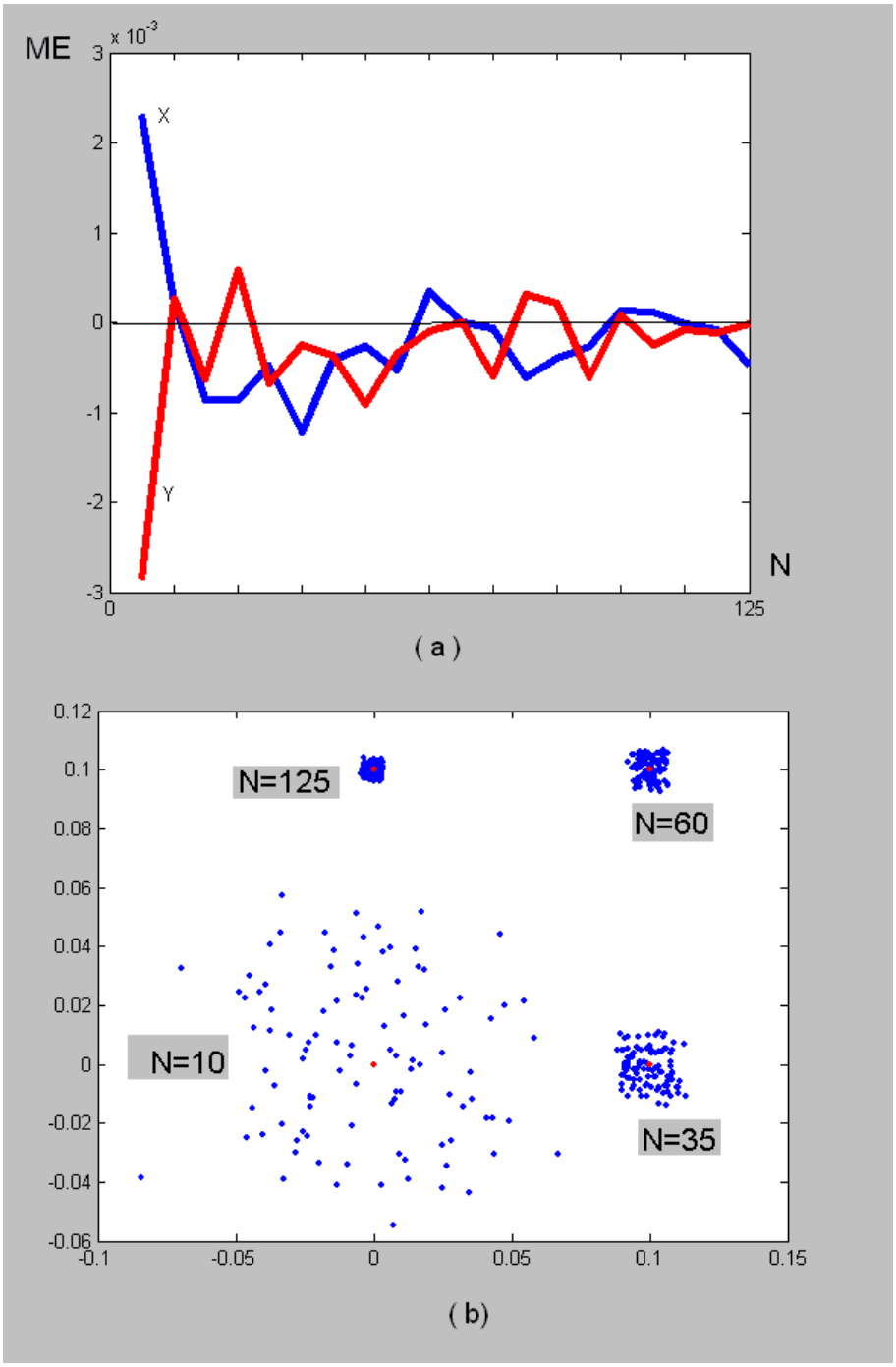


Figure 3.13: (a) is the mean error of the estimation
(b) is estimation results of 100 independent simulations. The red dot is the position of the object, the blue dots are the estimations

of each single multiple-path distance is $\sigma^2 N$. Using the same algorithm, the position estimation is still unbiased with variance, scaled up with factor N, $\sim O(N^{-1})$.

If the total received power is constant, then each single transmission power is scaled down by factor N^{-2}. It is easy to figure out that the estimation variance is $\sim O(1)$ using the same algorithm. Increasing the number of sensors does not help the estimation in this scenario. This suggests that we cannot achieve a better estimation accuracy by only adding more sensors. The total received power must be increased to achieve a better estimation accuracy.

3.2.3 MSSO in a sensor network with a single receiver

In this section we are going to study the tracking problem for a multiple transmitter, single receiver sensor network. Suppose that there are N transmitters, and only one receiver in a sensor network. If N is large, then we can track the object without motion model, if N is small, we need the motion model again to be able to track the object. Object tracking in a single receiver network is interesting because it allows the receivers to distributedly estimate the object position.

3.2.3.1 Cramer-Rao bound analysis

We derive the Cramer-Rao bound on the estimation problem for 2 different scenarios. First, if there are a large amount of transmitters, we can accurately estimate the position of the object (x, y) without any motion model, the analysis is based on the law of large numbers. Second, if there are only several transmitters. In this case we analyze the Cramer-Rao bound for both tracking without motion model and tracking with constant velocity model.

Case 1: N is large, where no motion model is needed.

The analysis for multiple transmitter multiple receiver network still holds. In stead the total receiver number M is 1 now. $J_{2\times 2}$ is the Fisher information matrix. Then

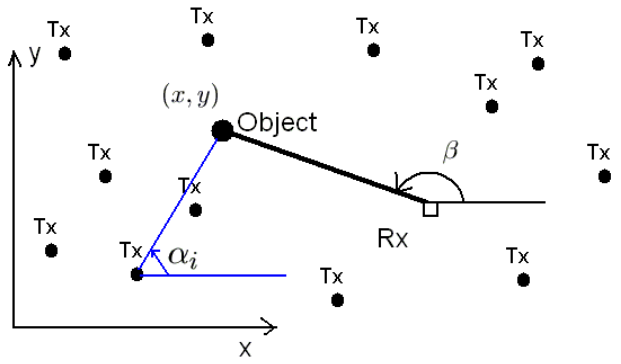


Figure 3.14: Multi-transmitter, single receiver, N is large

the Cramer- Rao bound for the estimation of x and y are $V(x)$ and $V(y)$ respectively.

Theorem 3.6. *Asymptotic CR bound for single receiver case: In an N transmitter, single receiver sensor network, assume that the transmitters are angularly uniformly distributed around (x, y). Then*

$$lim_{N\to\infty} N(V(x) + V(y)) = \frac{8\sigma^2}{3} \quad \textit{in probability} \tag{3.85}$$

Proof. : The proof is basically the same as the shown in Theorem where the number of receivers is also big. As show in Fig.3.14.

$$\begin{aligned} J_{11} &= \frac{1}{\sigma^2}\sum_{k=1}^{N}(cos(\beta) + cos(\alpha_k))^2 \\ J_{22} &= \frac{1}{\sigma^2}\sum_{k=1}^{N}(sin(\beta) + sin(\alpha_k))^2 \\ J_{12} &= \frac{1}{\sigma^2}\sum_{k=1}^{N}(cos(\beta) + cos(\alpha_k))(sin(\beta) + sin(\alpha_k)) \end{aligned} \tag{3.86}$$

$V(x)+V(y) = \frac{J_{11}+J_{22}}{J_{11}J_{22}-J_{12}^2}$. We assume the angle of $(T_x-(x, y))$ is uniformly distributed in $[0, 2\pi)$. Then we have the estimation variance of the position : $V(x)+V(y) = \frac{J_{11}+J_{22}}{J_{11}J_{22}-J_{12}^2}$.

Notice that the variance goes to zero by the law of large numbers.

$$lim_{N\to\infty}\frac{J_{11}}{N} = cos^2(\beta) + \frac{1}{2} \text{ in probability}$$
$$lim_{N\to\infty}\frac{J_{22}}{N} = sin^2(\beta) + \frac{1}{2} \text{ in probability}$$
$$lim_{N\to\infty}\frac{J_{12}}{N} = cos(\beta)sin(\beta) \text{ in probability}$$

so we have :

$$lim_{N\to\infty}N\frac{J_{11}+J_{22}}{J_{11}J_{22}-J_{12}^2} = \frac{cos^2(\beta)+\frac{1}{2}+sin^2(\beta)+\frac{1}{2}}{(cos^2(\beta)+\frac{1}{2})(sin^2(\beta)+\frac{1}{2})-sin^2(\beta)cos^2(\beta)}$$
$$= \frac{8}{3} \text{ in probability} \tag{3.87}$$

□

So we can claim that

$$\boxed{V(x)+V(y) \approx \frac{8\sigma^2}{3N}} \tag{3.88}$$

The order of the estimation variance is $\frac{1}{N}$ which is consistent with the order of the total received power. In Fig.3.15, we show the theoretical Cramer-Rao bounds vs. simulation results (Monte-Carlo method)in logarithm terms. In our simulation, we fix the object at $[0, 0]$, N transmitters are uniformly distributed in $[-1, 1] \times [1, 1]$, for each N we compute $\frac{V(x)+V(y)}{\sigma^2}$ for 100 times and get the average.

Case 2: N is small (especially 2), to directly calculate the position of the object based on the intersection of ellipses is not stable as shown in Fig. 3.21. Thus we assume the object is moving with a constant velocity. Then the argument we used in the previous analysis is no longer right, since the angles α_i 's cannot be assumed to be uniformly distributed in $[0, 2\pi]$. Thus the estimation problem here is similar to the problem in section 3.1 . And the Cramer-Rao bound analysis is similar to that in section 3.1.2.

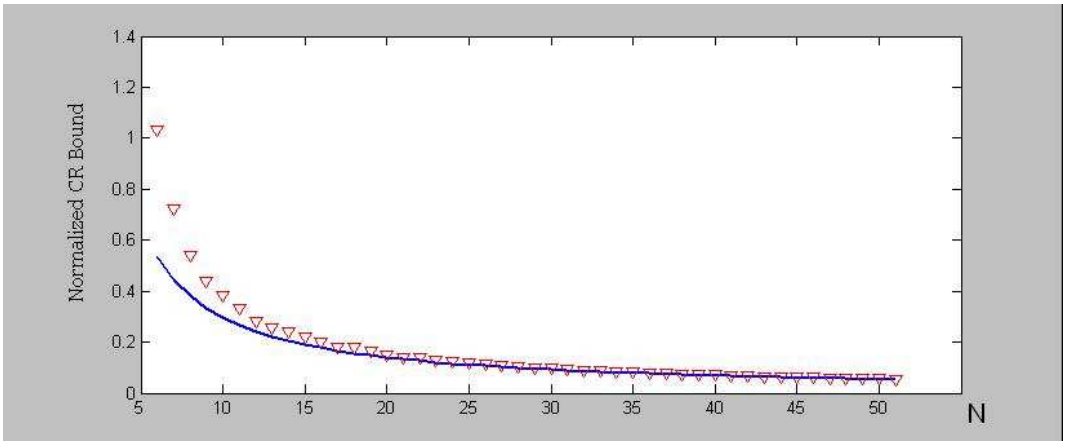


Figure 3.15: Blue curve is the theoretical CR bound $\frac{8}{3N}$, red triangles are simulation results

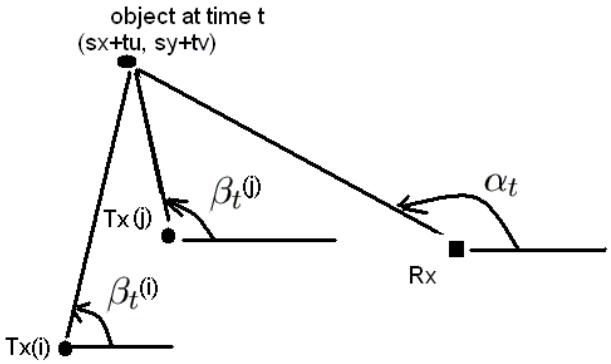


Figure 3.16: Multi-transmitter, single receiver, N is small
$(sx + tu, sy + tv)$ is the position of the object at time $t, s = 1 - t$.

Similar to section 3.1.2, we suppose that the measuring starts at time 0 when the object is at (x, y), ends at time 1 when the object is at (u, v). Then at time t the object is at $(sx + tu, sy + tv)$, where $s = 1 - t$. Then we have the following W measurements:

$$d_t(i) = \sqrt{(sx + tu - a_i)^2 + (sy + tv - b_i)^2} + \sqrt{(sx + tu - a)^2 + (sy + tv - b)^2} + \delta_t(i)$$
$$i = 1, 2, ..N, t = 0, 1/(W-1), 2/(W-1), ...(W-2)/(W-1), 1 \quad (3.89)$$

Where $d_t(i)$ is the multi-path length measurement from transmitter i with position (a_i, b_i) to the receiver with position (a, b) at time t. Similarly to the theorem in the Section 3.1.2, we have the following theorem.

Theorem 3.7. *Fisher Information Matrix of MTSR object tracking with motion model For a specific motion which starts from (x, y), ends at (u, v). If there are W multipath distance measurements for each transmitter. Let x, y, u, v be the 1st, 2nd, 3rd and 4th parameter to be estimated respectively. Let $J_{4\times4}$ be the Fisher Information Matrix for x, y, u, v.*

$$lim_{W\to\infty}\frac{\sigma^2 J}{W} = K(x, y, u, v) \quad (3.90)$$

Where $K(x, y, u, v)$ is a function of (x, y, u, v).

Proof. : Following the same procedure in the Section 3.1.2. We have

$$\begin{aligned} J_{11} &= \frac{1}{\sigma^2}\sum_{i=1}^{N}\sum_t\left(\frac{s(sx + tu - a_i)}{\sqrt{(sx + tu - a_i)^2 + (sy + tv - b_i)^2}} + \frac{s(sx + tu - a)}{\sqrt{(sx + tu - a)^2 + (sy + tv - b)^2}}\right)^2 \\ &= \frac{1}{\sigma^2}\sum_{i=1}^{N}\sum_t(cos(\beta_t(i)) + cos(\alpha_t))^2 s^2 \end{aligned} \quad (3.91)$$

Similarly we have the expression of J_{ii}:

$$J_{22} = \sum_{i=1}^{N}\frac{1}{\sigma^2}\sum_t(sin(\alpha_t) + sin(\beta_t(i)))^2 s^2 \quad (3.92)$$

$$J_{33} = \sum_{i=1}^{N}\frac{1}{\sigma^2}\sum_t(cos(\alpha_t) + cos(\beta_t(i)))^2 t^2 \quad (3.93)$$

$$J_{44} = \sum_{i=1}^{N}\frac{1}{\sigma^2}\sum_t(sin(\alpha_t) + sin(\beta_t(i)))^2 t^2 \quad (3.94)$$

J_{ij}'s:

$$J_{12} = J_{21} = \sum_{i=1}^{N} \frac{1}{\sigma^2} \sum_t (sin(\alpha_t) + sin(\beta_t(i)))(cos(\alpha_t) + cos(\beta_t(i)))s^2 \quad (3.95)$$

$$J_{13} = J_{31} = \sum_{i=1}^{N} \frac{1}{\sigma^2} \sum_t (cos(\alpha_t) + cos(\beta_t(i)))^2 st \quad (3.96)$$

$$J_{14} = J_{41} = \sum_{i=1}^{N} \frac{1}{\sigma^2} \sum_t (sin(\alpha_t) + sin(\beta_t(i)))(cos(\alpha_t) + cos(\beta_t(i)))st \quad (3.97)$$

$$J_{23} = J_{32} = \sum_{i=1}^{N} \frac{1}{\sigma^2} \sum_t (sin(\alpha_t) + sin(\beta_t(i)))(cos(\alpha_t) + cos(\beta_t(i)))st \quad (3.98)$$

$$J_{24} = J_{42} = \sum_{i=1}^{N} \frac{1}{\sigma^2} \sum_t (sin(\alpha_t) + sin(\beta_t(i)))^2 st \quad (3.99)$$

$$J_{34} = J_{43} = \sum_{i=1}^{N} \frac{1}{\sigma^2} \sum_t (sin(\alpha_t) + sin(\beta_t(i)))(cos(\alpha_t) + cos(\beta_t(i)))t^2 \quad (3.100)$$

Where $\alpha_t, \beta_t(i) \in [0, 2\pi), t = 0, 1/(W-1), 2/(W-1), ...(W-2)/(W-1), 1$ and

$$\begin{aligned}
cos(\alpha_t) &= \frac{sx + tu - a}{\sqrt{(sx + tu - a)^2 + (sy + tv - b)^2}}; \\
sin(\alpha_t) &= \frac{sy + tv - a}{\sqrt{(sx + tu - a)^2 + (sy + tv - b)^2}}; \\
cos(\beta_t(i)) &= \frac{sx + tu - a_i}{\sqrt{(sx + tu - a_i)^2 + (sy + tv - b_i)^2}}; \\
sin(\beta_t(i)) &= \frac{sy + tv - b_i}{\sqrt{(sx + tu - a_i)^2 + (sy + tv - b_i)^2}};
\end{aligned} \quad (3.101)$$

The geometric interpretation of $\alpha_t, \beta_t(i)$ is illustrated in Fig.3.16.

Notice that $(cos(\alpha_t) + cos(\beta_t(i)))^2(1-t)^2$ is a bounded continuous function on $t \in [0, 1]$, for all i's. If W is sufficiently large, we have the following result:

$$J_{11} \simeq \sum_{i=1}^{N} \frac{W}{\sigma^2} \int_0^1 (cos(\alpha_t) + cos(\beta_t(i)))^2 s^2 dt \quad (3.102)$$

Similar for J_{ij}'s. The Fisher information matrix J have the following form when W is sufficiently large (when the sum converges to the integral).

$$J = \frac{W}{\sigma^2} K(x, y, u, v) \quad (3.103)$$

$$K_{11} = \sum_{i=1}^{N} \int_0^1 (cos(\alpha_t) + cos(\beta_t(i)))^2 s^2 dt$$
$$K_{12} = J_{21} = \sum_{i=1}^{N} \int_0^1 (sin(\alpha_t) + sin(\beta_t(i)))(cos(\alpha_t) + cos(\beta_t(i)))s^2 dt \quad (3.104)$$

Similarly for other entries of K. □

Now we can calculate the Cramer-Rao bound on the motion estimation.

$$V(x) = \frac{\sigma^2}{N} {K^{-1}}_{11}; V(y) = \frac{\sigma^2}{N} {K^{-1}}_{22};$$
$$V(u) = \frac{\sigma^2}{N} {K^{-1}}_{33}; V(v) = \frac{\sigma^2}{N} {K^{-1}}_{44}; \quad (3.105)$$

We can numerically compute $K(x, y, u, v)$ and have an idea of how many multi-path length measurements at least we need to achieve a certain estimation variance ξ^2 for any of x, y, u, v.

$$W \geq \frac{\sigma^2}{\xi^2} max_i K_{ii}^{-1} \quad (3.106)$$

In all the following numerical calculations of Cramer-Rao bounds, we normalize the received power of the receiver. Or, equivalently, all the Cramer-Rao bound are multiplied by a N, where N is the number of transmitters.

Some numerical results

Now we give some plots of the Cramer-Rao bounds given different transmitter number N. In our experiments, we normalize the received power, i.e. for an N transmitter sensor network, we assume the multi-path distance estimation variance is $N\sigma^2$.

In simulation (1), the setup of the sensors is illustrated in Fig.3.17. Instead of 1 multi-path length measurement at a time, the receiver gets 2 multi-path length measurements from 2 transmitters of different positions. Without motion model, ${K^{-1}}_{11} + {K^{-1}}_{22} = \frac{1}{\sigma^2}({J^{-1}}_{11} + {J^{-1}}_{22})$ is shown in Fig.3.21, where J is defined in Eqn.3.50.$K = J\sigma^2$ is intrinsic to the estimation problem without motion model. With motion model, $max_i {J^{-1}}_{ii}$ are shown in Fig.3.22, where J is defined in Eqn.3.104. As can be seen, the additional transmitter dramatically increases the ability of the network to track

the motion of the object. However, the Cramer-Rao bound is still big with or without motion model (notice the Cramer-Rao bound plot is in logarithm scale) which means the estimation accuracy is still bad. Thus 2 transmitter, single receiver network is not capable of accruately tracking object.

In simulation (2), setup as in Fig.3.18, we have 3 transmitters uniformly placed on the circle with radius 2 around the receiver. As can be seen in Fig.3.23 and Fig.3.24, the Cramer-Rao bounds are smaller for both motion model and no motion model cases.

In simulation (3), setup as in Fig.3.19, we have $N = 4, 6, 10, 30$ transmitters uniformly placed on the circle with radius 2 around the receiver.

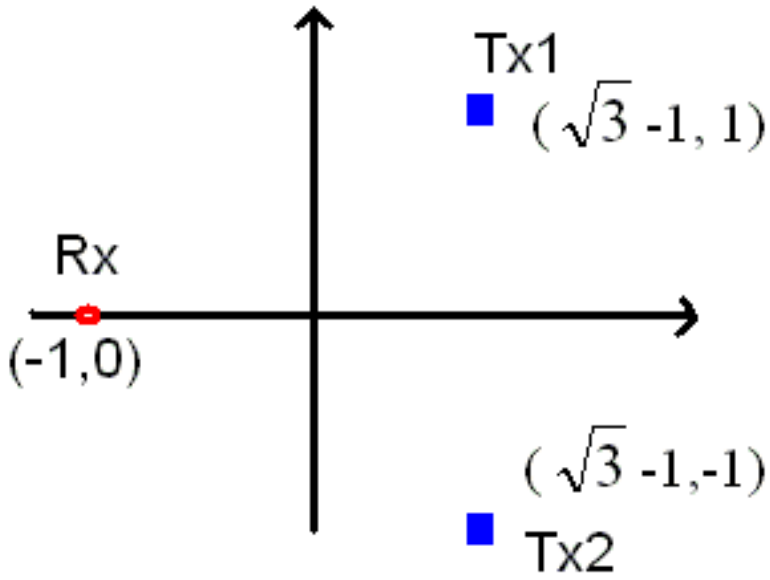


Figure 3.17: A 2 transmitter 1 receiver network, receiver is at $(-1, 0)$, transmitters are at $(\sqrt{3} - 1, 1), (\sqrt{3} - 1, -1)$

And finally we plot the Cramer-Rao bound for some points in a sensor network of different number N of transmitters. The setup of the transmitters, receiver is still as in Fig.3.19. The points are as shown in Fig.3.20. There are two sets of points, set 1 is $\{(x_k, y_k) : x_k = 0.2kcos(0.6458), y_k = 0.2ksin(0.6458)\}, k = 0, 1, ...30$, set 2 is $\{(x_k, y_k) : x_k = 0.2k, y_k = 0\}, k = 0, 1, ...30$. The distances from the points to the origin

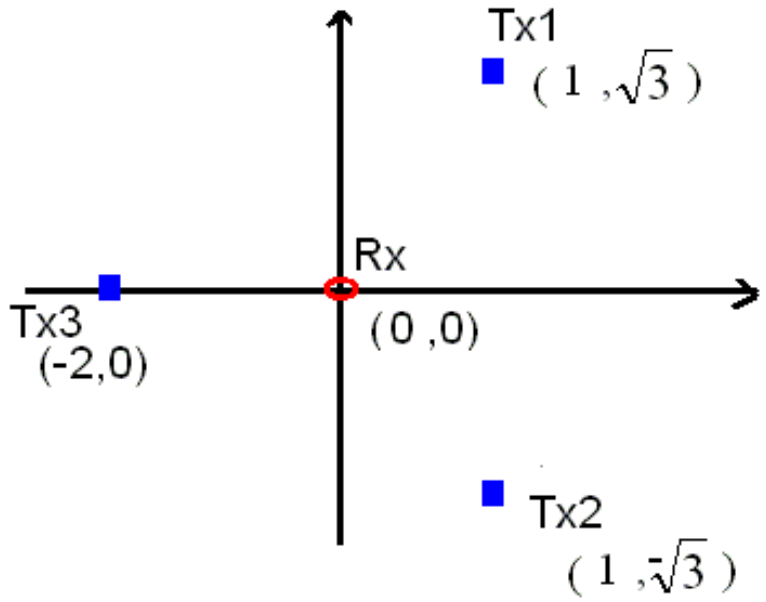


Figure 3.18: A 3 transmitter 1 receiver network, receiver is at $(0, 0)$, transmitters are uniformly on the circle with radius 2

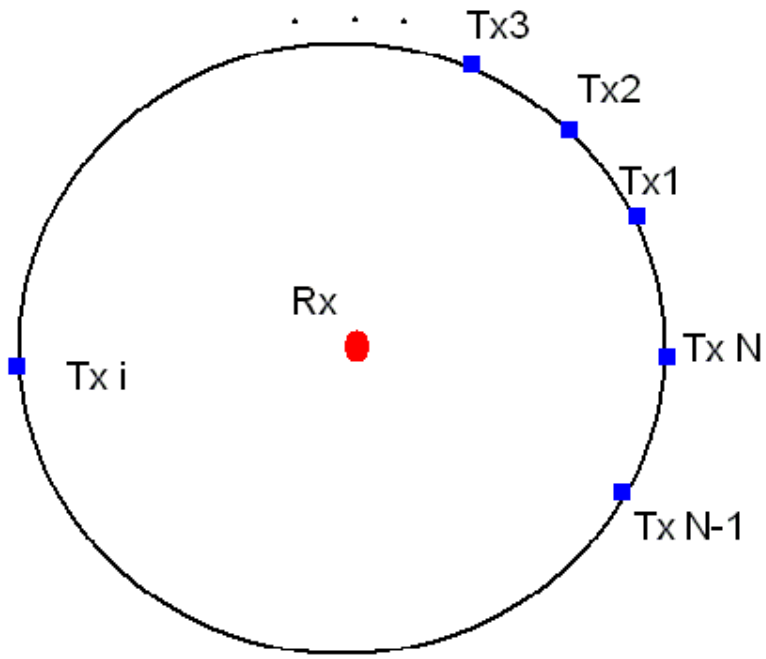


Figure 3.19: An N transmitter 1 receiver network, receiver is at $(0, 0)$, transmitters are uniformly on the circle with radius 2

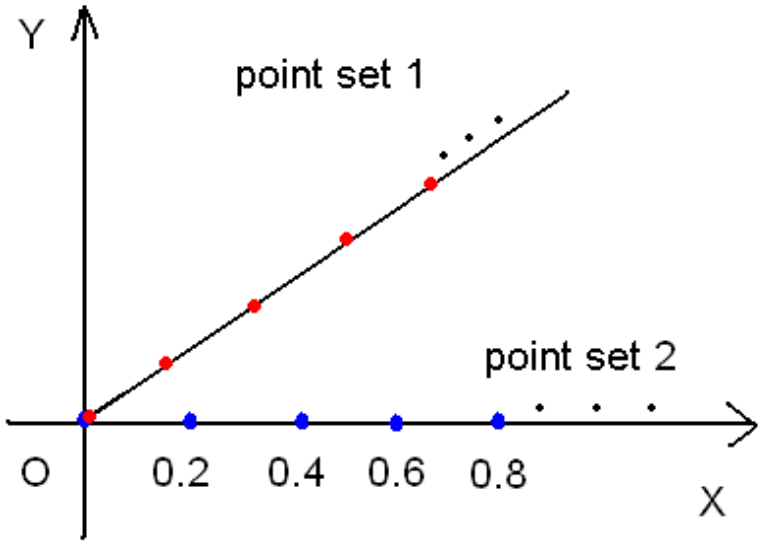


Figure 3.20: Set 2 are those blue points on the x-axis, 0.2 apart from 0 to 6. Set 1 are red points on the $y = 0.7536x$ line, 0.2 apart from origin to 6 away from origin.

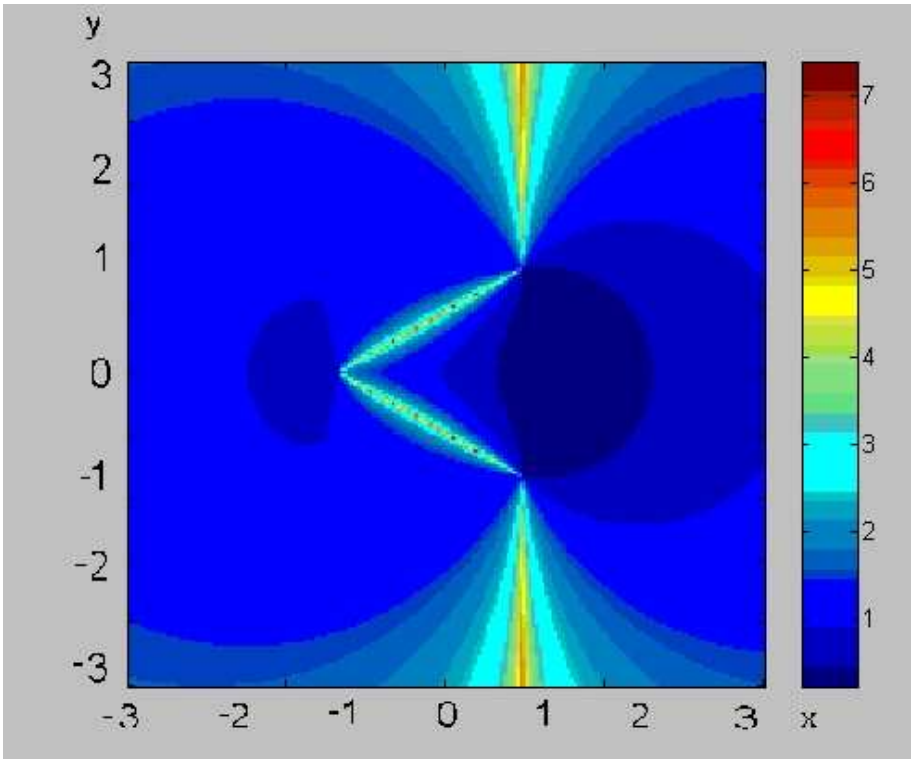


Figure 3.21: 2 Tx, The value on (x, y) indicates $log_{10}(\frac{2}{\sigma^2}({J^{-1}}_{11} + {J^{-1}}_{22}))$, no motion model

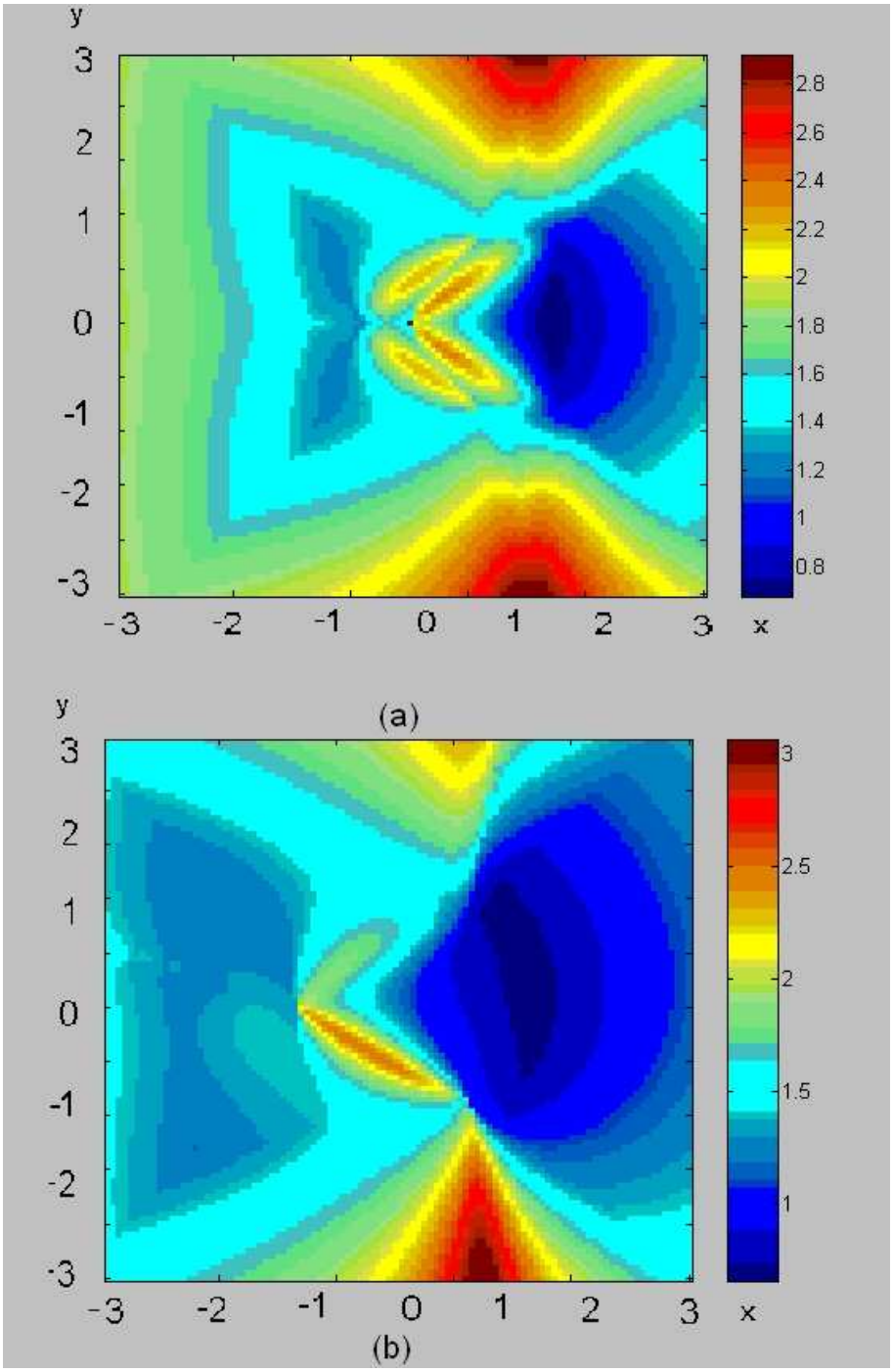


Figure 3.22: 2 Tx,The value on (x, y) indicates $log_{10}(2max_i {J^{-1}}_{ii})$, (a) is of the 4-tuple $(x, y, x-1, y)$. i.e. the object moves 1 leftward along the x-axis. (b) is of the 4-tuple $(x, y, 0.5, -0.5)$. i.e. the motion of the object ends at $(0.5, -0.5)$.

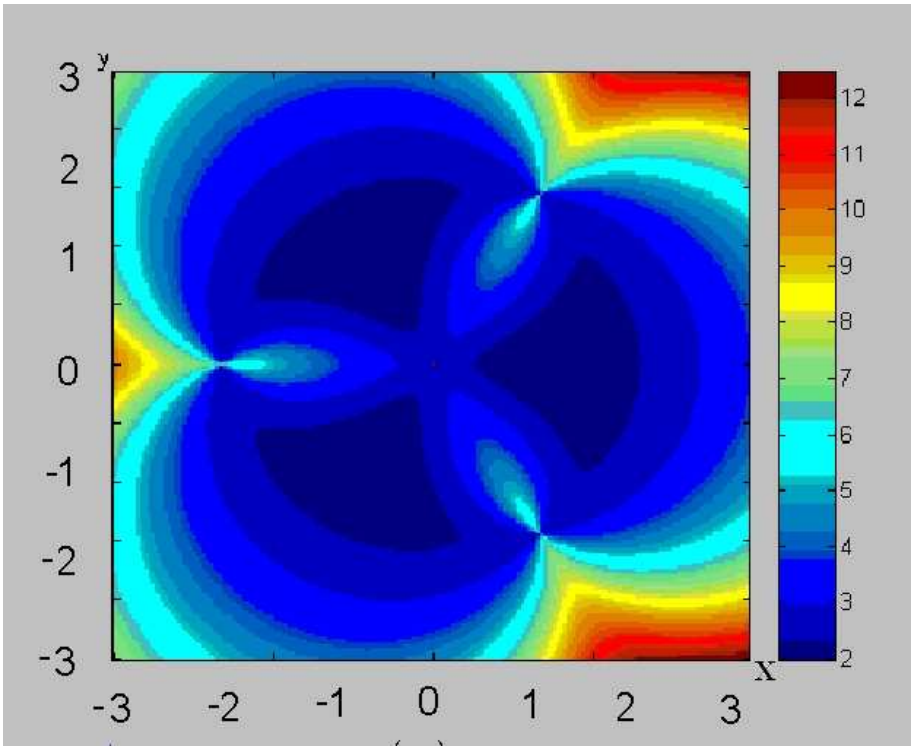


Figure 3.23: 3 Tx, The value on (x, y) indicates $3(\frac{1}{\sigma^2}(J^{-1}{}_{11} + J^{-1}{}_{22}))$, no motion model

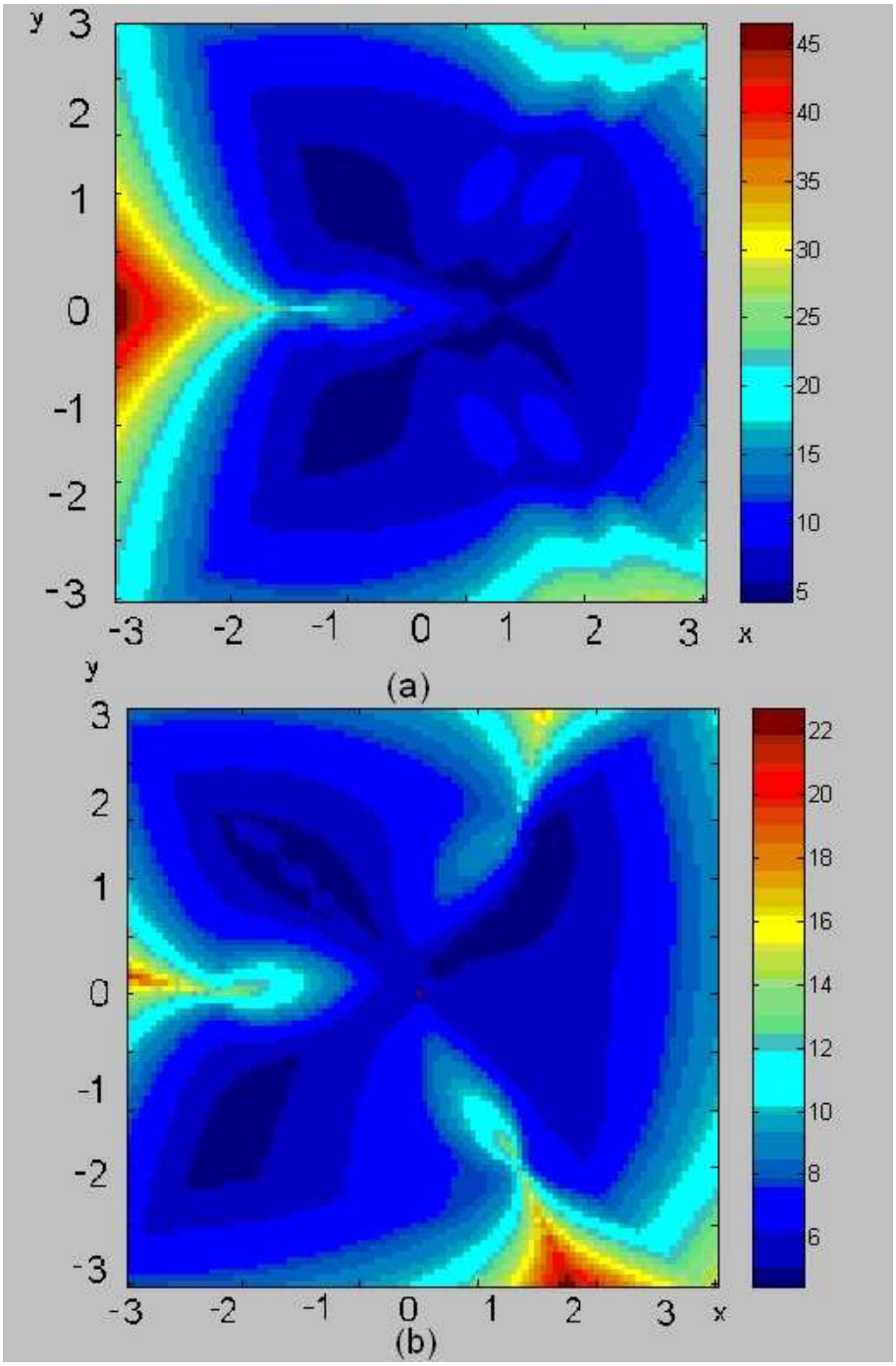


Figure 3.24: 3 Tx, The value on (x, y) indicates $3(max_i {J^{-1}}_{ii})$, (a) is of the 4-tuple $(x, y, x-1, y)$. i.e. the object moves 1 leftward along the x-axis. (b) is of the 4-tuple $(x, y, 0.5, -0.5)$. i.e. the motion of the object ends at $(0.5, -0.5)$.

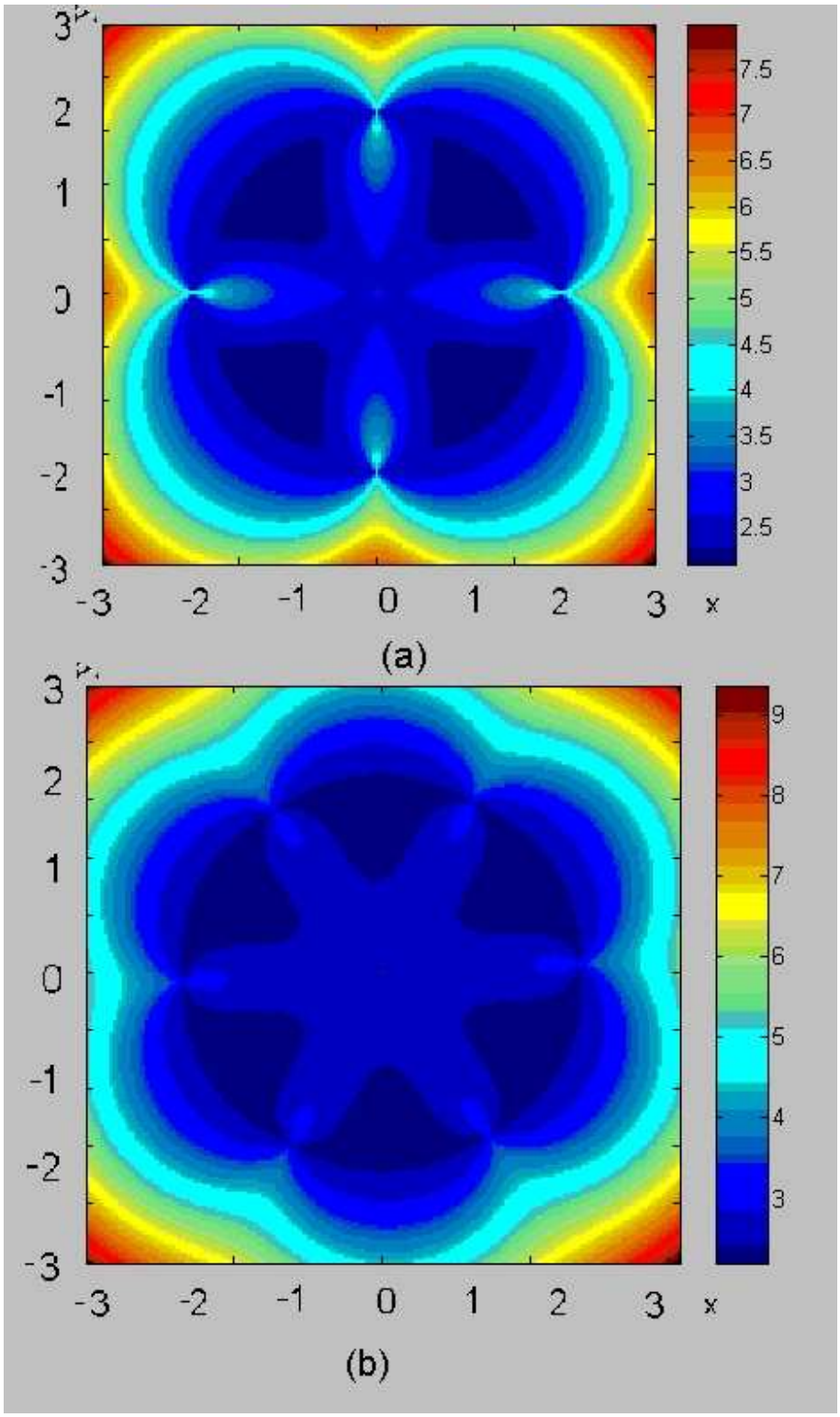


Figure 3.25: The value on (x, y) indicates $N(\frac{1}{\sigma^2}({J^{-1}}_{11} + {J^{-1}}_{22}))$, where $N = 4$ in (a), $N = 6$ in (b), no motion model

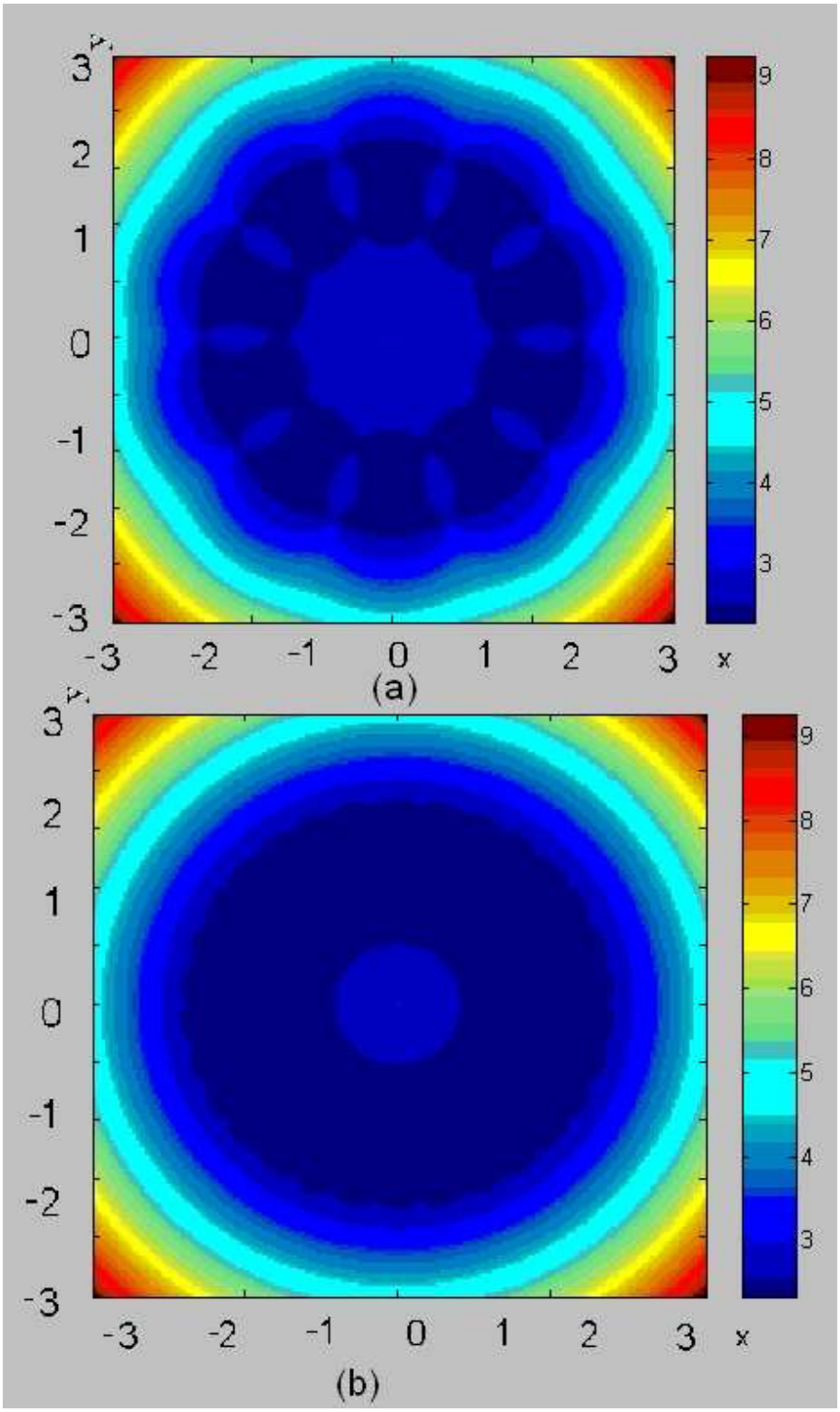


Figure 3.26: The value on (x, y) indicates $N(\frac{1}{\sigma^2}({J^{-1}}_{11} + {J^{-1}}_{22}))$, where $N = 10$ in (a), $N = 30$ in (b), no motion model

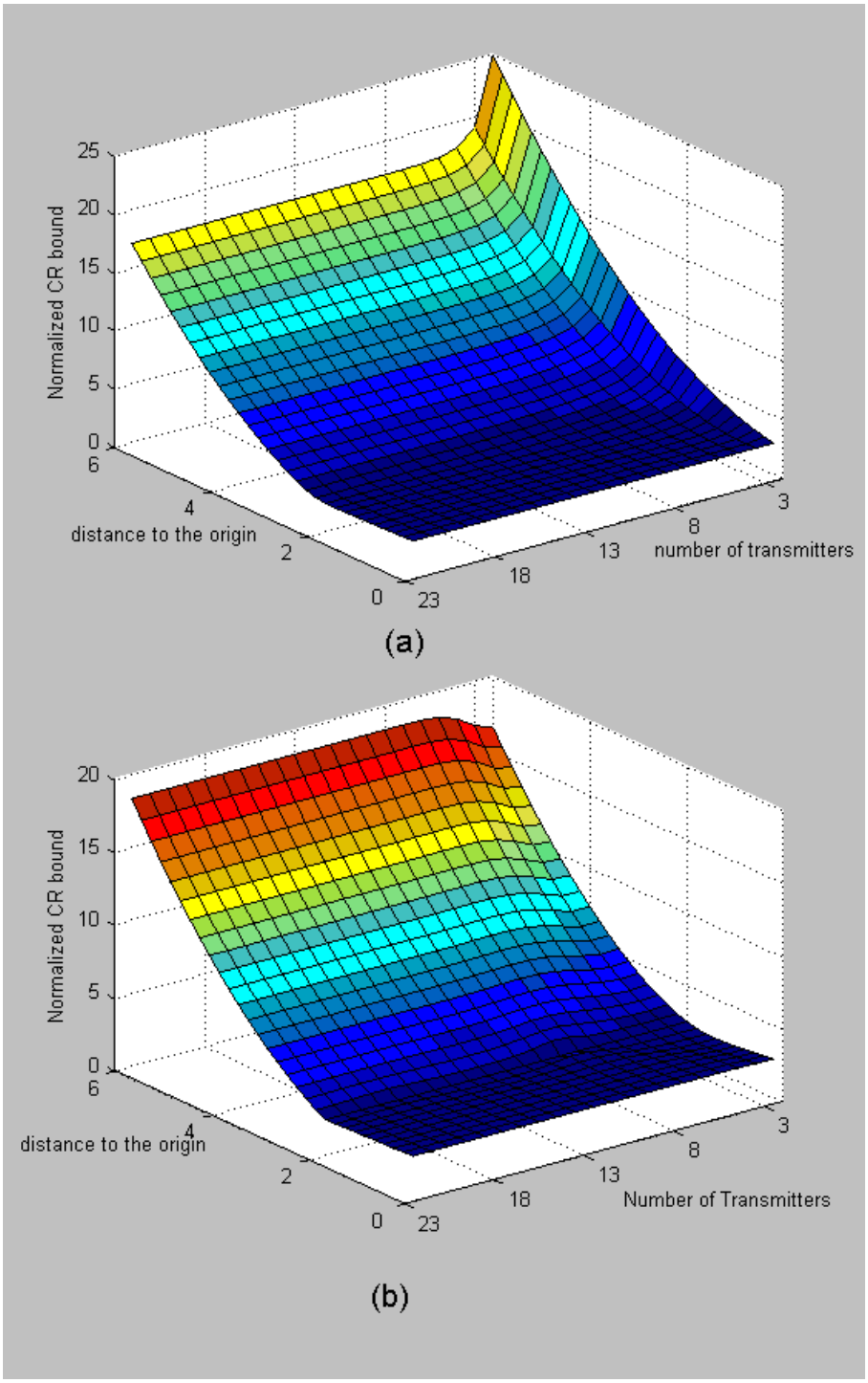


Figure 3.27: $N(\frac{1}{\sigma^2}({J^{-1}}_{11} + {J^{-1}}_{22}))$,
(a)Normalized CR bound for point set 2
(b)Normalized CR bound for point set 1

is in the range of $[0, 6]$.

As shown in Fig.3.27, we plot the normalized Cramer-Rao bound for each points given number of transmitters. The normalized Cramer-Rao bound is solely dependent on the geometry of the sensor network and the position of the object. It depicts the lower estimation bound given normalized received power. We make some observations here:

(1) The normalized Cramer-Rao bound $N(\frac{1}{\sigma^2}({J^{-1}}_{11} + {J^{-1}}_{22}))$ converges to a constant pretty fast as the number of the transmitters increases for all the points.

(2) The normalized Cramer-Rao bound converges to $\frac{8}{3}$ as we predicted when the point is inside the radius 2 circle. While it converges to a larger value, for those points outside the circle, since the assumption that those transmitters are uniformly distributed angularly fails.

(3)The normalized Cramer-Rao bound increases as the point moves away from the origin. In the next section we will explore the asymptotical property of the Cramer-Rao bound when the object moves to infinitely faraway.

3.2.4 Cramer-Rao bound for the outside region

As stated in the third observation above, if the object is outside the sensor field, the normalized Cramer-Rao bound increases as the distance between the object and the sensors increases. In this section, we explore the relation between the distance and the normalized Cramer-Rao bound.

3.2.4.1 CR bound in Euclidean metric space

First we explore the estimation variance in 2D Euclidean space,
$E((x - \hat{x})^2 + (y - \hat{y})^2)$. The geometry of the problem is illustrated in Fig. 3.28. We explore the more general problem of multiple transmitter and multiple receiver case. Suppose N transmitters at $(x_i, y_i), i = 1, 2, ...N$ and M receivers $(x'_j, y'_j), j = 1, 2, ...M$

are uniformly located inside a circle of radius r,centered in the origin. The distance from the object to the origin is L. Since we are interested in the Cramer-Rao bound of the position estimation $V(x) + V(y)$, without loss of generality, we assume the object is on the x-axis, i.e. $(x, y) = (L, 0)$.

Theorem 3.8. *CR bound $V(x) + V(y)$ for faraway region If the N transmitters and M receivers are uniformly distributed inside a circle with radius r. And the distance between the center of the sensor field and the object is L, s.t. $L \gg r$ Then*

$$lim_{N\to\infty,M\to\infty} MN(V(x) + V(y)) = (\frac{2L^2\sigma^2}{r^2}) \quad \textit{in probability} \tag{3.107}$$

Proof. : We first make some observations.

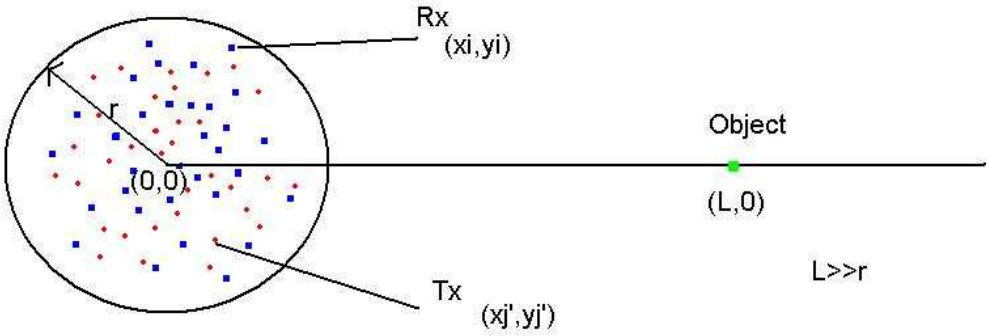


Figure 3.28: The object is far away from the sensor network

If the transmitter i at (x_i, y_i) are uniformly distributed inside the circular region with radius r, centered at the origin $O = (0, 0)$. Then $E(x_i) = E(y_i) = 0$, and:

$$E(x_i^2) = E(y_i^2) = \frac{\int_0^{2\pi}\int_0^r (\rho cos(\theta))^2 \rho d\rho d\theta}{\pi r^2} = \frac{r^2}{4} \tag{3.108}$$

Obviously x'_j, y'_j have the same mean and variance. Noticing that $L \gg r > x_i, y_i, x'_j, y'_j$. We have :

$$\frac{L - x_i}{\sqrt{(L - x_i)^2 + y_i^2}} \approx \frac{1}{\sqrt{1 + \frac{y_i^2}{(L-x_i)^2+y_i^2}}} \approx 1 \tag{3.109}$$

Similarly

$$\frac{y_i}{\sqrt{(L-x_i)^2+y_i^2}} \approx \frac{y_i}{L} \tag{3.110}$$

As shown in Eqn.3.50, following the standard process we have the Fisher information matrix $J_{2\times 2}$

$$\begin{aligned} J_{11} &= \frac{1}{\sigma^2}\sum_{i,j}\left(\frac{x-x_i}{\sqrt{(x-x_i)^2+(y-y_i)^2}}+\frac{x-x_j^{'}}{\sqrt{(x-x_j^{'})^2+(y-y_j^{'})^2}}\right)^2 \\ &= \frac{1}{\sigma^2}\sum_{i,j}\left(\frac{L-x_i}{\sqrt{(L-x_i)^2+(0-y_i)^2}}+\frac{L-x_j^{'}}{\sqrt{(L-x_j^{'})^2+(0-y_j^{'})^2}}\right)^2 \end{aligned} \tag{3.111}$$

Notice that the variance goes to zero when N and M are large and by using the law of large numbers together with the observations we made above:

$$\frac{J_{11}}{MN} \approx \frac{1}{MN\sigma^2}\sum_{i,j}(1+1)^2 \approx \frac{4}{\sigma^2} \tag{3.112}$$

Similarly :

$$\frac{J_{22}}{MN} \approx \frac{r^2}{2L^2\sigma^2} \tag{3.113}$$

$$\frac{J_{21}}{MN} = \frac{J_{12}}{MN} \approx 0 \tag{3.114}$$

All the $\approx$ are in the sense of convergence in probability as N, M goes to infinity. Now we have the Cramer-Rao bound

$$MN(V(x)+V(y)) = MN\frac{J_{11}+J_{22}}{J_{11}J_{22}-J_{12}^2} \approx \sigma^2\left(\frac{2L^2}{r^2}+\frac{1}{4}\right) \sim \sigma^2\left(\frac{2L^2}{r^2}\right) \tag{3.115}$$

□

So the position estimation variance

$$\boxed{V(x)+V(y) \sim \frac{\sigma^2}{MN}\left(\frac{2L^2}{r^2}\right)} \tag{3.116}$$

Following the same argument, we have the CR bound for an N transmitter, single receiver network.

$$\boxed{V(x)+V(y)\sim\frac{\sigma^2}{N}(\frac{4L^2}{r^2})} \tag{3.117}$$

We have some experimental results as show in Fig.3.29. For the MTMR case as in Fig(a), we randomly put 100 transmitters and 100 receivers inside the unit circle.For the MTSR case as in Fig(a), we randomly put 100 transmitters inside the unit circle, and the receiver at the origin $(0,0)$. The object is put on the x-axis from 0 to 100. The normalized CR bound is plotted in blue curve. And the theoretical bounds $\frac{2L^2}{r^2}$ for multiple transmitter, multiple receiver,$\frac{4L^2}{r^2}$ for multiple transmitter, single receiver are drawn as red crosses. They are consistent.

3.2.4.2 CR-bound in polar coordinate system

For some applications, the Euclidean position of the object is not interesting. Instead, we want to know the distance from the origin to the object ρ, and the direction of the line linking the origin and the object θ. The geometry is as shown in Fig.3.30. We now derive the Cramer-Rao estimation bound for the estimation of ρ,θ. The observations are still the multi-path distance measures $\overrightarrow{l}$.

Theorem 3.9. *CR bound in Polar coordinates for faraway region If the N transmitters and M receivers are uniformly distributed inside a circle with radius r. And the distance between the center of the sensor field and the object is L, s.t. $L\gg r$ Then*

$$lim_{N\to\infty,M\to\infty}MNV(\rho)=\frac{\sigma^2}{4}\quad \textit{in probability} \tag{3.118}$$

$$lim_{N\to\infty,M\to\infty}MNV(\theta)=\frac{2\sigma^2}{r^2}\quad \textit{in probability} \tag{3.119}$$

Proof. : The probability density is the same as in Eqn.3.47 by replacing (x,y) with the

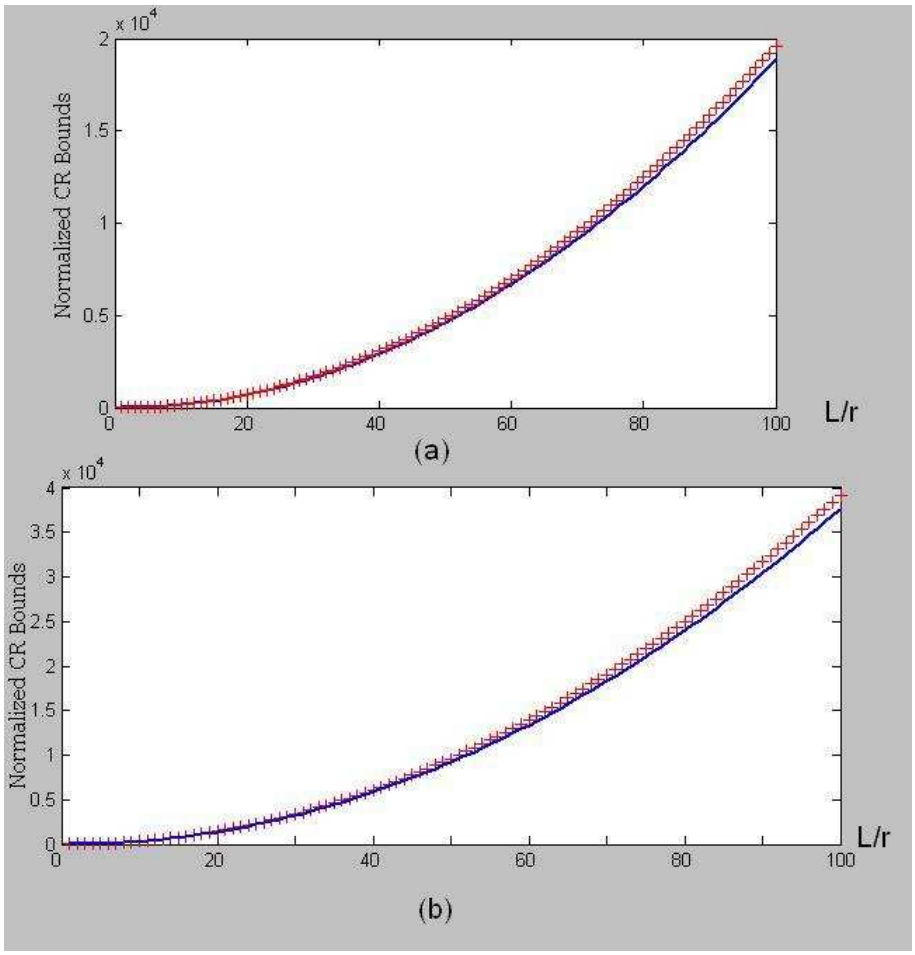


Figure 3.29: Theoretical bound (red plus)vs. experimental results (blue curve). (a)Normalized CR bound (V(x)+V(y)) for multiple transmitter, multiple receiver vs L/r
(b)Normalized CR bound (V(x)+V(y)) for multiple transmitter, single receiver vs L/r

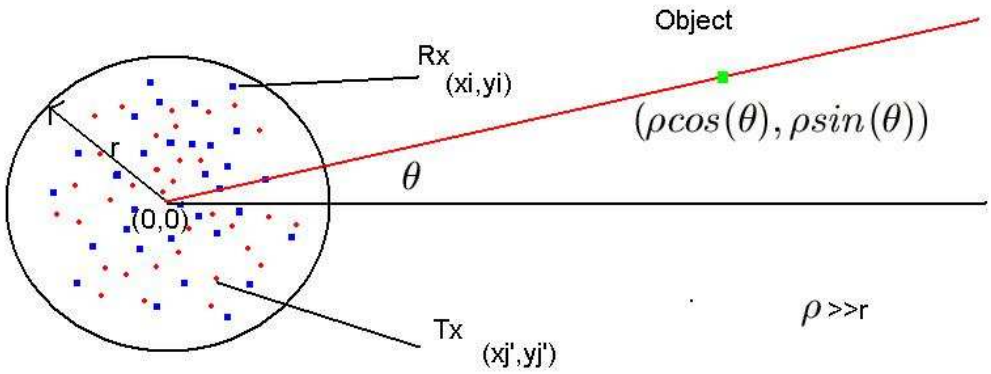


Figure 3.30: The object is far away from the sensor network, the position of the object is in polar coordinates

polar coordinates $(\rho cos(\theta), \rho sin(\theta))$.

$$p(\overrightarrow{l}|\theta,\rho) = \prod_{1\leqslant i\leqslant N, 1\leqslant j\leqslant M} \frac{1}{\sqrt{2\pi\sigma^2}} exp(\frac{-(l_{ij} - \sqrt{(\rho cos(\theta)-x_i)^2+(\rho sin(\theta)-y_i)^2} - \sqrt{(\rho cos(\theta)-x'_j)^2+(\rho sin(\theta)-y'_j)^2})^2}{2\sigma^2}) \tag{3.120}$$

Let θ be the first parameter, ρ the second parameter. The Fisher information matrix $J_{2\times 2}$ is as following.

$$\begin{aligned} ln(p(\overrightarrow{l}|\theta,\rho)) &= \frac{MN}{2}ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i,j}(l_{ij} - \\ &\quad \sqrt{(\rho cos(\theta)-x_i)^2+(\rho sin(\theta)-y_i)^2} - \sqrt{(\rho cos(\theta)-x'_j)^2+(\rho sin(\theta)-y'_j)^2})^2 \end{aligned}$$

$$\begin{aligned} J_{11} &= E(\frac{\partial^2 ln(p(\overrightarrow{l}|\theta,\rho))}{\partial\theta^2}) \\ &= \frac{1}{\sigma^2}\sum_{i,j}(\frac{-sin(\theta)\rho(\rho cos(\theta)-x_i)+cos(\theta)\rho(\rho sin(\theta)-y_i)}{\sqrt{(\rho cos(\theta)-x_i)^2+(\rho sin(\theta)-y_i)^2}} + \\ &\quad \frac{-sin(\theta)\rho(\rho cos(\theta)-x'_j)+cos(\theta)\rho(\rho sin(\theta)-y'_j)}{\sqrt{(\rho cos(\theta)-x'_j)^2+(\rho sin(\theta)-y'_j)^2}})^2 \\ &\approx \frac{1}{\sigma^2}\sum_{i,j}(sin(\theta)x_i - cos(\theta)y_i + sin(\theta)x'_j - cos(\theta)y'_j)^2 \end{aligned} \tag{3.121}$$

Using the law of large numbers and the known variances of x_i, y_i's. We have:

$$\frac{J_{11}}{NM} \approx \frac{r^2}{2\sigma^2} \tag{3.122}$$

Similarly :

$$\begin{aligned} J_{22} &= E(\frac{\partial^2 ln(p(\overrightarrow{l}|\theta,\rho))}{\partial\rho^2}) \\ &= \frac{1}{\sigma^2}\sum_{i,j}(\frac{cos(\theta)(\rho cos(\theta)-x_i)+sin(\theta)(\rho sin(\theta)-y_i)}{\sqrt{(\rho cos(\theta)-x_i)^2+(\rho sin(\theta)-y_i)^2}}+ \\ &\quad \frac{cos(\theta)(\rho cos(\theta)-x'_j)+sin(\theta)(\rho sin(\theta)-y'_j)}{\sqrt{(\rho cos(\theta)-x'_j)^2+(\rho sin(\theta)-y'_j)^2}})^2 \\ &\approx \frac{1}{\sigma^2}\sum_{i,j}(1+1)^2 \qquad (3.123) \\ \frac{J_{22}}{MN} &\approx \frac{4}{\sigma^2} \qquad (3.124) \\ \frac{J_{21}}{MN}=\frac{J_{12}}{MN} &\approx 0 \qquad (3.125) \end{aligned}$$

Thus:

$$lim_{N\to\infty,M\to\infty}MNV(\rho)=\frac{\sigma^2}{4} \text{ in probability} \qquad (3.126)$$

$$MNV(\theta)=\frac{2\sigma^2}{r^2} \text{ in probability} \qquad (3.127)$$

□

So the Cramer-Rao bound for the estimation of ρ is $V(\rho)=\frac{\sigma^2}{4NM}$, the estimation bound for θ is $V(\theta)=\frac{2\sigma^2}{NMr^2}$. Similarly for the N transmitter, 1 receiver case, assuming the receiver is at $(0,0)$. We have : $V(\rho)=\frac{\sigma^2}{4N}$, $V(\theta)=\frac{4\sigma^2}{Nr^2}$.

The reason why the variance of estimation error does not increase as the object moves farther away from the sensors is that the difference of multi-path distances for different sensors does not decrease to 0 as the object moves to infinity. Thus it is still possible to tell the θ-coordinate of the object.

We have some experimental results as show in Fig.3.31. For the MTMR case in Fig(a),We randomly put 100 transmitters and 100 receivers inside the unit circle with radius 2. For the MTSR case in Fig(b),We randomly put 4000 transmitters inside the unit circle with radius 2 and the receiver at the origin $(0,0)$. The object is put on the x-axis from 6 to 200. The normalized CR bound $V(\rho)$ is plotted in blue curve,$V(\theta)$ is

plotted in red curve. Notice $r = 2$ here, the simulation results are consistent with the theoretical analysis.

Here we have another experiment on a 6 transmitter, single receiver network. We put the receiver at the origin $(0, 0)$, the transmitters at $2(cos(\frac{2k\pi}{6}), sin(\frac{2k\pi}{6})), k = 0, 1, 2, 3, 4, 5$. The results are illustrated in Fig.3.32 and Fig.3.33. As can be seen, the Cramer-Rao bounds on both θ and ρ are larger inside the sensor field. Especially the bound on θ. The reason is that when the object is in the center of the sensor field, the multipath distance is not as sensitive to θ and ρ as when the object is far away from the sensor field. One extreme example is when the object is located in the origin $(0, 0)$, where θ is indifferent thus the Cramer-Rao bound is infinity.

3.2.4.3 Discussions on the position estimation variance

From the previous section, we know that there are different measures in the position estimation variance. If the purpose of the estimation is to figure out where the object is, then the variance $V(x) + V(y)$ is the right measure. However, if the absolute position of the object is not important, it could be true that $V(x) + V(y)$ does not best characterize the estimation variance for that specific interest. In this section, we explore the proper measure of the estimation variance for the following scenario. After estimating the position of the object, two new sensors join the network with position $(a_1, b_1), (a_2, b_2)$, and we are only interested in the multi-path distance estimation from sensor 1 to sensor 2 reflected by the object.

First we have the direct path length from sensor (a, b) to the object (x, y): $d = \sqrt{(a-x)^2 + (b-y)^2}$. Suppose the estimation of the position is deviated by a small vector $(\Delta x, \Delta y)$, then the error in the calculation in d is Δd.

$$\begin{aligned} \Delta d &= \frac{\partial d}{\partial x}\Delta x + \frac{\partial d}{\partial y}\Delta y = \frac{x-a}{d}\Delta x + \frac{y-b}{d}\Delta y \\ &= cos(\alpha)\Delta x + sin(\alpha)\Delta y \end{aligned} \tag{3.128}$$

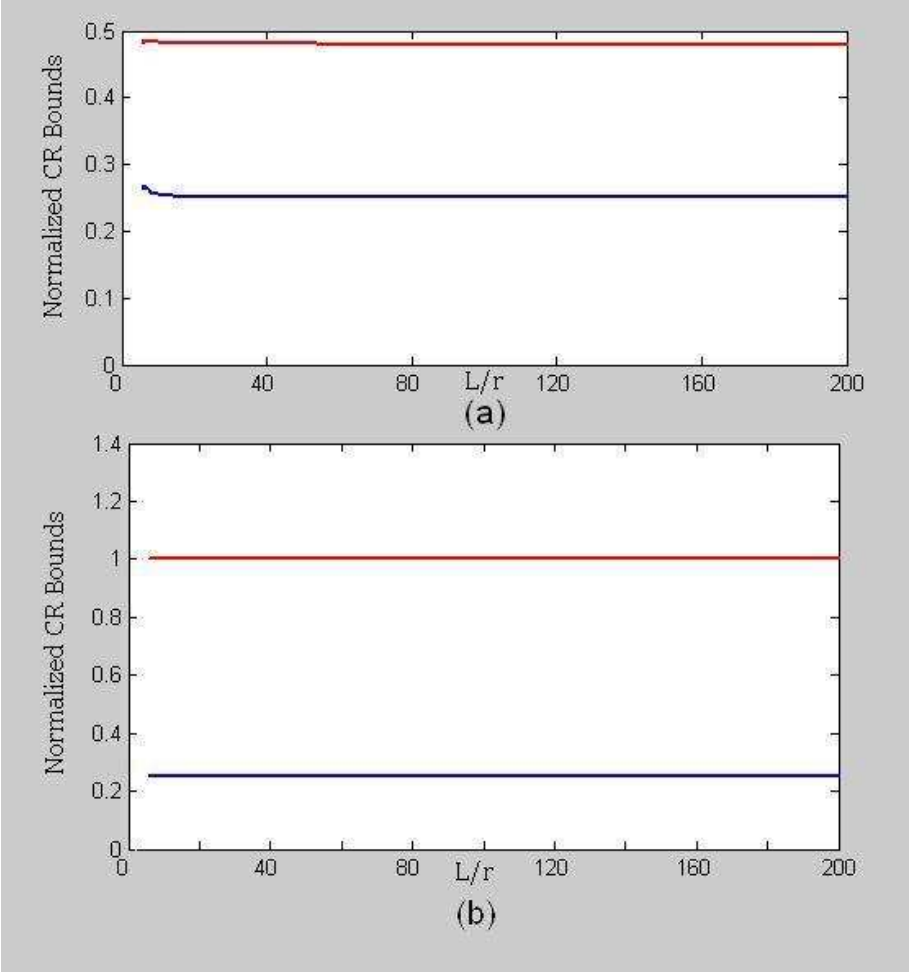


Figure 3.31: Experimental results (a)Normalized CR bound $\frac{NM}{\sigma^2}V(\rho)$(in blue), $\frac{NM}{\sigma^2}V(\theta)$(in red) for multiple transmitter, multiple receiver
(b)Normalized CR bound $\frac{N}{\sigma^2}V(\rho)$(in blue), $\frac{N}{\sigma^2}V(\theta)$(in red) for multiple transmitter, single receiver
Notice that they are all consistent with the theoretical results $V(\rho) \sim \frac{1}{4}$,$V(\theta) \sim \frac{2}{r^2}$ for MTMR, and $V(\theta) \sim \frac{4}{r^2}$ for MTSR

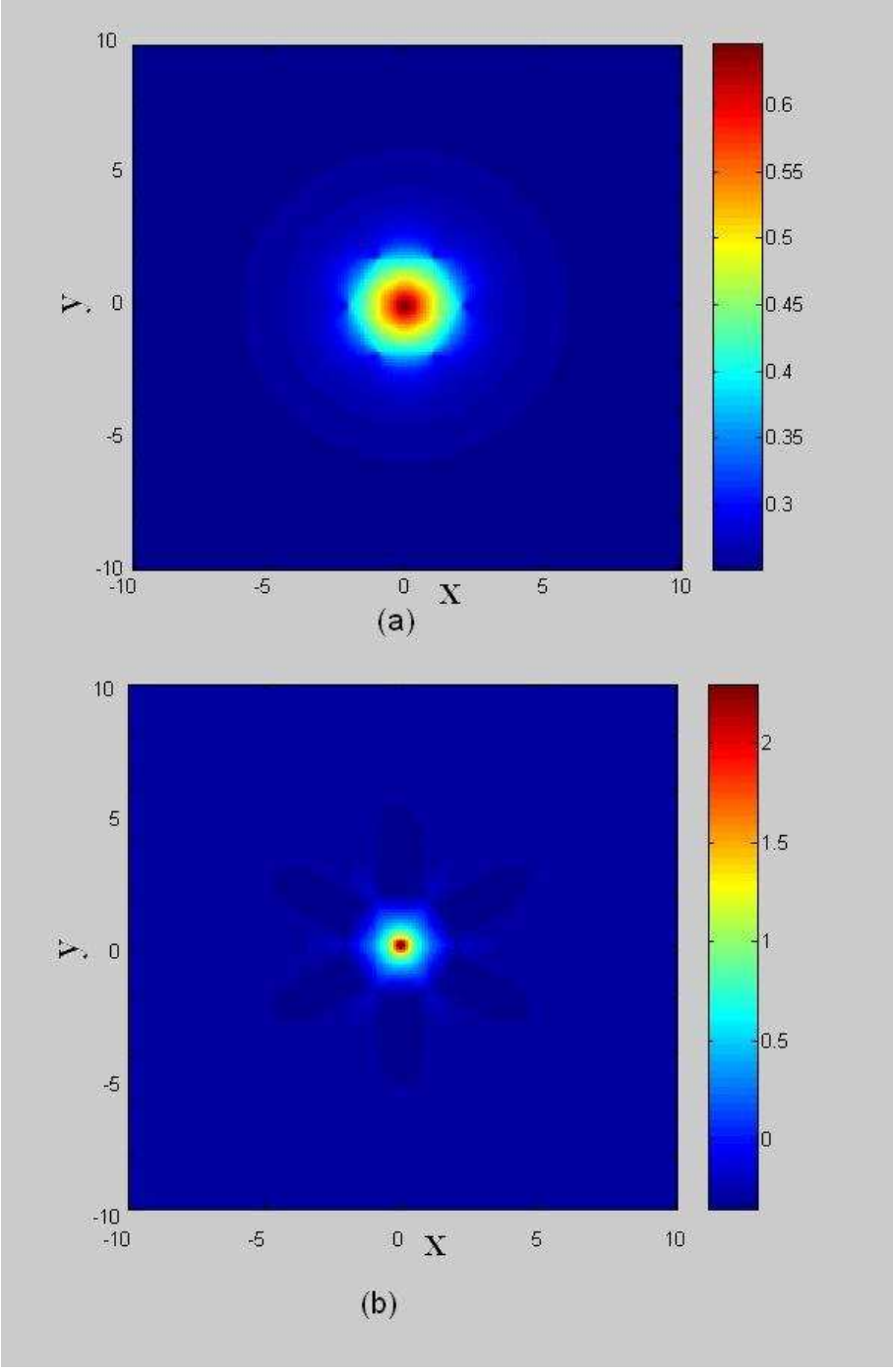


Figure 3.32: Normalized Cramer-Rao Bound (Polar coordinate) $(-10, 10) \times (-10, 10)$
(a)Normalized CR bound $V(\rho)$
(b)Normalized CR bound $log_{10}(V(\theta))$

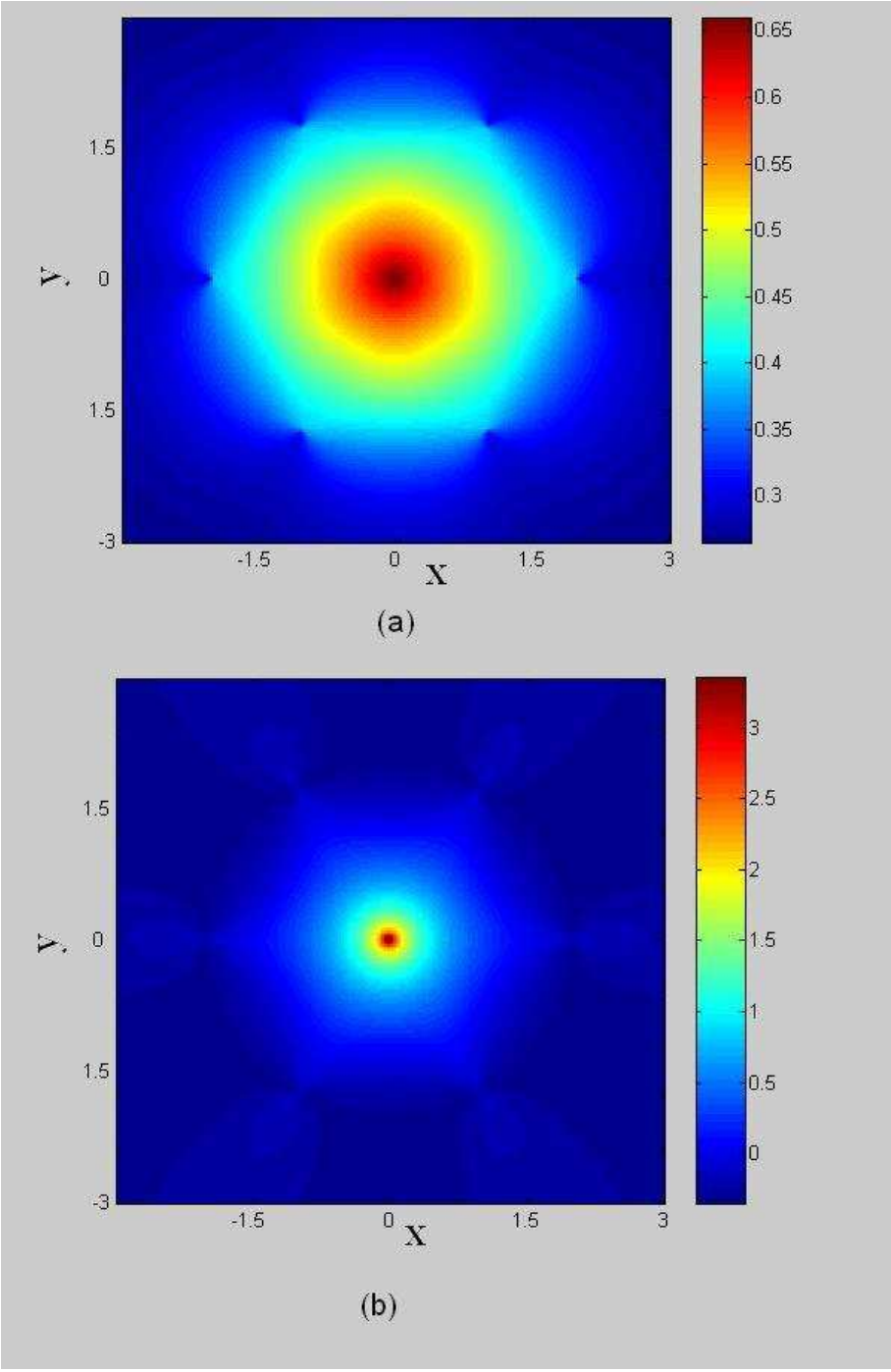


Figure 3.33: Normalized Cramer-Rao Bound (Polar coordinate)$(-3,3) \times (-3,3)$
(a)Normalized CR bound $V(\rho)$
(b)Normalized CR bound $log_{10}(V(\theta))$

The geometric interpretation of α is show in Fig.3.34

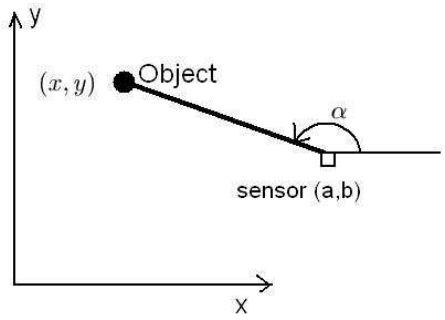


Figure 3.34: α

For sensor i, j, the multi-path distance $l = \sqrt{(a_i - x)^2 + (b_i - y)^2} + \sqrt{(a_j - x)^2 + (b_j - y)^2}$. The deviation $\Delta l = \Delta d_i + \Delta d_j = (cos(\alpha_i) + cos(\alpha_j))\Delta x + (sin(\alpha_i) + sin(\alpha_j))\Delta y$. Thus the the estimation variance of the multi-path distance l is $(\Delta l)^2$.

Case 1: the object is in the sensor network field. In this case, α can be thought as uniformly distributed in $[0, 2\pi]$. With basic trigonometry and assuming that $E(\Delta x) = E(\Delta y) = 0$, we know that $E(\Delta l^2) = E((cos(\alpha_i)+cos(\alpha_j))^2\Delta x^2+(sin(\alpha_i)+sin(\alpha_j))^2\Delta y^2) = V(x)+V(y)$. Thus the variance measure in the Euclidean metric is good in our purpose.

Case 2: the object is far away from all the sensors. In this case, all the α_k s are around the same angle α. Notice that $(x, y) = (\rho cos(\theta), \rho sin(\theta))$. For sensor pair $1, 2$ at $(a_i, b_i), i = 1, 2$,we have :

$$\begin{aligned}
\Delta l &= \Delta d_1 + \Delta d_2 = \sum_{i=1,2} \frac{\partial d_i}{\partial \rho}\Delta\rho + \frac{\partial d_i}{\partial \theta}\Delta\theta \\
&= \sum_{i=1,2} (cos(\alpha_i)cos(\theta) + sin(\alpha_i)sin(\theta))\Delta\rho + \sum_{i=1,2} (sin(\alpha_i)cos(\theta) - cos(\alpha_i)sin(\theta))\rho\Delta\theta \\
&= \sum_{i=1,2} cos(\alpha_i - \theta)\Delta\rho + \sum_{i=1,2} sin(\alpha_i - \theta)\rho\Delta\theta \qquad (3.129)
\end{aligned}$$

Notice that α_is are close to θ since the object is far away from the origin. Then we have $sin(\alpha_i - \theta) = \frac{k_i r}{\rho}, k_i \in [-1, 1]$ depending on the position of the sensor. Then we have $E(\Delta l^2) \approx \Delta\rho^2 + (k_1 + k_2)^2 r^2 \Delta\theta^2 \approx \Delta\rho^2 + 4r^2\Delta\theta^2$. So $V(\rho) + 4r^2 V(\theta)$ is a

good measure if the goal is to estimate the multi-path distance. If the goal is to predict the communication channel between 2 nodes, we need analyze the Cramer-Rao bound in both Euclidean and polar coordinates depending on the position of the object.

3.2.5 A linear algorithm for a small sensor network

3.2.5.1 The linear algorithm

For a sensor network with very few transmitters and receivers, we have the following simple linear estimation scheme. Notice that for a single transmitter-receiver pair: $T_x = (a_i, b_i), R_x = (u_j, v_j)$, if the multi-path distance measure is d_{ij}. Then we have the following equation :

$$\begin{aligned}
&d_{ij} = \sqrt{(x-a_i)^2 + (y-b_i)^2} + \sqrt{(x-u_j)^2 + (y-v_j)^2} \\
&(d_{ij} - \sqrt{(x-a_i)^2 + (y-b_i)^2})^2 = (x-u_j)^2 + (y-v_j)^2 \\
&(u_j - a_i)x + (v_j - b_i)y - d_{ij}\sqrt{(x-a_i)^2 + (y-b_i)^2} = \frac{u_j^2 + v_j^2 - a_i^2 - b_i^2 - d_{ij}^2}{2} \\
&(u_j - a_i)x + (v_j - b_i)y - d_{ij} l_{ti} = \frac{u_j^2 + v_j^2 - a_i^2 - b_i^2 - d_{ij}^2}{2}
\end{aligned} \tag{3.130}$$

Similarly, we have :

$$\begin{aligned}
&(a_i - u_j)x + (b_i - v_j)y - d_{ij}\sqrt{(x-u_j)^2 + (y-v_j)^2} = \frac{a_i^2 + b_i^2 - u_j^2 - v_j^2 - d_{ij}^2}{2} \\
&(a_i - u_j)x + (b_i - v_j)y - d_{ij} l_{rj} = \frac{a_i^2 + b_i^2 - u_j^2 - v_j^2 - d_{ij}^2}{2}
\end{aligned} \tag{3.131}$$

Where (x, y) is the position of the object. Write $l_{ti} = \sqrt{(x-a_i)^2 + (y-b_i)^2}$, the distance between the object and the i th transmitter. $l_{rj} = \sqrt{(x-u_j)^2 + (y-v_j)^2}$,F the distance between the object and the j th receiver. For an N transmitter M receiver sensor network, we have $2NM$ linear equations of $\vec{z} = (x, y, l_{t1}, l_{t2}, ...l_{tN}, l_{r1}, ...l_{rM})^T$. There are $N + M + 2$ unknowns, meanwhile we have $2NM$ equations. So we need $2NM \geq N + M + 2$, i.e $N + M \geq 4$. The linear equations are as following:

$$A\vec{z} = \vec{g} \tag{3.132}$$

Where the $2((i-1)M+j)-1$ th entry of the $2NM$ dimensional vector $\vec{g}$ is $\frac{u_j^2+v_j^2-a_i^2-b_i^2-d_{ij}^2}{2}$, the $2((i-1)M+j)$ th entry of $\vec{g}$ is $\frac{a_i^2+b_i^2-u_j^2-v_j^2-d_{ij}^2}{2}$. And if we write the k th row of A as $\alpha(\vec{k})$

$$\alpha(2((i-1)\vec{M}+j)-1)^T\vec{z} = (u_j-a_i)x+(v_j-b_i)y-d_{ij}l_{ti} \tag{3.133}$$

$$\alpha(2((i-\vec{1})M+j))^T\vec{z} = (a_i-u_j)x+(b_i-v_j)y-d_{ij}l_{rj} \tag{3.134}$$

$i=1,2,..N; j=1,2,..M$. With well conditioned matrix A, we can directly solve

$$\vec{z} = (A^TA)^{-1}A^T\vec{g} \tag{3.135}$$

From now on, we focus the discu ssion on $N=2, M=2$. The Cramer-Rao bound is different for different setups of transmitters and receivers. We give 2 setups of the transmitters and receivers as shown in Fig.3.35. The Cramer-Rao bound plots are shown in Fig.3.36, where σ^2 is the multi-path distance measure variance.

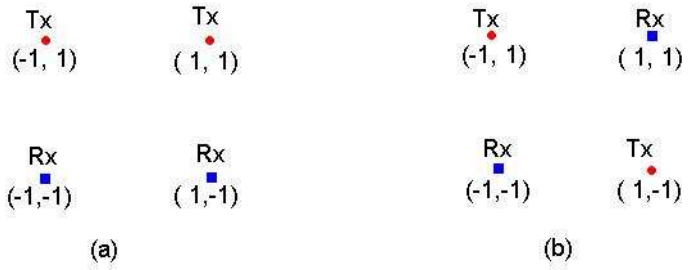


Figure 3.35: (a) setup 1: parallel (b) setup 2: crossing

The second setup has some obvious advantages in the sense of smaller average CR-bound. Furthermore, for the first setup there is a singular point at the origin, meanwhile there is no such singular point for setup 2. And in order to use the linear equation in Eqn.3.132 to solve for x, y, we also need the condition number $\chi(A) = \|A\|_2\|(A^TA)^{-1}A^T\|_2$ to be reasonably small. For the second setup, we have the condition number plot in Fig.3.37. The condition numbers are small for the setup, so we can

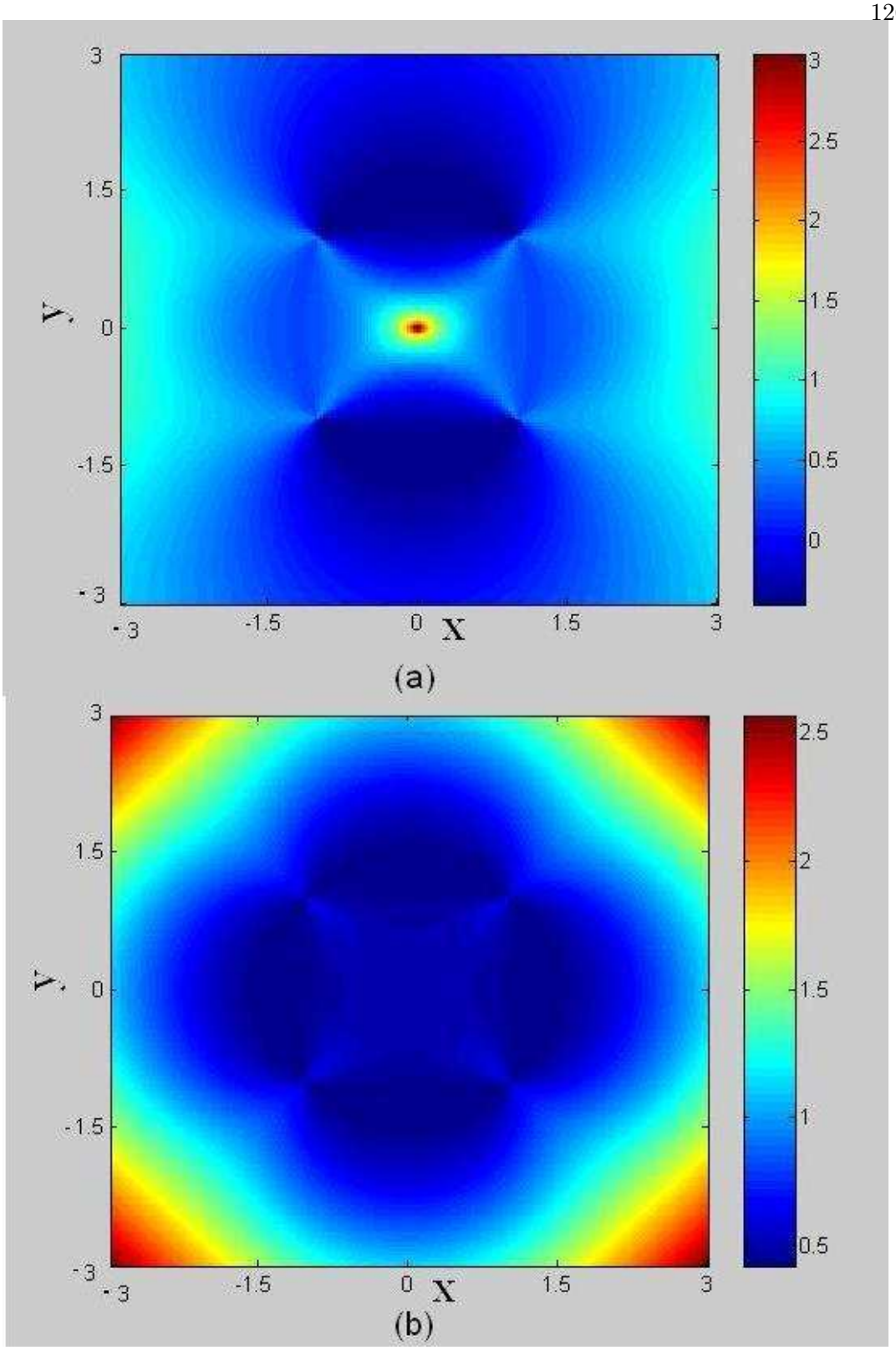


Figure 3.36: (a)$log_{10}(\frac{V(x)+V(y)}{\sigma^2})$ CR bound for setup 1

(b) $\frac{V(x)+V(y)}{\sigma^2}$ CR bound for setup 2

directly use Eqn.3.132 to solve for x, y.

$$\vec{z} = (A^T A)^{-1} A^T \vec{g} \tag{3.136}$$

Where $\vec{z}$ and $\vec{g}$ are $6 \times 1, 8 \times 1$ vectors respectively, A is a 8×6 matrix.

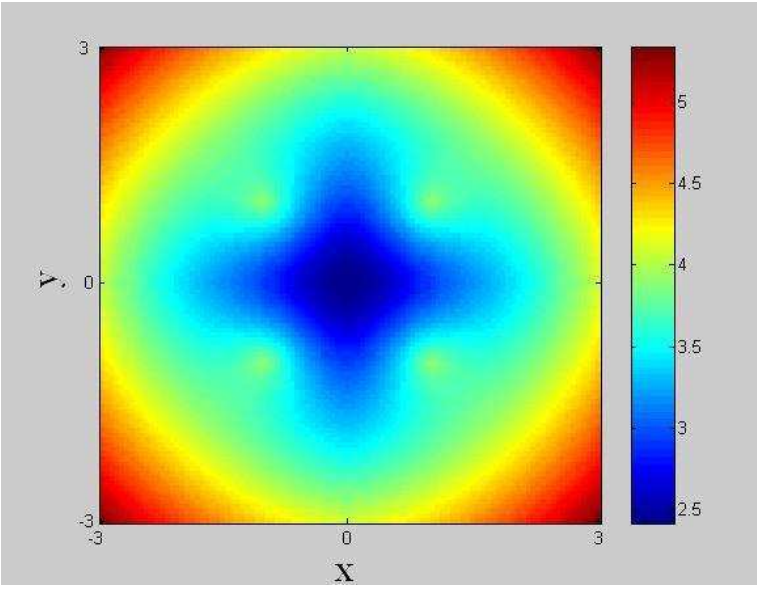


Figure 3.37: $\chi(A)$

3.2.5.2 Simulation Results

In our simulation, we put the receivers at $(1,1),(-1,-1)$, transmitters at $(-1,1),(1,-1)$ as shown in Fig.3.35(b). $(a_1,b_1)=(-1,1),(a_2,b_2)=(1,-1)$,$(u_1,v_1)=(-1,-1),(u_2,v_2)=(1,1)$. In real experiment, we expect noises not only on the multi-path distance measure, but also on the positions of the transmitter and receivers. So we have the multi-path distance measurement

$$d_{ij} = \sqrt{(a_i + \xi_{ai} - x)^2 + (b_i + \xi_{bi} - y)^2} + \sqrt{(u_j + \xi_{uj} - x)^2 + (v_j + \xi_{vj} - y)^2} + \epsilon_{ij}$$

Where ξ_{ai} is the additive noise to the x coordinate of the i th transmitter, similar for $\xi_{bi}, \xi_{uj}, \xi_{vj}$, they are assume to be IID Gaussian noises $\sim N(0, \varsigma^2)$. ϵ_{ij} is the additive noise to the multi-path distance measure from the i the transmitter to the j th receiver.

And $\epsilon'_{ij}s$ are are assumed to be IID Gaussian $\sim N(0, \sigma^2)$.

And in Fig.3.38 we plot the average square estimation error $(\hat{x} - x)^2 + (\hat{y} - y)^2$ over 100 independent experiments. Where $\hat{x}, \hat{y}$ are directly calculated from Eqn.3.136. In Fig.3.38(a), we have the plots for $\sigma_1^2 = 4 \times 10^{-4}$, $\varsigma^2 = 0$, (no sensor position error case). in Fig.3.38(b) we have the plots for $\sigma_1^2 = 4 \times 10^{-4}$, $\varsigma^2 = 10^{-4}$.

As can be seen in the Cramer-Rao plot in Fig.3.36, setup 2 is much better than setup 1. This observation brought up an interesting question, what is the optimal placement for the transmitters and receivers? The design placement of the sensor network is likely to be interesting to study, especially for a small sensor network.

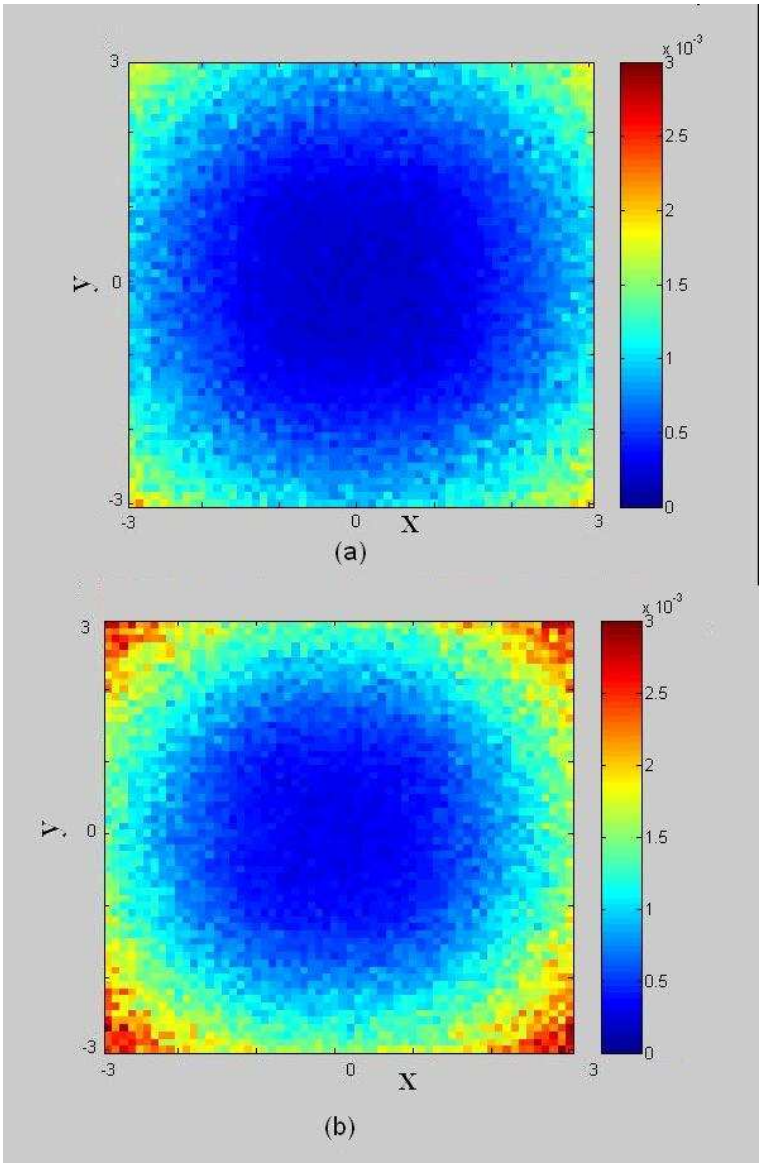


Figure 3.38: Average value of $(\hat{x}-x)^2+(\hat{y}-y)^2$
(a)$\sigma_1^2 = 4\times10^{-4}$,$\varsigma^2 = 0$ (b)$\sigma_1^2 = 4\times10^{-4}$,$\varsigma^2 = 10^{-4}$

3.3 Multiple Object Tracking by a Multiple-Sensor Sensor network (MSMO)

In the case where multiple objects are in the sensor network field, for a transmitter-receiver pair, there are multiple reflections bouncing from the objects. As show in Fig.3.39.

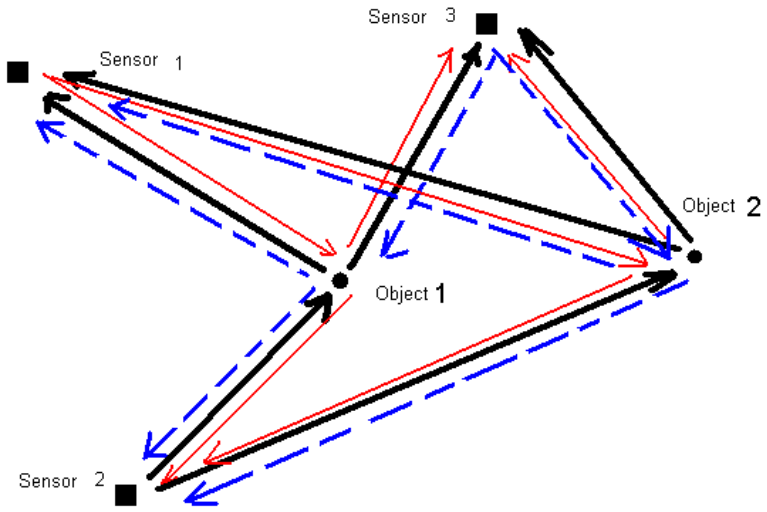


Figure 3.39: Multiple-Sensors, Multiple Objects

If there are M objects in the field. For a single transmitter-receiver pair transmitter i, receiver j, there are M mutli-path lengh estimations $d_1(i,j) < d_2(i,j) < ... < d_M(i,j)$. Since there is no intrinsic ID for the objects, we will order the objects by the order of the transmitter-receiver pair $1, 2$. i.e. the object with multi-path measurement $d_1(1,2)$ will be named $object-1$, the object with measurement $d_2(1,2)$ will be called $object-2$.. and so on. For another transmitter-receiver pair $k, l, (k,l) \neq (1,2)$, there are M mutli-path lengh $d_1(k,l), d_2(k,l), ..., d_M(k,l)$, but which measurement is for $object-m, m = 1, 2, ..., M$ is unknown. We can use the algorithm mentioned in the previous section to help estimating the positions of the objects.

3.3.1 An Exhaustive Search Algorithm

We can exhaust all the the possibilities to search for the optimal position estimation of the objects. To simplify the problem suppose for every object k, for any sensor pair (i, j), sensor i receives the signal from sensor j reflected by $object - k$. In an N sensors, M objects network. There are $N(N-1)$ transmitter-receiver pairs. For transmitter i, receiver j, there are M mutli-path lengh estimations $d_1(i,j), d_2(i,j), ..., d_M(i,j)$.

Let F be the set of all $1-1$ mappings from $1, 2, ...M$ to $1, 2, ...M$, i.e. $\forall f \in F$, $f(i) \in 1, 2, ..., M, \forall i$ and $f(i) \neq f(j)$, if $i \neq j$. And F contains all such f. Write $G = F \times F \times ... \times F = F^{N(N-1)-1}$.

Given the position of the i th sensor (a_i, b_i). Assign $N(N-1)$ multi-path distnace measurements $\overrightarrow{l} = (l(1,2), l(1,3), ..., l(1,N), l(2,1), ..., l(2,N), ..., l(N,1), ..., l(N,N-1))$, where $l(i,j)$ is the multi-path distance measurements at sensor j from the signal sent by sensor i. Using the multi-sensor single-object position estimation algorithm in the previous section, we write $(u, v) = H(\overrightarrow{l})$, where (u, v) is the estimation of the position of the object.

Thus we have the following exhaustive search algorithm:

Definition 3.1 (position estimation for a map g, $(\overrightarrow{u(g)}, \overrightarrow{v(g)})$). *: Suppose $g \in G$, we write $g_k, k = 1, 2, ..., N(N-1)-1$ as the kth map of g, then $\overrightarrow{u(g)} = (u(g)_1, ..., u(g)_M), \overrightarrow{v(g)} = (v(g)_1, ..., v(g)_M)$, $(u(g)_i, v(g)_i)$ is the position estimation of $object - i$ given map g. i.e. $(u(g)_i, v(g)_i) = H(\overrightarrow{l_i})$. $\overrightarrow{l_i} = (d_i(1,2), d_{g_1(i)}(1,3), ..., d_{g_{N(N-1)-1}(i)}(N, N-1)$. For the simplicity of future notation, we define map K, $K(1,2) = 0, K(1,3) = 1, ..., K(N,N-1) = N(N-1)-1$ and g_0 is the identical map no matter what g is .*

Now we have the following exhaustive search algorithm :

$\overrightarrow{u(g^*)}, \overrightarrow{v(g^*)}$ is the position estimation of the M objects. Where $(u(g^*)_i, v(g^*)_i)$ is

the position estimation of the i th object. Where

$$g^* = argmin_g \sum_{t=1}^{M} \sum_{1 \le i \ne j \le N)} (\sqrt{(u(g)_t - a_i)^2 + (v(g)_t - b_i)^2} + \sqrt{(u(g)_t - a_j)^2 + (v(g)_t - b_j)^2} - d_{g_{K(i,j)}(t)}(i,j))^2 \tag{3.137}$$

In one word, the algorithm searches for all possible correspondence between the multi-path distance measurements, and find the optimal one. And there are $M!$ different correspondence for 2 different transmitter-receiver pair, and if we have $N(N-1)$ transmitter-receiver pairs in the sensor network, there are total $(M!)^{N(N-1)-1}$ different correspondence. Which is huge even there are only 2 objects. Thus the algorithm which simply searches all the possible correspondences are not quite practical.

3.3.2 Hough Transform Inspired Algorithm

In this section, we introduce an algorithm inspired by the Hough Transform [14]. The Hough transform is a standard method in finding some geometric structures from a noisy image. Given a noisy image as the input from the image space, the Hough transform is trying to find the line structures in the dual space of line equations. The idea behind our scheme is similar. Given the multi-path distances, we simply want to find out the points on the dual space (2D space) with large likelihood. Again, we are dealing with an N sensor M objects sensor network. And we suppose that the additive noise to the mutli-path measurements are iid Gaussian $\sim N(0, \sigma^2)$. Now given transmitter i, receiver j, a point (x, y), we define a likelihood function for the 4-tuple (x, y, i, j). The likelihood function $L_{i,j}(x, y)$ tells how likely there exists *an* object on the point (x, y) given the multi-path distance measurements from transmitter i to receiver j, $d_1(i,j), d_2(i,j), ...d_{M_{i,j}}(i,j)$

Definition 3.2 (Likelihood function $L_{i,j}(x,y)$ of (x,y,i,j)).

$$L(x,y,i,j) = max_k \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(\sqrt{(x-a_i)^2+(y-b_i)^2}+\sqrt{(x-a_j)^2+(y-b_j)^2}-d_k(i,j))^2}{2\sigma^2}} \tag{3.138}$$

In the above definition, we do not assume that the jth receiver has exactly M multi-path measurements, which is true if 2 objects have the same multi-path distance measurements.

With the assumption on the measurement error, we can define the likelihood function of a point (x,y), which tells how likely there is *an* object on the point (x,y).

Definition 3.3 (Likelihood function $L(x,y)$ of (x,y)).

$$\begin{aligned} L(x,y) &= \prod_{(i,j)\in S} L_{i,j}(x,y) \\ &= \prod_{(i,j)\in S} max_k \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(\sqrt{(x-a_i)^2+(y-b_i)^2}+\sqrt{(x-a_j)^2+(y-b_j)^2}-d_k(i,j))^2}{2\sigma^2}} \\ &= (\frac{1}{\sqrt{2\pi\sigma^2}})^{|S|} e^{\sum_{i,j\in S} max_k -\frac{(\sqrt{(x-a_i)^2+(y-b_i)^2}+\sqrt{(x-a_j)^2+(y-b_j)^2}-d_k(i,j))^2}{2\sigma^2}} \end{aligned} \tag{3.139}$$

Where S={(i,j)|i is a transmitter, j is a receiver}. The larger $L(x,y)$ is, the more likely there is an object on the point (x,y). Notice that the only thing really matters is the exponential part. Thus we have the following definition of the cost function of point (x,y).

Definition 3.4 (Cost function $C(x,y)$ of (x,y)).

$$C(x,y) = \sum_{(i,j)\in S} min_k(\sqrt{(x-a_i)^2+(y-b_i)^2}+\sqrt{(x-a_j)^2+(y-b_j)^2}-d_k(i,j))^2 \tag{3.140}$$

Notice that, the smaller $C(x,y)$ is, the more likely there is an object is on (x,y). In the noiseless case, there are generally M points with 0 cost function and those points are where the objects are . The cost function is always non-negative, so we have M

local optimal points in terms of the cost function. But the cost function $C(x, y)$ is not convex for as can be seen in the following example. $C(x, y) \geq 0$, if $C(x, y)$ is convex then if $C(x_1, y_1) = 0, C(x_2, y_2) = 0$ implies that $C(tx_1 + (1-t)x_2), (ty_1 + (1-t)y_2) = 0$, $\forall t \in [0, 1]$. This is not always true because if the multipath distance measures are noise-free, then $C(x, y) = 0$ on only finite points. And there are a huge number of local optimal points for the cost function $C(x, y)$. However, our goal is NOT to find the point (x, y) with the smallest cost function. Thus we will not use any traditional optimal algorithm to solve the problem. Instead we will use an approximation algorithm inspired by the Hough Transform. Our scheme is consisted of two steps. In the first step, we use Hough Transform to find some candidate points. With candidate points, we get associates between multipath distance measures $d_k(i, j)'s$ and the objects. Then we invoke the MSSO algorithm to give the final estimation of the positions of the objects.

3.3.2.1 The Hough Transform Inspired Algorithm

The Hough Transform is to first discretize a bounded space. Then search for optimal solutions on the grids. In our problem, we can assume the objects are in a bounded $2D$ space. So we first define the searching region (x_l, x_r, y_u, y_d) where all those objects are possibly presented, without loss of generality we let $x_r - x_l = y_d - y_u = D$ where $x_l < min(a_i), x_r < max(a_i), y_d < min(b_i), y_u > max(b_i)$. We will constraint our search to the rectangular region $\{(x, y)|x_l < x < x_r, y_d < y < y_u\}$. Then we discretize the searching region into a $N \times N$ grid. i.e. we will do our searching on the points $(x_l + iW, y_d + jW), 0 \leqslant i \leqslant N, 0 \leqslant j \leqslant N$, and $W = \frac{D}{N}$. Our scheme is to search for a local optimal point in the grid points, then set a threshold and pick all those points better than the threshold to be candidate points for the objects. The cost function $C(x, y)$ is not good for our purposes. First, $C(x, y)$ is not bounded so it is hard to set a threshold. Second is that if for a transmitter-receiver pair, a multi-path distance measurement for an object at x, y is corrupted by a noise with very large magnitude

which is likely to happen when some multipaths are blocked by the objects, then $C(x, y)$ would be largely affected by that single *bad* measurement especially when the number of transmitter-receiver pairs are not big. i.e. $C(x, y)$ is not very stable in a sensor network with small number of sensors. In Appendix M, we will give an example of how one bad measurement could affect the cost function. Instead of using the Cost function $C(x, y)$, we use the following score function $S(x, y)$ which is related to $C(x, y)$.

Definition 3.5 (Score function $S(x, y)$ of (x,y)).

$$S_{i,j}(x, y) = min_k(\sqrt{(x - a_i)^2 + (y - b_i)^2} + \sqrt{(x - a_j)^2 + (y - b_j)^2} - d_k(i, j))^2, for\ (i,j) \in S$$
$$S(x, y) = \sum_{(i,j)\in S} MAX(1 - \frac{S_{i,j}(x, y)}{K}, 0) \tag{3.141}$$

K is a parameter to be determined. Notice that $0 \leq S(x, y) \leq |S|$. The value of K determines the behavior of the score function $S(x, y)$. If K is too large, the score function cannot distinguish the object with another point very well. If K is too small, then a little bit noise or the discretized error could give a 0 score to the true object position, which again cannot tell any differences between an object with an arbitrary point. A proper value for K should be able to tolerate the noises and the discretizing errors yet still distinguish the objects from those points which are far away from the objects.

In searching for a local optimal point, since we are working on a discrete space, we have to define the term local. A proper size of the local region should be defined. In the second step of the algorithm, we discard all the candidate local optimal points below $r \times N_{Tx}N_{Rx}$, where (x, y) is a candidate point, N_{Tx}, N_{Rx} are number of transmitters and receivers respectively. The threshold parameter r is another value we need to select. A proper value of r would only leave the *right* local optimal points above the threshold. We will analyze the proper selections of K, r in the discussions.

Now we summarize the Hough transform inspired algorithm :

1)Discretizing the square region into $N \times N$ grids.

2)Searching on all the grid-points for local maximal points of score function $S(x, y)$, generate a candidate set $\{(x_i, y_i), i = 1, ...K\}$.

3) For those (x_i, y_i), s.t. $S(x_i, y_i) > rN_{Tx}N_{Rx}$, report a discovery of an object.

3.3.2.2 Simulation Results

In our simulation, all the sensors and objects are uniformly distributed in a $T \times T$ region, where $T = \frac{3aN}{D}$. There are M objects. We pick the parameters K, r as following:

$$K = W^2 + \sigma^2, r = 0.4 \tag{3.142}$$

And the local optimal points are those which are optimal in the 3×3 region around it.

3.3.2.3 Discussions on the Hough Transform Inspired Algorithm

In an N sensor, M objects sensor network. The algorithm needs to compute the value of all functions $S_{(i,j)}(x, y)$. Suppose for each transmitter-receiver pair (i, j), we have M multi-path distance measurements, for each $S_{(i,j)}(x, y)$, we need to do M basic calculations as in Eqn. 3.141. So totally we need to do $MN^2|S|$ basic calculations. If all the sensors can both send and receive signals, that number is $MN^2L(L-1)$.

And the accuracy of the Hough Algorithm is ultimately limited by the size of the grids which is $\frac{D}{N}$. However, we can use the MSSO algorithm to give a more accurate estimation of the position of the objects. Suppose (x, y) is one of the object positions reported by the Hough Algorithm. Then for each transmitter-receiver pair (i, j), we write the following multi-path distance measurement to be the *true* one.

$$l(i,j) = argmin_{d_k(i,j)}(\sqrt{(x-a_i)^2 + (y-b_i)^2} + \sqrt{(x-a_j)^2 + (y-b_j)^2} - d_k(i,j))^2$$
$$k = 1, 2, ..M \tag{3.143}$$

Then we can apply the MSSO algorithm to give a better estimation of the objects. Here we have a 2-step algorithm. The 1st step is to use the Hough transform algorithm

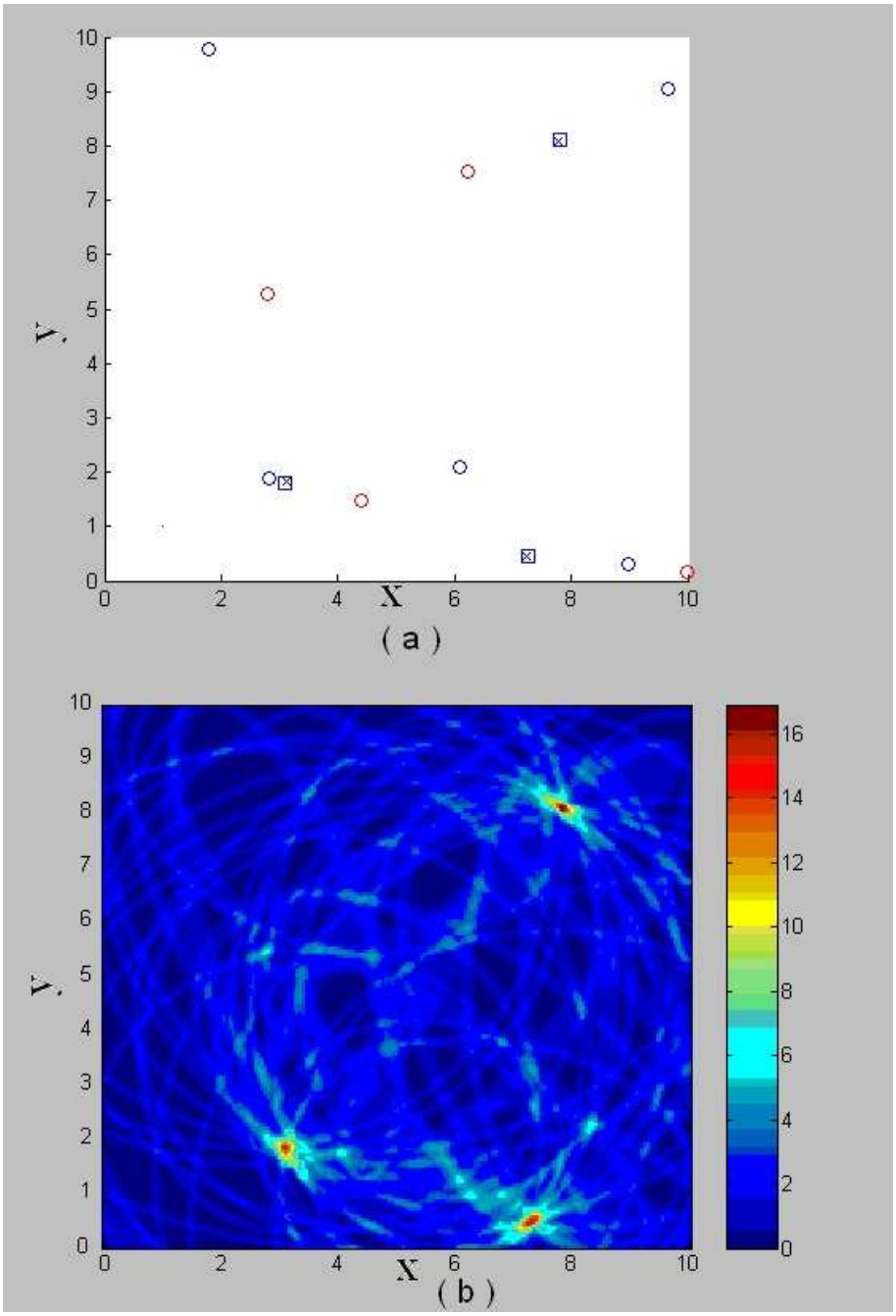


Figure 3.40: 5 transmitters, 4 receivers, 3 objects D=10, N=200,$\sigma^2 = 0.01$
(a) blue circles: Tx; red circles: Rx; $\times$: object; $\square$ estimation
(b) score function $S(x, y)$

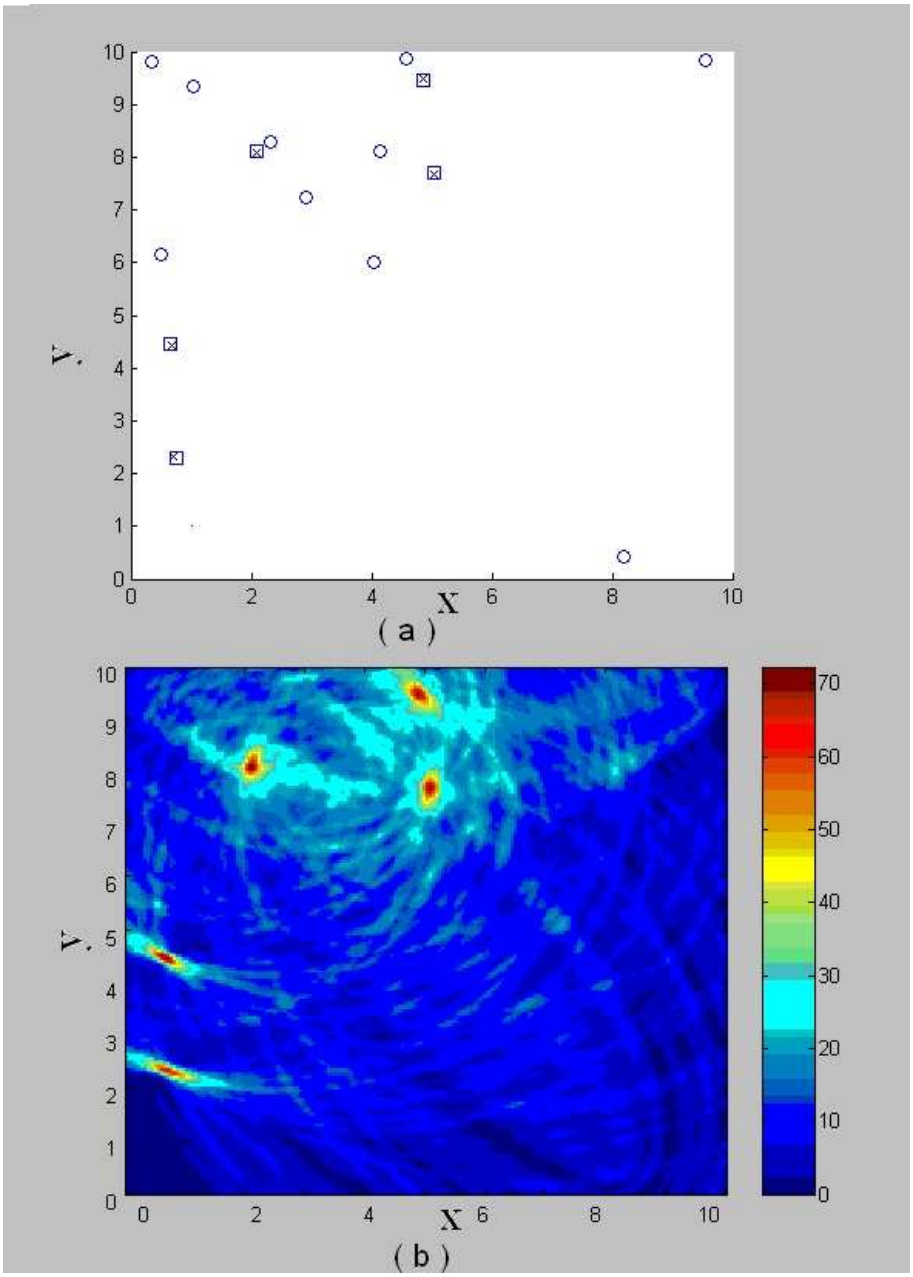


Figure 3.41: 10 transmitters-receivers, 5 objects D=10, N=200,$\sigma^2 = 0.01$
(a) blue circles: Tx-Rx; ; $\times$: object; $\square$ estimation
(b) score function $S(x, y)$

to match each object with a candidate grid, then match a multi-path distance from each Tx-Rx pair to that grid. The 2nd step is to input all those multi-path distances to the MSSO algorithm and give the ultimate estimation of the positions of the objects.

If the Hough transform algorithm matches all the *true* distances to the object, the overall performance would be the same as the performance of MSSO.

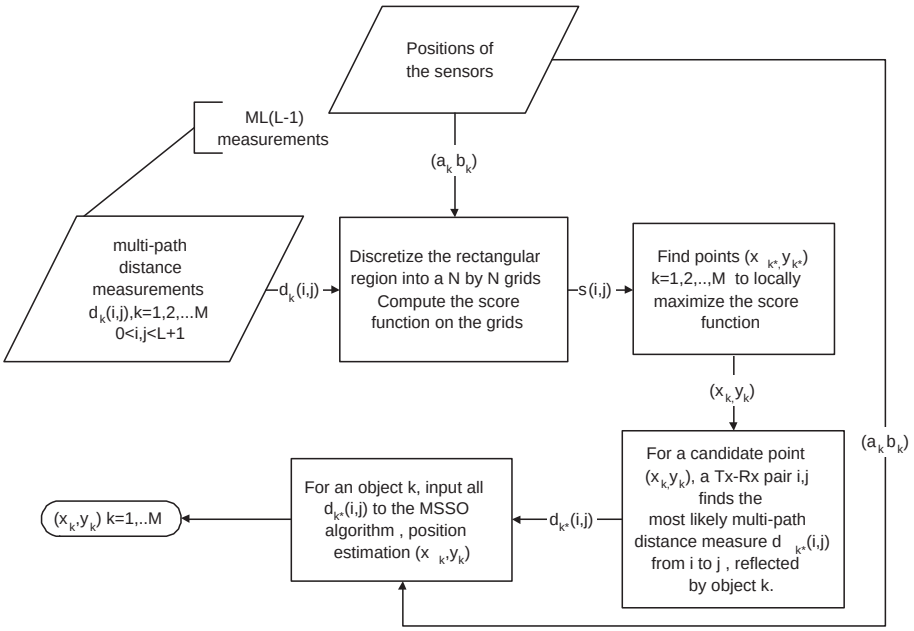


Figure 3.42: Flowchart of MSMO position estimation

As can be seen, the proper selections of K and r are crucial to the success of the algorithm. Here we analyze the behavior of the score function $S(x, y)$, then deduce the criteria for the selections of K and r. As shown in Fig.3.43(a). Point B at (u, v), is the closest grid point to the true position of the object, A at (x, y). Notice that $\|u - x\| \leq \frac{W}{2}, \|v - y\| \leq \frac{W}{2}$.

We hope the Score function to be prominently larger on point B than other grid points. For a single transmitter receiver pair i, j, with position $(a_i, b_i), (a_j, b_j)$ respectively. The multi-path distance measurement is $l = \sqrt{(x - a_i)^2 + (y - b_i)^2} + \sqrt{(x - a_j)^2 + (y - b_j)^2} + \epsilon$. Where ϵ is assumed to be Gaussian $\sim N(0, \sigma^2)$. Now we

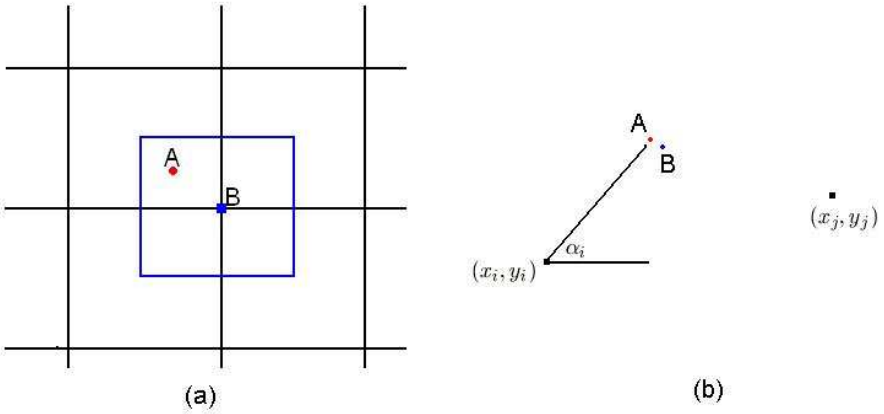


Figure 3.43: (a)A is the object, B is the closest grid point (b)definition of α

have the cost function on (u, v) as following:

$$\begin{aligned} C(u,v) &= (\sqrt{(u-a_i)^2+(v-b_i)^2} - \sqrt{(x-a_i)^2+(y-b_i)^2} \\ &+ \sqrt{(u-a_j)^2+(v-b_j)^2} - \sqrt{(x-a_j)^2+(y-b_j)^2} - \epsilon)^2 \end{aligned} \quad (3.144)$$

Noticing that $u-x, v-y$ are bounded by $\frac{W}{2}$, half of the grid size, which is generally very small. Thus by doing the Taylor expansion, we have:

$$C_{i,j}(u,v) = ((cos(\alpha_i) - cos(\alpha_j))(u-x) + (sin(\alpha_i) - sin(\alpha_j))(v-y) - \epsilon)^2$$

Where $cos(\alpha_i) = \frac{x-a_i}{\sqrt{(x-a_i)^2+(y-b_i)^2}}$,$sin(\alpha_i) = \frac{y-b_i}{\sqrt{(x-a_i)^2+(y-b_i)^2}}$, as show in Fig.3.43(b). Assume that the angles α's are uniformly distributed in $[0, 2\pi]$ and the independence of the noise. We have : $E(C_{i,j}(u,v)) = (u-x)^2 + (v-y)^2 + \sigma^2 \leq \frac{W^2}{2} + \sigma^2$.Now we pick K to be $W^2 + 2\sigma^2$, which is around the expectation value of the cost function. $S_{i,j}(u,v) = max(1 - \frac{C_{i,j}(u,v)}{K}, 0)$, if the size of the grid is much smaller than the square root of the noise variance,i.e. $W^2 << \sigma^2$, then we have the distribution of $S(i,j)$ as following, let $\zeta = 1 - S_{i,j}(u,v)$. Then

$$Pr(\zeta = 1) = 2\int_{-\infty}^{-\sqrt{2}} \frac{1}{\sqrt{2\pi}} exp(-\frac{x^2}{2})dx \approx 0.1573 \quad (3.145)$$

$$p(\zeta) = \frac{1}{\sqrt{\pi\zeta}} exp(-\zeta), \text{for } \zeta \in [0,1) \quad (3.146)$$

$E(S_{i,j}(u,v)) \approx 1 - \frac{1}{2\sqrt{2\pi}}(\int_{-\sqrt{2}}^{\sqrt{2}} x^2 exp(-\frac{x^2}{2})dx + 2\int_{-\infty}^{-\sqrt{2}} exp(-\frac{x^2}{2})dx) \approx 0.629$. So the total score function $S(u,v) = \sum_{i,j} S_{i,j}(u,v) \approx 0.629 N_{Tx} N_{Rx}$,where (x,y) is a candidate point, N_{Tx}, N_{Rx} are number of transmitters and receivers respectively.

We now have the following hypothesis testing problem:

H_0: (u,v) is the closest grid point to an object.

H_1: (u,v) is not the closest grid point to an object.

And the decision is made on the observations of $S(u,v) = \sum_{i,j} S_{i,j}(u,v)$ with decision rule: reject H_0, if $S(x,y) < rN_{TR}$, where N_{TR} is the total transmitter receiver pairs. Assuming the independence of $S_{i,j}(u,v)$ and with the assumption that $W^2 << \sigma^2$. We

have the following table for r on the plane of N_{TR}vs $Pr(reject\ H_0|H_0)$. The results are based on our Monte Carlo experiment of sample size 20000.

	$\alpha = 0.01$	$\alpha = 0.05$	$\alpha = 0.10$	$\alpha = 0.20$	$\alpha = 0.50$
$N_{TR} = 5$	0.2261	0.3463	0.4067	0.4902	0.6370
$N_{TR} = 10$	0.3513	0.4302	0.4768	0.5308	0.6330
$N_{TR} = 15$	0.3993	0.4692	0.5052	0.5500	0.6326
$N_{TR} = 30$	0.4699	0.5172	0.5425	0.5730	0.6305
$N_{TR} = 60$	0.5169	0.5515	0.5691	0.5903	0.6301
$N_{TR} = 100$	0.5417	0.5682	0.5818	0.5981	0.6293
$N_{TR} = 200$	0.5681	0.5865	0.5962	0.6079	0.6290
$N_{TR} = 400$	0.5851	0.5986	0.6053	0.6135	0.6289

We pick r to be 0.4 which would work for not too big $N_{Tx}N_{Rx}$.

3.4 Conclusions and Future Work

In this chapter, we studied the object tracking by a sensor network problem. We argued that in a single transmitter, single receiver network, the position of the object can not be accurately estimated by showing that the Cramer Rao bound is very big even with an unrealistic constraint on the motion. Furthermore if the object moves with arbitrary velocity, then it is impossible to track the motion of the object in a single transmitter, single receiver network because two different motions could yield same multipath distances measures all the time.

For tracking an object in a multiple nodes sensor network, we computed the Cramer-Rao bounds in different scenarios based on the number of transmitter or receivers and analyzed the asymptotic behavior of the Cramer-Rao bounds as the number of sensors goes to infinity. It turns out that the estimation accuracy is proportional to the total received SNR in the sensor network. First we showed that the Cramer-Rao bound is inversely proportional to the the number of total transmitter receiver pairs, if the object is in the center of the sensor field. Then we found that the Cramer-Rao bound in the Euclidean coordinate increases proportionally to the square of distance between the object and the sensors if the object is faraway from the sensor field, even if the distance measure remains the same accuracy.

We then gave an algorithm which is order optimal. For a sensor network with a small number of sensors, we gave a linear algorithm and we thoroughly studied the tracking problem in a 2 transmitter, 2 receiver sensor network. By seeing the performance difference of two different placements of sensors, we believe that the design placement of the sensor network is likely to be interesting to study, especially for a small sensor network. At last, we found that the tracking problem is more challenging if there are multiple objects in the sensor field. We then present a heuristic algorithm that works well in simulations.

In our report, we always assume the sizes of the objects are small. So we consider objects as points on the 2D plane. However, the sizes of objects are often not negligibly small. How to take the sizes of the objects into account needs to be studied. In the indoor environment, it is more challenging to take the structural objects (walls, floors, ceilings, etc.) into account.

Also in our report, we always assume the sensors are static. In practice the sensors can move in the sensor field. How to estimate the positions of the objects while simultaneously estimate the motions of the sensors using multi-path distance measures is a challenging problem. In the object tracking by multiple-sensor sensor network problem, we proposed an estimation scheme without taken the motion of the object(s) into account. A Kalman filter like estimation scheme, which takes the motion into account, could improve the estimation accuracy meanwhile reduces the computational and communication requirements.

We have discussed the object tracking problem in the high SNR (per transmitter receiver pair) regime where the the multipath distance measures are reasonably accurate. In the low SNR regime where multipath distance measures cannot be extracted, the reliable object tracking problem remains open. We believe that the estimation accuracy will depend primarily on the total SNR.

As mentioned in chapter 1, the communication channel(s) in a sensor network is largely determined by the geometry and the positions of the objects in the sensor field. With accurate position information, we can possibly accurately estimate and predict the communication channels. And the relation between channel capacity and the tracking accuracy is a very interesting problem.

Bibliography

[1] Sachin Adlakha Randolph Moses A. Savvides, Wendy Garber and M. B. Strivastava. On the error characteristics of multihop node localization in ad-hoc sensor networks. In IPSN, 2003.

[2] Y.; Estrada R. Arnold, J.; Bar-Shalom and R. Mucci. Target parameter estimation using measurements acquired with a small number of sensors. In Oceanic Engineering, IEEE Journal of , Volume: 8 , Issue: 3 ,, Jul 1983.

[3] G. Baltes, R.; van Keuk. Tracking multiple manoeuvering targets in a network of passive radars. In Radar Conference, 1995., Record of the IEEE 1995 International , 8-11 May 1995.

[4] Srdan Capkun, Maher Hamdi, and Jean-Pierre Hubaux. GPS-free positioning in mobile ad-hoc networks. In HICSS, 2001.

[5] Victor S. Chernyak. Fundamentals of Multisite Radar Systems. Gordon and Breach Science Publishers, 1998.

[6] Jan Beutel Chris Savarese and Koen Langendoen. Robust positioning algorithms for distributed ad-hoc wireless sensor networks. In USENIX Annual Technical Conference, 2001.

[7] Jan Beutel Chris Savarese and Jan Rabaey. Locationing in distributed ad-hoc wireless sensor networks. In ICASSP , 2001.

[8] Thomas M. Cover and Joy A. Thomas. Elements of Inofrmation Theory. John Wiley and Sons Inc., New York, 1991.

[9] K. Ramachandram R. Brodersen D. Tse, A. Sahai. UWB tracking project, NSF progress report 2. June 25, 2003.

[10] Y. Dersan, A.; Tanik. Passive radar localization by time difference of arrival. In MILCOM, Volume :2, 2002.

[11] Ralph Deutsch. Estimation theory. Prentice Hall, 1965.

[12] Yin Chengyou; Xu Shanjia; Wang Dongjin;. Location accuracy of multistatic radars (trn) based on ranging information. In Radar, 1996. Proceedings., CIE International Conference of , 8-10 Oct. 1996.

[13] Lars Elden and Linde Wittmeyer-Koch. Theory and Applications of Numerical Analysis. Academic Press, London and New York, 1973.

[14] David Forsyth and Jean Ponce. Computer Vision, A Modern Approach. Prentice Hall, New Jersey, 2003.

[15] Lewis Girod and Deborah Estrin. Robust range estimation using acoustic and multimodal sensing. In where, 2001.

[16] Piyush Gupta and P. R. Kumar. The capacity of wireless networks. IEEE Transactions on Information Theory, IT-46:388–404, March 2000.

[17] Allen Hatcher. Algebraic Topology. Cambridge University Press, 2002.

[18] Kenneth Hoffman and Ray Kunze. Linear Algebra. Prentice Hall, 1992.

[19] Yogesh Sankarasubramaniam Ian F. Akyildiz, Weilian Su and Erdal Cayirci. A survey on sensor networks. IEEE Communications Magazine, 40:102 –114, Issue: 8 , Aug. 2002.

[20] G. Boriello J. Hightower and R. Want. Spoton: An indoor 3d location sensing technology based on rf signal strength. In University of Washington CSE Report 2000-0202, Feb 2000.

[21] Leonidas J. Guibas Jaewon Shin and Feng Zhao. A distributed algorithm for managing multi-target identities in wireless ad-hoc sensor networks. In IPSN, 2003.

[22] Gaurav Sukhatme Krishna Chintalapudi, Ramesh Govindan and Amit Dhariwal. Ad-hoc localization using ranging and sectoring. In Infocom, 2004.

[23] Kristofer S. J. pister Lance Doherty and Laurent El Ghaoui. Convex position estimation in wireless sensor networks. In IEEE Infocom, volume 3, pages 1655 –1663, 2001.

[24] Koen Langendoen and Niels Reijers. Distributed localization in wireless sensor networks: a quantitative comparison. In Computer Networks,Volume 43 , Issue 4 , 2003.

[25] Alberto Leon-Garcia. Probability and Random Processes for Electrical Engineering. Addison-Wesley Publishing Company, 1994.

[26] John McCorkle. A tutorial on ultrawideband technology. In IEEE P802.15 Working Group for Wireless Personal Area Networks (WPANS), 2000.

[27] Randolph L. Moses and Robert Patterson. Self-calibration of sensor networks. In SPIE, 2002.

[28] D. Niculescu and B. Nath. Ad hoc positioning system (aps). In In Proceedings of Globecom 2001, 2001.

[29] John Heidemann Nirupama Bulusu and Deborah Estrin. Gps-less low cost outdoor localization for very small devices. In IEEE Person. Commun., 2000.

[30] Thomas Marzetta Petre Stoica. Parameter estimation problems with singular information matrices. In IEEE Trans on Signal Processing, volume 40, 2001.

[31] G.M. Phillips and P.J. Taylor. Numerical Analysis. Academic Press, London and San Diego, 1990.

[32] Howard Shrobe Radhika Nagpa and Jonathan Bachrach. Organizing a global coordinate system from local information on an ad hoc sensor network. In IPSN, 2003.

[33] Dushyanth. Krishnamurthy Randolph L. Moses and Robert Patterson. A self-localization method for wireless sensor networks. In Eurasip, 2003.

[34] M.J.D. Rendas and J.M.F. Moura. Cramer-rao bound for location systems in multipath environments. Signal Processing, IEEE Transactions on, 39:2593–2610, 1991.

[35] A. Savvides and M. B. Strivastava. Distributed fine-grained localization in ad-hoc networks of sensors. submitted to IEEE transaction on Mobile Computing.

[36] S.T. Smith. Intrinsic cramer-rao bounds and subspace estimation accuracy. In Sensor Array and Multichannel Signal Processing Workshop. 2000. Proceedings of the 2000 IEEE, pages 489 –493, 2000.

[37] D.K.P.; Hongbo Sun; Yilong Lu; Weixian Liu Tan. Feasibility analysis of gsm signal for passive radar. In Radar, 2003. Proceedings. 2003 IEEE Conference on, 5-8 May 2003.

[38] E.; Winton, P.; Hammerle. High resolution position estimation using partial pulses. In Electronics Letters , Volume: 36 , Issue: 10 ,, 11 May 2000.

[39] Hongyuan Zha Xiang Ji. Sensor positioning in wireless ad-hoc sensor networks using multidimensional scaling. In Infocom, 2004.

Appendix A

Uniqueness of the positions of a point set

If a point set S is assigned an anchor-free coordinate system and N points in this point set are anchors (with known global positions). As mentioned in **Section 1.1.2** . In general, the point set can be fixed into the anchored coordinate system if $N \geq 3$.

In Fig.A.1 we have a simple example. S, consisting of 7 points, is point set with an anchor-free coordinate system. Both red set and blue set have the same anchor-free coordinate system as shown in Fig.A.1. N black points are anchors with known global positions. If $N = 0, 1$, then as shown in Fig.A.1(a),and (b), there could be infinitely many solutions for the positions of the points, if $N = 2$, there could be 2 possible solutions as in (c). Meanwhile, if $N \geq 3$ as shown in Fig.A.1(d), there is a unique solution for the positions of the point set S.

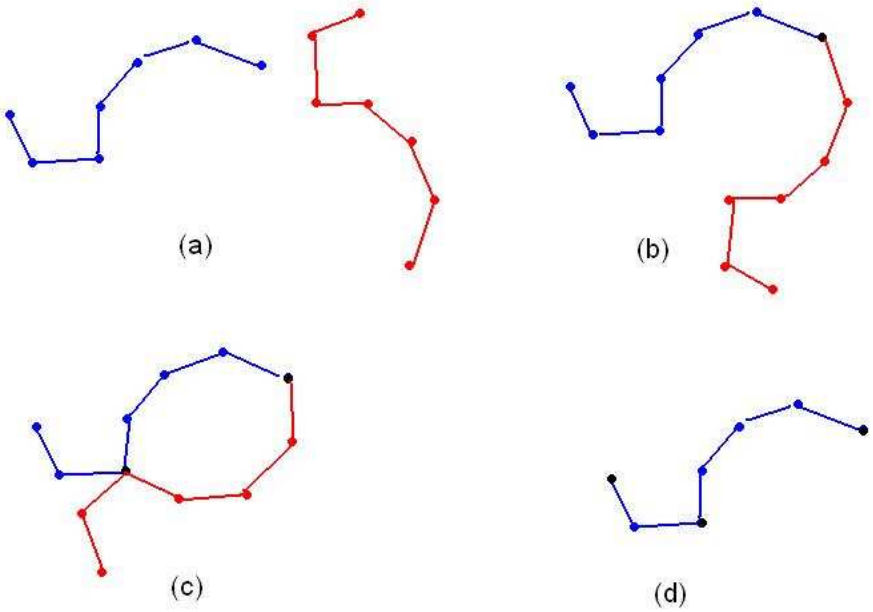


Figure A.1: An illustration of uniqueness of the positions of a point set (a) 0 points has known positions. (b) 1 point has known positions. (c) 2 points has known positions. (d) 3 points has known positions.

Appendix B

Proof of Eqn.2.25

In Eqn.2.25 in **Section 2.2.1** we used the following results.

Theorem B.1. *For a positive definite $N \times N$ matrix J.*

$$J = \begin{pmatrix} A & B \\ B^T & C \end{pmatrix} \tag{B.1}$$

Where A is an $M \times M$ symmetric matrix, C is an $N-M \times N-M$ symmetric matrix and B is an $M \times N-M$ matrix. If we write

$$J^{-1} = \begin{pmatrix} A' & B' \\ B'^T & C' \end{pmatrix} \tag{B.2}$$

Where A and A' have the same size, B and B' have the same size, so do C and C'.

$\boxed{C' - C^{-1} \text{ is positive semi-definite}}$

First we need several lemmas.

Lemma B.1. *A is positive definite.*

Proof: $\forall \vec{x} \in R^M, x \neq 0$. Let $\vec{y} = (\vec{x}, \vec{0})^T$, where $\vec{0}$ is the $1 \times (N-M)$ all 0 vector, $\vec{y}$ is an N dimensional vector. Then $\vec{x}^T A \vec{x} = \vec{y}^T J \vec{y} > 0$.

The last inequality is true because J is positive definite,and $\vec{y} \neq 0$. $\vec{x}$ is arbitrary, so A is positive definite. □

Similarly C is positive definite, and thus A, C are nonsingular.

Lemma B.2. *$A - BC^{-1}B^T$ is positive definite.*

Proof: First notice that for a positive definite matrix J, J can be written as $J_H^T J_H$, where J_H is an $N \times N$ non-singular matrix. Write $J_H = (S \;\; R)$, where S is an $N \times M$, R is an $N \times (N - M)$ matrix. Then

$$A = S^T S; \quad B = S^T R; \quad C = R^T R \tag{B.3}$$

C is nonsingular, so R has full rank $N - M$. The singular value decomposition of R is $R = U\Lambda V$, where U is an $N \times N$ matrix, $U^T U = UU^T = I$,V is an $(N-M) \times (N-M)$ matrix, $V^T V = VV^T = I$, and Λ is an $N \times (N - M)$ matrix.

$$\Lambda = \begin{pmatrix} diag(\lambda_1, ...\lambda_{N-M}) \\ 0_{M\times(N-M)} \end{pmatrix} \tag{B.4}$$

$\lambda_i > 0$ because R has full rank $N - M$. Now:

$$\begin{aligned} &A - BC^{-1}B^T = S^T S - S^T R(R^T R)^{-1}R^T S = S^T(I - R(R^T R)^{-1}R^T)S = \\ &S^T(I - (U\Lambda V)((U\Lambda V)^T(U\Lambda V))^{-1}(U\Lambda V)^T)S = S^T(I - U\Lambda V(V^T\Lambda^T\Lambda V)^{-1}V^T\Lambda^T U^T)S = \\ &S^T(I - U\Lambda VV^T(\Lambda^T\Lambda)^{-1}V^T V\Lambda^T U^T)S = S^T(I - U\Lambda(\Lambda^T\Lambda)^{-1}\Lambda^T U^T)S = \\ &S^T U(I - \Lambda(\Lambda^T\Lambda)^{-1}\Lambda^T)U^T S = S^T U\Delta U^T S \end{aligned} \tag{B.5}$$

Where $\Delta = diag(\delta_1, \delta_2, ...\delta_N)$, where $\delta_i = 0, i = 1, 2, ..., N - M$ and $\delta_i = 1, N - M < i \leq N$. Obviously $A - BC^{-1}B^T$ is positive semi-definite. Suppose $\exists \vec{x} \in R^M, \vec{x} \neq 0$, but $\vec{x}^T S^T U\Delta U^T S\vec{x} = 0$. Then we have $U^T S\vec{x} = (y_1, y_2, ..., y_N)^T = \vec{y}$ and $y_{N-M+1}, ...y_N$ all equal to 0. Now $S\vec{x} = U\vec{y}$ and from the fact that $y_{N-M+1}, ...y_N$ all equal to 0, we have: $\Lambda(\Lambda^T\Lambda)^{-1}\Lambda^T\vec{y} = \vec{y}$. Write $\vec{z} = V^T(\Lambda^T\Lambda)^{-1}\Lambda^T\vec{y}$, then $S\vec{x} = U\vec{y} = U\Lambda V\vec{z} = R\vec{z}$,where $\vec{x} \neq 0$. This contradicts to the fact that $(S \;\; R)$ is full rank. □.

Similarly $C - B^T A^{-1}B$ is positive definite, and thus both are full rank.

Lemma B.3. $(C - B^T A^{-1}B)^{-1} = C^{-1}B^T(A - BC^{-1}B^T)^{-1}BC^{-1} + C^{-1}$

Proof. : Notice that both A and $(A - BC^{-1}B^T)$ are full rank, then,

$$
\begin{aligned}
&(C^{-1}B^T(A - BC^{-1}B^T)^{-1}BC^{-1} + C^{-1})(C - B^TA^{-1}B) \\
&= I + C^{-1}B^T(A - BC^{-1}B^T)^{-1}B - C^{-1}B^TA^{-1}B - C^{-1}B^T(A - BC^{-1}B^T)^{-1}BC^{-1}B^TA^{-1}B \\
&= I + C^{-1}B^T((A - BC^{-1}B^T)^{-1} - A^{-1} - (A - BC^{-1}B^T)^{-1}BC^{-1}B^TA^{-1})B \\
&= I + C^{-1}B^T((A - BC^{-1}B^T)^{-1}(A - BC^{-1}B^T)A^{-1} - A^{-1})B = I
\end{aligned}
$$

□

Lemma B.4. J^{-1} *If we write*

$$J = \begin{pmatrix} A & B \\ B^T & C \end{pmatrix} \text{ and } J^{-1} = \begin{pmatrix} A' & B' \\ B'^T & C' \end{pmatrix}$$

Where A and A' have the same size, and so do B and B', C and C'.
Then $C' = (C - B^TA^{-1}B)^{-1}$

Proof. : Given the form of J^{-1}, we have $B^TB' + CC' = I_{(N-M)\times(N-M)}$ and $AB' + BC' = 0$. From the latter equation, we get $B' = -A^{-1}BC'$. Substitute into the first equation, we get: $-B^TA^{-1}BC' + CC' = I_{(N-M)\times(N-M)}$. Notice the dimensions of the matrices all match. Thus we get the desired result. □

Now we can give the proof of Theorem B.1.

Proof. : $C' = (C - B^TA^{-1}B)^{-1}$ following Lemma B.4. Then from Lemma B.3, we know $(C - B^TA^{-1}B)^{-1} = C^{-1}B^T(A - BC^{-1}B^T)^{-1}BC^{-1} + C^{-1}$. Thus
$C' - C^{-1} = C^{-1}B^T(A - BC^{-1}B^T)^{-1}BC^{-1} = C^{-1^T}B^T(A - BC^{-1}B^T)^{-1}BC^{-1}$ The second equality follows since $C^T = C$. So all we need to prove is that $(A - BC^{-1}B^T)^{-1}$ is positive definite. And this is true following LemmaB.2. □

Definition B.1. *Up-left submatrix*
$1 \le n \le m$, the up-left $n \times n$ submatrix of an $m \times m$ matrix A is an $n \times n$ matrix B, s.t. $B(i,j) = A(i,j), \forall 1 \le i \le n, 1 \le j \le n$.

Corollary B.1. *Monotonically increasing matrices*

For a positive definite $N \times N$ matrix matrix J. Let $1 \leq n_1 \leq n_2 \leq ... \leq n_M = N$, let A be the up-left $n_i \times n_i$ submatrix of A. Let, B_i be the up-left $n_1 \times n_1$ submatrix of A_i^{-1}. Then we have:

$$A_i^{-1} = B_1 \leq B_2 \leq B_3 \leq ... \leq B_M \tag{B.6}$$

Proof. : Notice that an up-left submatrix of a positive definite matrix is still positive definite as shown in Lemma B.1. Repeatedly applying Theorem B.1, we get the desired result. □

Appendix C

An Example of M/K Equivalence Class Estimation

In **Section 2.2.2** we proposed the equivalence class estimation problem framework for a class of estimation problems. An M/K equivalence class estimation is an estimation problem in a M dimensional space and the equivalence class manifold is K dimensional. In this appendix, we give an example of 2/1 equivalence class estimation problem. i.e. the parameter space is 2 dimensional, the equivalence class manifold is 1 dimensional. Suppose the parameter $\theta = (x, y) \in R^2$, and the observation is $\sqrt{x^2 + y^2} + \delta$, where $\delta \sim N(0, \sigma^2$. The equivalence classes are $\widetilde{\theta}_t = \{(x, y) | x^2 + y^2 = t\}, t \geq 0$. And except for $\widetilde{\theta}_0$, $\widetilde{\theta}_t$ is a 1 dimensional manifold.

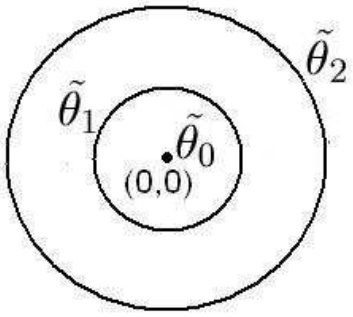


Figure C.1: Equivalence classes are circles centered at $(0, 0)$

Appendix D

There is no subspace in which parameter estimation can be done for the anchor-free coordinate estimation problem

In **Section 2.2.2.3**, we proposed a framework for estimation, the equivalence class estimation problem to deal with the singular Fisher Information Matrix. The intrinsic reason for the singularity of the Fisher Information Matrix is the redundancy of the parameter (vector $\vec{\theta}$) space. For example if a parameter θ_1 is unobservable then the Fisher Information matrix is singular. But we can remove that unobservable parameter from the estimation then we can still analyze the estimation problem in the subspace. However, the redundancy is not always trivial. We can transform an equivalence class estimation problem into a parameter estimation problem in a subspace, if and only if we can find a subspace s.t. the metric in that subspace is the same as the metric for the equivalence classes.

Here we give a simple example to show that there is no subspace in which a parameter estimation can be done for anchor-free coordinate estimation problem. We are going to show that for any subset what we can do parameter estimation, the metric is no loner the same as the metric of the equivalence classes by a simple example.

Assuming we are doing the localization estimation for M points from the measured distances between point pairs.. Suppose the parameter estimation could be properly done, then $\exists S \subseteq R^{2M}$, s.t. $\forall \theta, \alpha, \beta \in R^{2M}$, $\exists \theta^* \in \widetilde{\theta} \bigcap S$,$\exists \beta^* \in \widetilde{\beta} \bigcap S$,$\exists \alpha^* \in \widetilde{\alpha} \bigcap S$, and $d(\alpha^*, \beta^*) = d(\widetilde{\alpha}, \widetilde{\beta})$,$d(\alpha^*, \theta^*) = d(\widetilde{\alpha}, \widetilde{\theta})$,$d(\theta^*, \beta^*) = d(\widetilde{\theta}, \widetilde{\beta})$

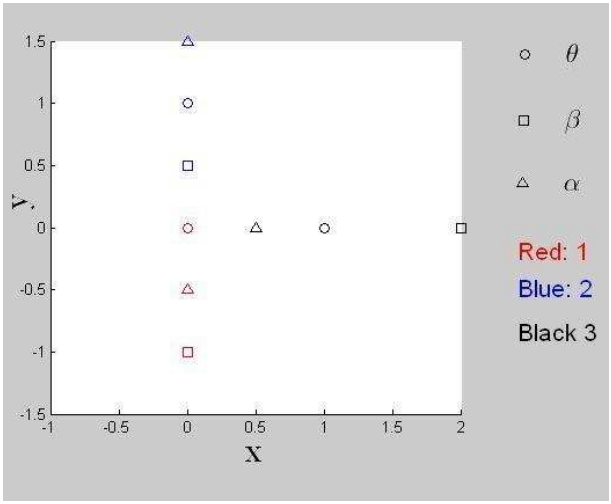


Figure D.1: Counter example
θ_1 Red ◯,θ_2 Blue ◯,θ_3 Black ◯
β_1 Red □,β_2 Blue □,β_3 Black □
α_1 Red △,α_2 Blue △,α_3 Black △

Let $M = 3$. First, without loss of generality, we assume $\theta^* = ((0,0),(0,1),(1,0))$, because we can always translate and rotate S to make any given point on it, without altering the metric on S.

Let $\beta = ((0,-1),(0,0.5),(2,0))$,$\alpha = ((0,-0.5),(0,1.5),(0.5,0))$.

From the optimization problem in Eqn. 2.83, we know that the only $\beta^* \in R^6$, s.t $d(\theta^*,\beta^*) = d(\widetilde{\theta},\widetilde{\beta})$, is $((-0.602,-0.179),(-0.025,1.205),(1.628,-0.025))$, the only $\alpha^* \in R^6$, s.t $d(\theta^*,\alpha^*) = d(\widetilde{\theta},\widetilde{\alpha})$, is $((0.259,-0.513),(0.038,1.474),(0.701,0.038))$. But the only $\alpha^{**} \in R^6$, s.t $d(\beta^*,\alpha^{**}) = d(\widetilde{\beta},\widetilde{\alpha})$, is $((0.125,-0.490),(0.224,1.506),(0.649,-0.016))$.

$\alpha^* \neq \alpha^{**}$, contradiction!

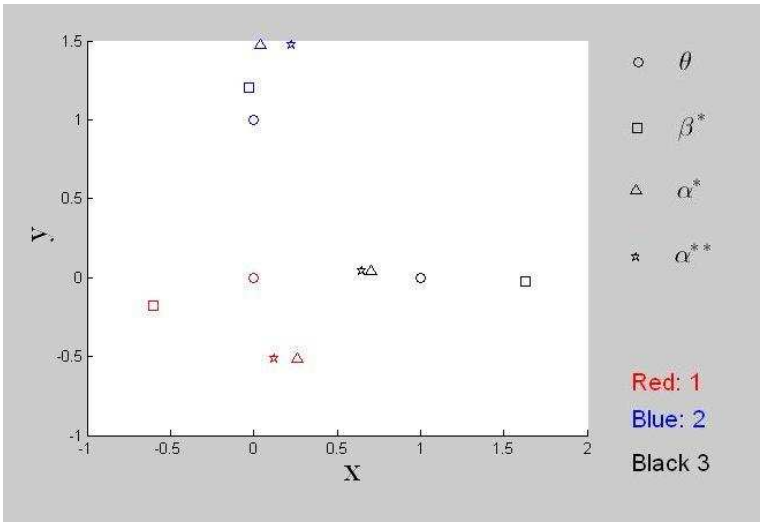


Figure D.2: Counter example
θ_1 Red $\bigcirc$,θ_2 Blue $\bigcirc$,θ_3 Black $\bigcirc$
β_1^* Red $\square$,β_2^* Blue $\square$,β_3^* Black $\square$
α_1^* Red $\triangle$,α_2^* Blue $\triangle$,α_3^* Black $\triangle$
α_1^{**} Red $\star$,α_2^{**} Blue $\star$,α_3^{**} Black $\star$

Appendix E

Non-convexity of Eqn.2.65

The anchored free localization problem can be treated as an optimization problem as in **Section2.3.1**. However the optimization problem is not convex thus traditional optimization schemes are unlikely to work well.

Here we give a simple example to show the non-convexity of 2.65. Given 3 points on the plane, with distance measurements $d_{i,j} = 2$, $i, j \in \{1, 2, 3\}$. If we fix the positions of point 1 and point 2 at $(-1, 0), (1, 0)$ respectively. The following figure shows $\sum_{i,j}(\hat{d}_{i,j} - d_{i,j})^2$ for the different positions of point 3,(x, y), it is obviously non-convex. Where $\hat{d}_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$.

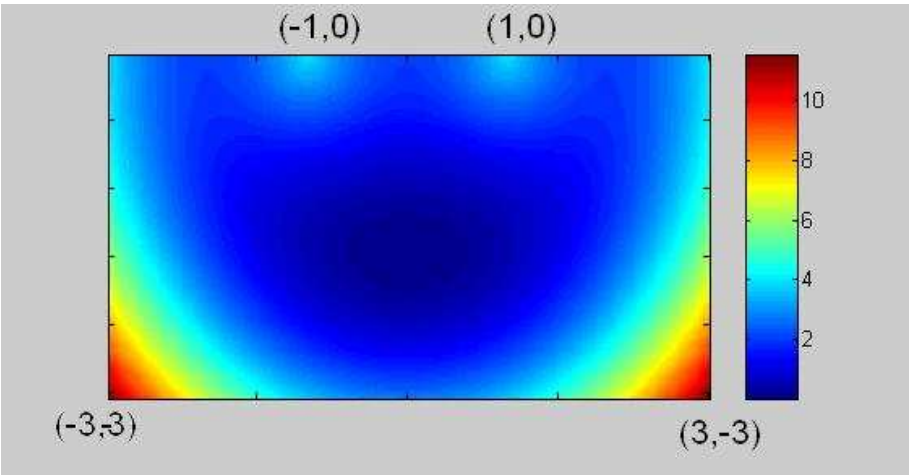


Figure E.1: Values of Eqn.2.65 in the region $[-3, 3] \times [-3, 0$

Appendix F

Multiple local minimum of Eqn.2.69 of variable d_{12}

As shown in **Section 2.3.1.1**, in the first step of our 2-step anchored localization scheme, we can optimize the distance of d_{12}, the distance between the base points in an anchor-free coordinate. Here we give a simple example to show that using our algorithm in the first step of the coordinate estimation inside a cell, the optimization problem in Eqn.2.69 of variable d_{12} has multiple minimum points. Suppose we have 4 points with noisy direct path distances measure: $d_{12} = 3, d_{13} = 4, d_{14} = \sqrt{2}, d_{23} = \sqrt{2}, d_{24} = 1, d_{34} = 0.1$. Then we let d_{12} take value from 0 to 5 and for each value of d_{12}, $p_2 = (d_{12}, 0)$ we estimate the positions of p_3 and p_3. Then we calculate the following function: $E = \sum_{i,j \in C} (\hat{d}_{i,j} - d_{i,j})^2$.

Where $\hat{d}_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$ and $p_i = (x_I, y_i)$. Clearly, E has multiple minimum points.

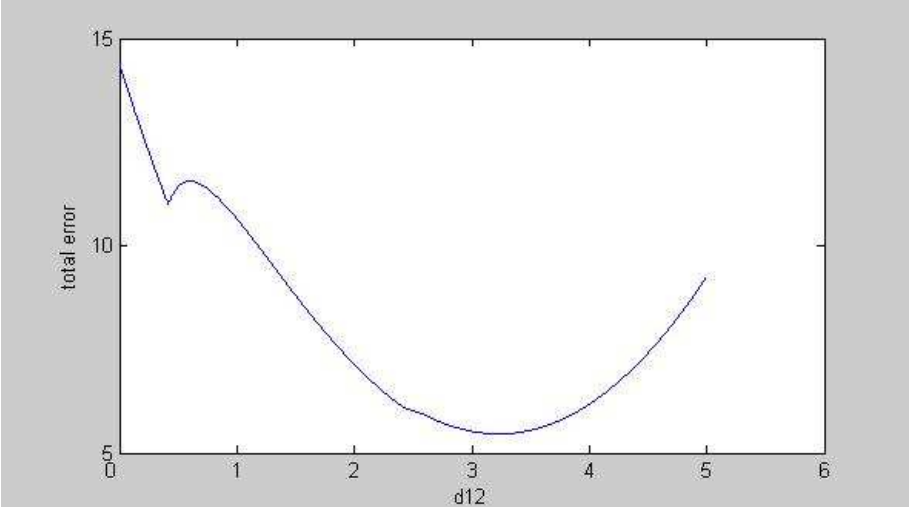


Figure F.1: Total squared error E vs d_{12}

Appendix G

The bias of the ML anchor-free coordinate estimation

In **Section 2.3.1** we proposed an anchored localization scheme. From the simulation results, we can observe that our estimation scheme is usually biased. In this appendix, we are going to give a rigid proof on the bias of a particular scenario. Suppose we know the anchored positions of two points P_1, P_2, P_3 with position $(0,0), (d,0), (a,b)$ respectively, and a point P with unknown position (x,y), WLOG, $y > 0$. The setup is shown in Fig.G.1. We have the measures of the distances from P to point P_i, l_i, where $l_i = d_i + \epsilon_i$, where ϵ_i are iid $\sim N(0, \sigma^2)$. Suppose we are about to estimate the position of P using the scheme mentioned in the anchor-free coordinate estimation. Let P_3 be the reference point. Then the position estimation of (x,y) is $l_1 cos(\alpha), l_1 sin(\alpha)$, where $cos(\alpha) = \frac{d^2+l_1^2-l_2^2}{2dl_1}$. We are going to argue that $\hat{y} = l_1 sin(\alpha)$ is biased. In fact $y > 0$ implies $E(\hat{y} - y) \leq 0$.

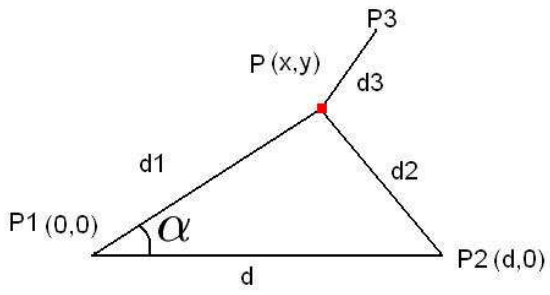


Figure G.1: Local coordinate estimation

From the estimation scheme we know that $sin(\alpha) = \sqrt{1 - cos(\alpha)^2} = \sqrt{1 - (\frac{l_1^2 + d^2 - l_2^2}{2dl_1})^2}$,

so

$$\hat{y} = l_1\sqrt{1 - (\frac{l_1^2 + d^2 - l_2^2}{2dl_1})^2} = \sqrt{\frac{4l_1^2d^2 - (l_1^2 + d^2 - l_2^2)^2}{4d^2}} \tag{G.1}$$

$l_1 = d_1 + \epsilon_1, l_2 = d_2 + \epsilon_2$. We have :

$$\hat{y} = \sqrt{\frac{4d_1^2d^2 - (d_1^2 + d^2 - d_2^2)^2}{4d^2} + e} = \sqrt{y^2 + e} \tag{G.2}$$

$$e = \frac{4(\epsilon_1^2 + 2\epsilon_1 d_1)d^2 + 2(2d_1\epsilon_1 + \epsilon_1^2 - 2d_2\epsilon_2 - \epsilon_2^2)(d_1^2 + d^2 - d_2^2) - (2d_1\epsilon_1 + \epsilon_1^2 - 2d_2\epsilon_2 - \epsilon_2^2)^2}{4d^2}$$

And notice that

$$\sqrt{A + e} = \sqrt{A} + \frac{e}{2\sqrt{A}} - \frac{e^2}{8\sqrt{A^3}} + O(e^3) \tag{G.3}$$

Substitute A with y^2, and take the expectation we have

$$E(\hat{y}) = y + E(\frac{e}{2y}) - E(\frac{e^2}{8y^3}) + E(O(e^3)) = y + \frac{E(e)}{2y} - \frac{E(e^2)}{8y^3} + E(O(e^3)) \tag{G.4}$$

Omit the terms with order 3 or higher for ϵ_i and notice that $E(\epsilon_2^{K_2}\epsilon_1^{K_1}) = 0, K_1 + K_2$ is odd. We have:

$$\begin{aligned} E(e) &= E(\frac{4\epsilon_1^2d^2 + 2(\epsilon_1^2 - \epsilon_2^2)(d_1^2 + d^2 - d_2^2) - 4d_1^2\epsilon_1^2 - 4d_2^2\epsilon_2^2}{4d^2}) = \frac{(d^2 - d_1^2 - d_2^2)\sigma^2}{d^2} \\ E(e^2) &= E((\frac{d_1(d_1^2 + 3d^2 - d_2^2)}{d^2})^2e_1^2 + (\frac{d_2(d_1^2 + d^2 - d_2^2)}{d^2})^2e_2^2) \\ &= \sigma^2\frac{d_1^2(d_1^2 + 3d^2 - d_2^2)^2 + d_2^2(d_1^2 + d^2 - d_2^2)^2}{d^4} \end{aligned} \tag{G.5}$$

So

$$E(\hat{y} - y) = E(\frac{e}{2y}) - E(\frac{e^2}{8y^3}) = \frac{4E(e)y^2 - E(e^2)}{8y^3} \tag{G.6}$$

Now we are going to prove that the above value is smaller than 0.

Case 1: $d^2 - d_1^2 - d_2^2 < 0$, then $E(e) < 0$, so $E(\hat{y} - y) < 0$.

Case 2: $d^2 - d_1^2 - d_2^2 \geq 0$, it is enough to prove that $4E(e)y^2 - E(e^2) < 0$, equivalent to prove

$$(d^2 - d_1^2 - d_2^2)(4d_1^2d^2 - (d_1^2 + d^2 - d_2^2)^2) - d_1^2(d_1^2 + 3d^2 - d_2^2)^2 - d_2^2(d_1^2 + d^2 - d_2^2)^2 < 0 \tag{G.7}$$

Let $A = (d_1^2 + d^2 - d_2^2)$. Then it is equivalent to

$$(A - 2d_1^2)(4d_1^2d^2 - A^2) - d_1^2(A + 2d^2)^2 - d_2^2A^2 < 0$$

$$-A^3 + A^2d_1^2 - 8d_1^4d^2 - 4d_1^2d^4 - d_2^2A^2 < 0 \tag{G.8}$$

$$A^2(d_1^2 - A) - 8d_1^4d^2 - 4d_1^2d^4 - d_2^2A^2 < 0 \tag{G.9}$$

The last inequality is true because $d_1^2 - A = d_2^2 - d^2 < 0$. Thus we proved that with $y > 0$, $E(\hat{y} - y) < 0$. □

Appendix H

The optimization problem in Eqn. 2.83

In **Section 2.3**, we proposed an anchored localization scheme. In our scheme, we repeatedly need to solve the following problem: N points with positions $z_i = (a_i, b_i)^T, i = 1, ...N$, another N points with positions $z'_i = (c_i, d_i)^T, i = 1, ...N$. The following optimization problem is stated.

$$E = argmin_{\alpha,T} \sum_{i=1}^{N} \|R(\alpha)z_i + T - z'_i\|_2^2 \tag{H.1}$$

Where

$$R(\alpha) = \begin{pmatrix} cos(\alpha) & -sin(\alpha) \\ sin(\alpha) & cos(\alpha) \end{pmatrix}, T = \begin{pmatrix} t_x \\ t_y \end{pmatrix} \tag{H.2}$$

Eqn. H.1 could be written as following

$$E = argmin_{\alpha,T} \sum_{i=1}^{N} (cos(\alpha)a_i - sin(\alpha)b_i - c_i + t_1)^2 + (cos(\alpha)b_i + sin(\alpha)a_i - d_i + t_2)^2 \tag{H.3}$$

This is an unconstrained optimization problem which means that the we only need to get the derivative of Eqn.H.3, set the derivative to 0, and solve the equations.

We get the following results:

$$\frac{dE}{dt_1} = \sum_{i=1}^{N}(cos(\alpha)a_i - sin(\alpha)b_i - c_i + t_1) = 0 \tag{H.4}$$

$$\text{Thus } t_1 = -\frac{1}{N}\sum_{i=1}^{N}(cos(\alpha)a_i - sin(\alpha)b_i - c_i) \tag{H.5}$$

$$\text{Similarly } t_2 = -\frac{1}{N}\sum_{i=1}^{N}(cos(\alpha)b_i + sin(\alpha)a_i - d_i) \tag{H.6}$$

$$\begin{aligned}\frac{dE}{d\alpha} = &\sum_{i=1}^{N}(cos(\alpha)a_i - sin(\alpha)b_i - c_i + t_1)(-cos(\alpha)b_i - sin(\alpha)a_i) + \\ &\sum_{i=1}^{N}(cos(\alpha)b_i + sin(\alpha)a_i - d_i + t_2)(cos(\alpha)a_i - sin(\alpha)b_i) = 0\end{aligned} \tag{H.7}$$

It turns out that the Eqn. H.7 can be simplified by grouping $cos(\alpha)$ and $sin(\alpha)$ terms:

$$Acos(\alpha) + Bsin(\alpha) = 0 \tag{H.8}$$

$$\alpha_1 = arctan(-\frac{A}{B}) \tag{H.9}$$

$$\alpha_2 = arctan(-\frac{A}{B}) + \pi \tag{H.10}$$

$$\text{Where } A = \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{N}(a_i d_k - b_i c_k) + \sum_{i=1}^{N}(-a_i d_i + b_i c_i) \tag{H.11}$$

$$B = \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{N}(-a_i c_k - b_i d_k) + \sum_{i=1}^{N}(a_i c_i + b_i d_i) \tag{H.12}$$

Now we can substitute α_i into Eqn. H.5 and Eqn. H.6 to get $t_{1i}, t_{2i}, i = 1, 2$. Then substitute $\alpha_i, t_{1i}, t_{2i}, i = 1, 2$ into Eqn. H.3, pick the i to minimize Eqn. H.3 . Thus we have solved the optimization problem. □

Appendix I

STSR Object Tracking of Constant Acceleration Motion

In **Section 3.1** we studied the object tracking problem. In a single transmitter single receiver sensor network, the motion can possibly be estimated only if there is a proper motion model. In Section 3.1 we showed that if the object is moving with a constant velocity, then theoretically the object can be tracked. In this appendix, we prove that if the object moves with a constant acceleration, the motion is also trackable.

Theory

If an object A is moving in a sensor network field with constant acceleration, then there are 6 parameters to be estimated. (x, y, u, v, a, b), where (x, y) is the starting point of the motion, (u, v) is the velocity of the object at (x, y),(a, b) is the constant acceleration. At time t, the object is at point

$$(x_t, y_t) = (x + tu + \frac{t^2 a}{2}, y + tv + \frac{t^2 b}{2}) \tag{I.1}$$

Thus the 6 parameters uniquely determine the motion of the object. At time t, the multi-path distance is d_t. The analysis here is similar to the analysis for the constant velocity case.

$$\frac{{x_t}^2}{p_t^2} + \frac{{y_t}^2}{q_t^2} = 1 \tag{I.2}$$

Where $p_t = \frac{d_t}{2}, q_t = \sqrt{(\frac{d_t}{2})^2 - 1}$, substitute Eqn.I.1 into the above equation. We get :

$$\frac{(x + tu + \frac{t^2 a}{2})^2}{p_t^2} + \frac{(y + tv + \frac{t^2 b}{2})^2}{q_t^2} = 1$$

$$\frac{x^2 + t^2u^2 + \frac{t^4}{4}a^2 + 2txu + t^2ax + t^3ua}{p_t^2} + \frac{y^2 + t^2v^2 + \frac{t^4}{4}b^2 + 2tyv + t^2ax + t^3vb}{q_t^2} = 1$$

Thus we have a linear equation for $(x^2, u^2, a^2, xu, xa, ua, y^2, v^2, b^2, yv, yb, vb)$, 12 unknowns. However, noticing that the coefficients for u^2, xa are always the same, the coefficients for v^2, yb are always the same, thus the linear equation system is not full rank given 12 unknowns. We actually have to combine u^2, xa into a single unknown, and v^2, yb into a single unknown to make the linear equation system full rank. Now we have a linear equation for $(x^2, u^2 + xa, a^2, xu, ua, y^2, v^2 + yb, b^2, yv, vb)$, 10 unknowns. Generally we need 10 linear equations to solve 10 unknowns, so we need 10 multi-path distance measurement to recover the vector $(x^2, u^2 + xa, a^2, xu, ua, y^2, v^2 + yb, b^2, yv, vb)$, then with some ambiguity, we can solve for the motion (x, u, a, y, v, b). The ambiguity comes from the symmetry of the geometry. We now argue that there are 4 possible solutions for (x, u, a, y, v, b) from a single solution of $(x^2, u^2+xa, a^2, xu, ua, y^2, v^2+yb, b^2, yv, vb) = (q_1, q_2, ..., q_{10})$. First, there are 2 possible solutions for x, $\sqrt{q_1}$ and $-\sqrt{q_1}$, as long as x is fixed, $u = \frac{q_4}{x}, a = \frac{q_5}{u}$ are fixed. Same for y, v, b, so we have totally 4 possible solutions for (x, u, a, y, v, b) given one solution from the linear equation system. Similarly we can generalize the analysis to high order motion models.

Stability Analysis

Here we analyze the stability issue of the linear equation system for $(x^2, u^2+xa, a^2, xu, ua, y^2, v^2+yb, b^2, yv, vb)$. Assuming out goal is to estimate the positions of the object at time $0, 1/2, 1$. Knowing these 3 positions, we can fully recover the whole motion of the object given constant acceleration assumption. Then we have $x_0 = x, x_{1/2} = x + \frac{u}{2} + \frac{a}{8}, x_1 = x + u + \frac{a}{2}$, $y_0 = y, y_{1/2} = y + \frac{v}{2} + \frac{b}{8}, y_1 = y + v + \frac{b}{2}$. Noticing the transform matrix is not ill-conditioned. Thus the condition number of the matrix for the

$(x^2, u^2+xa, a^2, xu, ua, y^2, v^2+yb, b^2, yv, vb)$ is a good measure on how good we can estimate the location of the object at time $0, \frac{1}{2}, 1$. Write $\overrightarrow{z} = (x^2, u^2+xa, a^2, xu, ua, y^2, v^2+yb, b^2, yv, vb)^T$. We have the following linear equation system: $A\overrightarrow{z} = \overrightarrow{D}$. Where the k'th element of $\overrightarrow{D}$ is ${p_k}^2{q_k}^2$. The k'th row of matrix A is

$$({q_k}^2, {q_k}^2t^2, \frac{{q_k}^2t^4}{4}, 2{q_k}^2t, {q_k}^2t^3, {p_k}^2, {p_k}^2t^2, \frac{{p_k}^2t^4}{4}, 2{p_k}^2t, {p_k}^2t^3) \tag{I.3}$$

Where ${p_k}^2 = \frac{{d_k}^2}{4}, {q_k}^2 = {p_k}^2 - 1$, and

$d_k = \sqrt{(x + \frac{ku}{N} + \frac{k^2a}{2N^2} - 1)^2 + (y + \frac{kv}{N} + \frac{k^2b}{2N^2})^2} + \sqrt{(x + \frac{ku}{N} + \frac{k^2a}{2N^2} + 1)^2 + (y + \frac{kv}{N} + \frac{k^2b}{2N^2})^2}$

And we have the $\chi(A)$ to measure the stability of the the linear equation system.

As can be seen, $\chi(A)$ is generally huge for the constant acceleration model. Thus a little error in multi-path distance measurements would result in a huge error in the motion estimation.

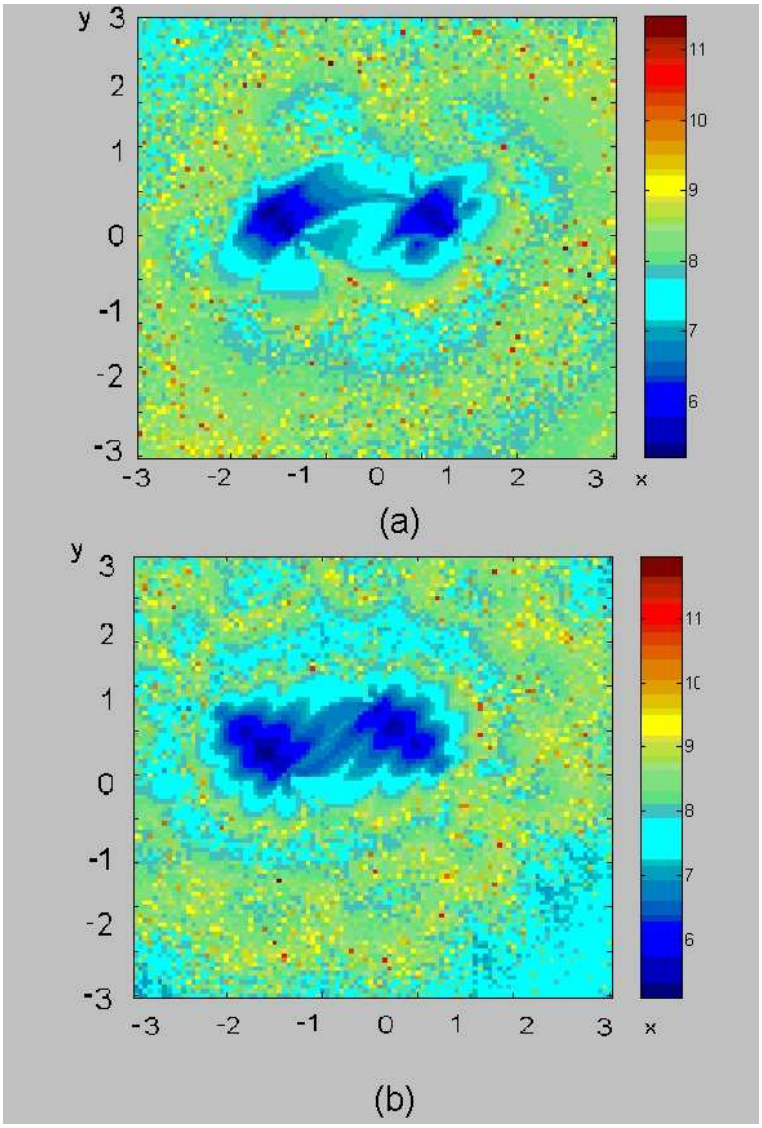


Figure I.1: The value on $(x, y) \in [-3, 3] \times [-3, 3]$ indicates $log_{10}(\chi(A))$, (a) $(u, v) = (0, 0), (a, b) = (1, 1)$;(b) $(u, v) = (1, 0), (a, b) = (0, 1)$

Appendix J

Degenerate Cases for STSR

In the object tracking by a single transmitter single receiver problem, we proposed a simple linear scheme in **Section 3.1.1**. In that scheme we need to compute the pseudo-inverse of a matrix A. A is dependent on the motion of the object. In this appendix we are going to show that A is full rank except 3 degenerate cases. Thus in the 3 degenerate cases we cannot use the simple scheme to track the object.

In the single transmitter single receiver network, the object is assumed to be moving with constant velocity. We have the A matrix defined in Eqn.3.8. In general A is full rank (6) except the following 3 cases. First we need a well known result which can be found in [18]:

Lemma J.1. *dependence of polynomials: $K > N + 1$, then any K polynomials with order not bigger than N are linear dependent.*

In the following analysis, we treat the $k'th$ row of matrix A as a function of k. Again, the motion starts at (x, y), ends at (u, v).

Case (1): The object moves along the perpendicular bisector of the transmitter-receiver pair. As shown in Fig.J.1, the red arrow. $d_k^2 = 2(y_k^2+1) = 2((\frac{k}{N}v+\frac{N-k}{N}y)^2+1)$, so $a_k^2 = (\frac{k}{N}v + \frac{N-k}{N}y)^2 + 1, b_k^2 = (\frac{k}{N}v + \frac{N-k}{N}y)^2$

The $k'th$ row of A is $(b_k{}^2(1 - \frac{k}{N})^2, 2b_k{}^2(1 - \frac{k}{N})\frac{k}{N}, b_k{}^2(\frac{k}{N})^2, a_k{}^2(1 - \frac{k}{N})^2, 2a_k{}^2(1 - \frac{k}{N})\frac{k}{N}, a_k{}^2(\frac{k}{N})^2)$ Notice that a_k, b_k are quadratic function of k, thus the $k'th$ row of A is

a vector of 6 degree 4 polynomials of k. From the lemma we know that A is singular.

Case (2): The object moves along the transmitter-receiver line and the motion is in between the transmitter and the receiver. As shown in Fig. J.1, as the black arrow. $d_k = 2$, so $a_k^2 = 1, b_k^2 = 0$

The $k'th$ row of A is $(0, 0, 0, (1 - \frac{k}{N})^2, 2(1 - \frac{k}{N})\frac{k}{N}, (\frac{k}{N})^2)$, A is obviously singular.

Case (3): The object moves along the transmitter-receiver line. And the motion is outside of the transmitter-receiver line segment. As shown in Fig.J.1, the blue arrow. $d_k = 2x_k = 2(\frac{k}{N}u + \frac{N-k}{N}x)$, so $a_k^2 = (\frac{k}{N}u + \frac{N-k}{N}x)^2, b_k^2 = (\frac{k}{N}u + \frac{N-k}{N}x)^2 - 1$

The $k'th$ row of A is $({b_k}^2(1 - \frac{k}{N})^2, 2{b_k}^2(1 - \frac{k}{N})\frac{k}{N}, {b_k}^2(\frac{k}{N})^2, {a_k}^2(1 - \frac{k}{N})^2, 2{a_k}^2(1 - \frac{k}{N})\frac{k}{N}, {a_k}^2(\frac{k}{N})^2)$ Notice that a_k, b_k are quadratic functions of k, thus the $k'th$ row of A is a vector of 6 degree 4 polynomials of k. From the lemma we know that A is singular.

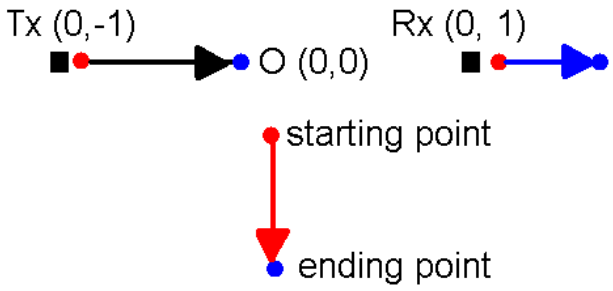


Figure J.1: A matrix defined in Eqn.3.8 is singular if the object moves along the arrows in the figure.

If the object moves along the transmitter-receiver line, **and** it crosses the transmitter or receiver. Then the matrix A for the whole motion is full-rank (6). Without loss of generality, we assume the object first moves inside the line segment of the Tx and Rx, then moves across the receiver, then moves outside of the line segment. Then the matrix A is composed of 2 parts. In the first part, the rows are like what we have in case (2), in the other part, the rows are like those in case (3). Assuming in the first part, we have L measurements, in part 2 we have M measurements. And $M, L \geq 3, M + L = N$. Now we have A as following:

$$A = \begin{pmatrix} 0 & D \\ E & F \end{pmatrix}, \tag{J.1}$$

Where $0, D$ are $L \times 3$ matrices, E, F are $M \times 3$ matrices. A is full-rank (6) if and only if E, D are all full-rank(3). The k'th row of D is $((1-\frac{k}{N})^2, 2(1-\frac{k}{N})\frac{k}{N}, (\frac{k}{N})^2)$, and we know the 3 quadratic functions $(1-z)^2, 2(1-z)z, z^2$ are linearly independent, so as long as we have 3 different k's there ($L \geq 3$), D is full-rank (3).

As for E, similar to case(3) the $k-L$'th row of E is $({b_k}^2(1-\frac{k}{N})^2, 2{b_k}^2(1-\frac{k}{N})\frac{k}{N}, {b_k}^2(\frac{k}{N})^2)$, where $b_k^2 = (\frac{k}{N}u + \frac{N-k}{N}x)^2 - 1$. x, u are the starting and ending $x-position$ for the object respectively. Replace $\frac{k}{N}$ with z. We have the $k-L$'th row of E is $(uz+x(1-z))^2((1-z)^2, 2(1-z)z, z^2)$, those 3 polynomials are linearly independent as long as $x \neq u$. So if we have not less than 3 measurements, the matrix E is full-rank(3).

We just proved that E, D are all of rank 3, we can claim that matrix A is full-rank (6).

Appendix K

Convergence of $\chi(A_N)$

In **Section 3.1.1** we gave a linear scheme for object tracking by a single transmitter single receiver network. If we have N multipath distance measurements, then we have a linear equation system as shown in Eqn.3.9. To solve that equation we directly apply the pseudo-inverse of the matrix A_N as shown in Eqn.3.10. The condition number of A_N is interesting because it tells if the solution in Eqn 3.9 is stable. Here we are going to prove $\chi(A_N)$ converges to a finite real value or goes to infinity.

Lemma K.1. *Convergence of $A_N^T A_N/N$ Write $B_N = \frac{A_N^T A_N}{N}$*

$$lim_{N\to\infty} B_N = B \quad \textit{entry-wisely} \tag{K.1}$$

Where B is a real matrix, all entries of B are bounded.

Proof. : If the starting point is (x, y), ending point is (u, v), let $(x(t), y(t)) = (1 - t)(x, y) + t(u, v)$,
$d(t) = \sqrt{(x(t) - 1)^2 + y(t)^2} + \sqrt{(x(t) + 1)^2 + y(t)^2}$, $a(t) = \frac{d(t)}{2}$, $b(t) = \sqrt{\frac{d(t)}{2}^2 - 1}$

Let $t = \frac{k}{N}$, then the kth row of A_N is
$(b(t)^2(1 - t)^2, 2b(t)^2(1 - t)t, b(t)^2(t)^2, a(t)^2(1 - t)^2, 2a(t)^2(1 - t)t, a(t)^2(t)^2)$ the $(1, 1)$ element of B_N:

$$B_N(1,1) = \frac{1}{N} \sum_{t\in\{0,\frac{1}{N},\frac{2}{N},\dots\frac{N}{N}\}} (b(t)^2(1-t)^2)^2$$

$$lim_{N\to\infty} B_N(1,1) = \int_0^1 (b(t)^2(1-t)^2)^2 dt \tag{K.2}$$

Notice that $(1-t)^2$ and $b(t)^2$ are bounded continuous function in $[0,1]$, the limit exists. Thus we proved the $(1,1)$ element of B_N converges, similarly other entries of B_N converge. So $\frac{A_N^T A_N}{N}$ converges. □

Obviously B is positive semi-definite, thus all eigenvalues of B is not less than 0.

Theorem K.1. *If the minimal eigenvalue of B, $minEig(B) > 0$, then*

$$lim_{N\to\infty}\chi(A_N) = \chi(B) \tag{K.3}$$

On the other hand, if $minEig(B) = 0$, then the condition number goes to infinity.

Proof. : First observe that $\chi(A_N) = \sqrt{cond(A_N^T A_N)} = \sqrt{cond(\frac{A_N^T A_N}{N})} = \sqrt{cond(B_N)}$. And $cond(B) = \frac{maxEig(B)}{minEig(B)}$. And for a finite dimension non-singular matrix B, 6×6 here, if a sequence of matrices B_N converges to B entry-wisely. Then the eigenvalues of B_N all converge to eigenvalues of B.

$$\begin{aligned} lim_{N\to\infty} maxEig(B_N) &= maxEig(B) \\ lim_{N\to\infty} minEig(B_N) &= minEig(B) \end{aligned} \tag{K.4}$$

And the lemma tells that $\frac{A_N^T A_N}{N}$ converges to B entry-wisely.We know $\chi(A_N)$ converges to $\chi(B)$. □

Appendix L

Confusing Paths for STSR

[1]In **Section 3.1**, we assumed that the object moves with a constant velocity. If the multipath distance measures are accurate then we can accurately estimate the motion of the object as shown in **Section 3.1.1**. In this appendix, we are going to show that if the object is allowed to move with an arbitrary velocity, then it is impossible to track the motion of the object by a single transmitter, single receiver network even the multipath distance measures are noiseless.

In a single transmitter, single receiver network, the position of object A is $P(t) = (x(t), y(t)), t \in [0, 1]$. If we only constrain the motion to be continuous (or smooth), then it is impossible to estimate the position of the object. We are going to prove this by showing there exists 2 different motions with the same multi-path distance measurements for $t \in [0, 1]$. This result is summarized in Theorem L.1.

Suppose the transmitter is at $(-1, 0)$, receiver is at $(0, 1)$. Given a point P at $P = (x, y)$, define $d(P) = \sqrt{(x-1)^2 + y^2} + \sqrt{(x+1)^2 + y^2}$, $d(P)$ is the multi-path distance of point P.

Definition L.1. *Good pair: A pair of points (P_1, P_2), s.t. $2 < d(P_1) < d(sP_1 + (1-s)P_2) < d(P_2), \forall s \in (0, 1)$. essentially, if an object moves from P_1 to P_2, the multi-path distance increases monotonically.*

Definition L.2. *path: [17]A path is a continuous map from $[0, 1]$ to R^2.*

[1] The author would like to thank Hansen Bow for his original idea which inspired this work.

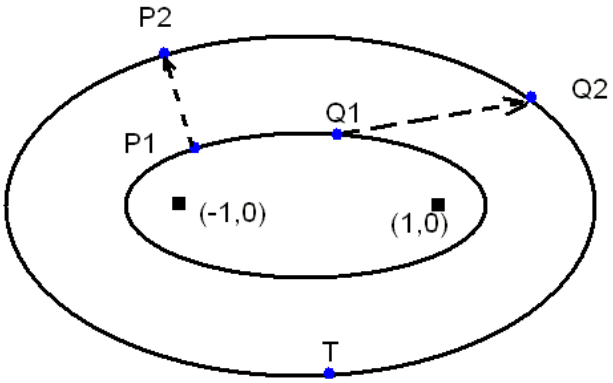


Figure L.1: $(P1, P2)$ is a good pair,$(Q1, Q2)$ is a good pair, too. $(P1, T), (P2, T)$ are not good pairs.

Definition L.3. *Confusing paths Objects A, B moves along the path $P_A(t), P_B(t), t \in [0,1]$. i.e. at time t, A is at point $P_A(t) = (x_A(t), y_A(t))$. If $d(P_A(t)) = d(P_B(t)), \forall t \in [0,1]$, then we call $P_A(t), P_B(t)$ confusing paths.*

Theorem L.1. *Existence of confusing pairs: Given any 4 points P_1, P_2, Q_1, Q_2, s.t. $(P_1, P_2), (Q_1, Q_2)$ are good pairs as shown in Fig.L.1. $d(P_i) = d(Q_i), i = 1, 2$. An object A moves with constant velocity from P_1 to P_2, from time 0 to 1, i.e. $P_A(t) = (1-t)P_1 + tP_2$. Another object B moves along path $P_B(t), t \in [0,1]$, then there exits a path $P_B(t)$ s.t.:*
(1)$P_B(t) = (1 - S(t))Q_1 + S(t)Q_2$, where S is a continuous map from $[0,1]$ to $[0,1]$ and $S(0) = 0, S(1) = 1$, i.e. object B moves straightly and continuously from Q_1 to Q_2.
(2) $d(P_A(t)) = d(P_B(t)), \forall t \in [0,1]$

proof: At time t, $P_A(t) = (x_t, y_t) = (1-t)P_1 + tP_2$,$P_B(t) = (u_t, v_t) = (1 - S(t))Q_1 + S(t)Q_2$ if $d(P_A(t)) = d(P_B(t))$, then we have

$$\frac{u_t^2}{a_t^2} + \frac{v_t^2}{b_t^2} = 1 \tag{L.1}$$

Where $a_t = \frac{d(P_A(t))}{2}, b_t = \sqrt{(\frac{d(P_A(t))}{2})^2 - 1}$, let $Q_1 = (q_{1x}, q_{1y}), Q_2 = (q_{2x}, q_{2y})$, then $u_t = (1 - S(t))q_{1x} + S(t)q_{2x}$,$v_t = (1 - S(t))q_{1y} + S(t)q_{2y}$. Substitute these into Eqn.L.1, we have:

$$\begin{aligned}
&\frac{((1 - S(t))q_{1x} + S(t)q_{2x})^2}{a_t^2} + \frac{((1 - S(t))q_{1y} + S(t)q_{2y})^2}{b_t^2} = 1 \\
&(\frac{(q_{2x} - q_{1x})^2}{a_t^2} + \frac{(q_{2y} - q_{1y})^2}{b_t^2})S(t)^2 + 2(\frac{(q_{2x} - q_{1x})q_{1x}}{a_t^2} + \frac{(q_{2y} - q_{1y})q_{1y}}{b_t^2})S(t) + \\
&(\frac{q_{1x}^2}{a_t^2} + \frac{q_{1y}^2}{b_t^2} - 1) = F(S(t)) = 0
\end{aligned} \tag{L.2}$$

Where F is a quadratic function. Since, $(P_1, P_2), (Q_1, Q_2)$ are good pairs. So $d(Q_1) = d(P_2) > d(P_A(t)) > d(P_1) = d(Q_1), \forall t \in (0,1)$. Thus we have:

$$\begin{aligned}
F(0) &= \frac{q_{1x}^2}{a_t^2} + \frac{q_{1y}^2}{b_t^2} - 1 < 0 \\
F(1) &= \frac{q_{2x}^2}{a_t^2} + \frac{q_{2y}^2}{b_t^2} - 1 > 0
\end{aligned} \tag{L.3}$$

$F(0) < 0 < F(1), \forall t \in (0,1)$, thus there is a unique solution for $S(t)$. We give the explicit expression for $S(t)$:

$$S(t) = \frac{-(\frac{(q_{2x}-q_{1x})q_{1x}}{a_t^2} + \frac{(q_{2y}-q_{1y})q_{1y}}{b_t^2})}{\frac{(q_{2x}-q_{1x})^2}{a_t^2} + \frac{(q_{2y}-q_{1y})^2}{b_t^2}} + \frac{\sqrt{(\frac{(q_{2x}-q_{1x})q_{1x}}{a_t^2} + \frac{(q_{2y}-q_{1y})q_{1y}}{b_t^2})^2 - (\frac{(q_{2x}-q_{1x})^2}{a_t^2} + \frac{(q_{2y}-q_{1y})^2}{b_t^2})(\frac{q_{1x}^2}{a_t^2} + \frac{q_{1y}^2}{b_t^2} - 1)}}{\frac{(q_{2x}-q_{1x})^2}{a_t^2} + \frac{(q_{2y}-q_{1y})^2}{b_t^2}} \quad \text{(L.4)}$$

It can be easily verified that $S(t)$ is a continuous function in $[0,1]$, and furthermore $S'(t), S''(t)$ are continuous in $[0,1]$. To prove the continuousness, we only need to verify all those functions are bounded since a_t, b_t are bounded continuous function of $t \in [0,1]$, and $S(t)$ is an algebraic function of a_t, b_t.

An example of confusing pairs:

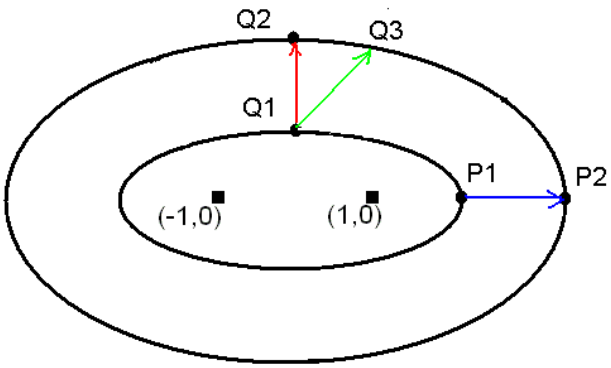


Figure L.2: Confusing paths

As illustrated in Fig.L.2,$P_1 = (2,0), P_2 = (4,0), Q_1 = (0,\sqrt{3}), Q_2 = (0,\sqrt{15}), Q_3 = (2.141, 3.272)$. $(P_1,P_2),(Q_1,Q_2),(Q_1,Q_3)$ are good pairs. And $d(P_1) = d(Q_1), d(P_2) = d(Q_2) = d(Q_3)$, if an objet A moves from P_1 to P_2 with constant velocity from time 0 to 1. At time $t \in [0,1]$, the position of A is $P_A(t) = (1-t)P_1 + tP_2$. From the theorem we just proved, $\exists S_1(t), S_2(t)$. S.t. an object B moves from Q_1 to Q_2, an object C move s from Q_1 to Q_3 from time 0 to 1, and at time t, B is at $(1-S_1(t))Q_1 + S_1(t)Q_2$, C is at $(1-S_2(t))Q_1 + S_2(t)Q_3$, and $d(P_A(t)) = d(P_B(t)) = d(P_C(t))$. Where $S_i(t), i = 1,2$ is smooth.

$S_i(t)$ and $S_i'(t)$ are shown in Fig.L.3. As can be seen, $S_i(t)'s$ are quite close to the path with constant velocity which is in blue.

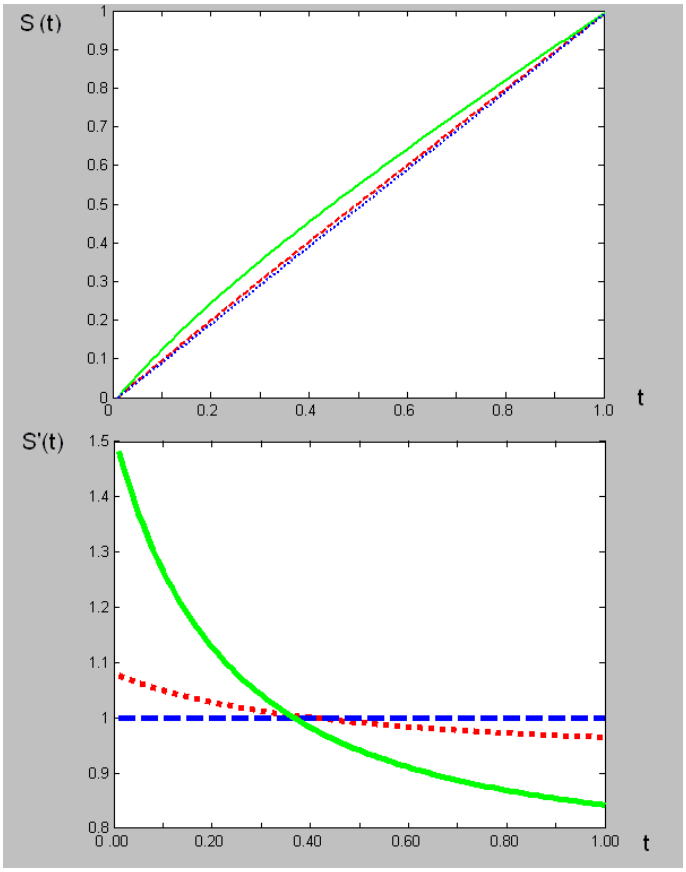


Figure L.3: Blue line is the motion with constant velocity, red curves are the $S_1(t), S_1'(t)$ for a path from Q_1 to Q_2 to confuse with a path from P_1 to P_2 with constant velocity. Green curves are $S_2(t), S_2'(t)$ for a path from Q_1 to Q_3 to confuse with the same path.

Appendix M

Stability of the Cost function

In multiple object tracking problem, the first step is always to associate the multipath distances measures with the objects. In our 2-step scheme, we first discretize the space then search for the local optimal grid points of some function. The cost function as defined in Eqn.3.140 is a natural choice. However, as pointed out in **Section 3.3.2.1**. The cost function $C(x, y)$ defined in is not quite stable. In the sense that if one very *bad* multi-path distance measurement is presented (likely to happen due to the sensor failure), then $C(x, y)$ would be largely affected by that single *bad* measurement. In the contrary, the score function $S(x, y)$ defined in Eqn.3.141 is more stable as shown in Fig.M.3.

Here we give an example. In a 2 transmitter, 2 receiver sensor network as shown in Fig.M.1. The transmitters are located at $(0, 0), (0.5, 0), (1, 0)$, receivers are at $(0, 1), (0.5, 1), (1, 1)$. The object is at $(0.5, 0.5)$. Then all the multi-path distance measurements should $\in [1, \sqrt{2}]$. And we have the cost function $C(x, y)$ in Fig.M.2(a). If the multi-path distance measure from $Tx3$ to $Rx2$ reflected by the object is 1.957 instead of the true one 1.207, then we have the cost function $C(x, y)$ as show in Fig.M.2(b). And the point with the minimum cost function is $(0.4, 0.12)$ instead of the true location of the object $(0.5, 0.5)$.

Meanwhile in Fig.M.3, we plot the score functions $S(x, y)$ for noiseless and noisy case. It can be seen that a single sensor failure does not change the output as dramat-

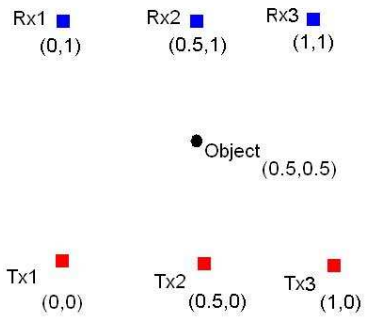


Figure M.1: A simple sensor network

ically as it does on the cost function $C(x, y)$. From the second plots, we can see that if the multi-path distance measure is corrupted by a big noise, then that single false measure would only affect the region around the false ellipse, but would do nothing on other part of the plane.

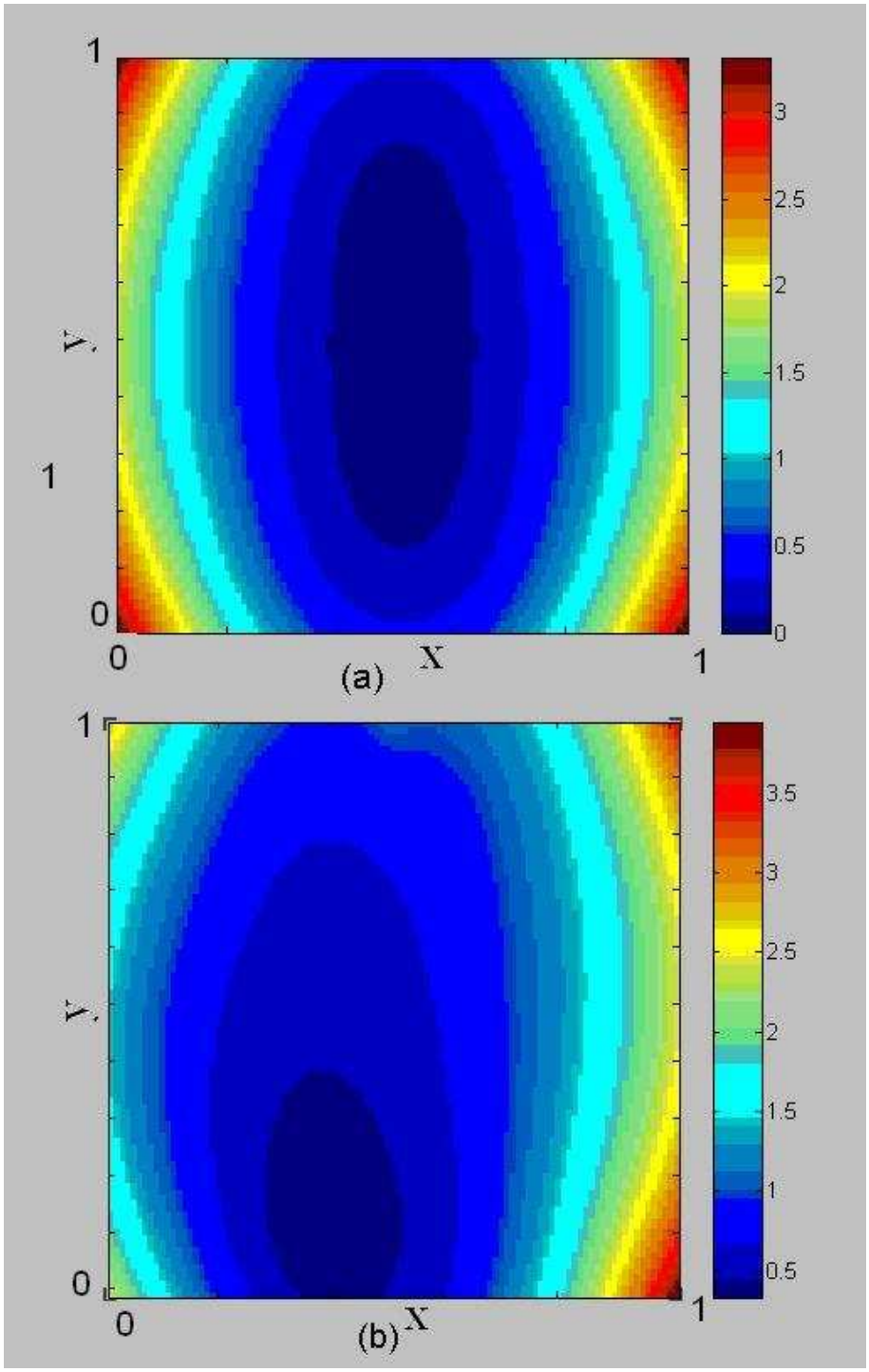


Figure M.2: (a) C(x,y) for true multi-path distances (b) C(x,y) for noisy multi-path distance measures

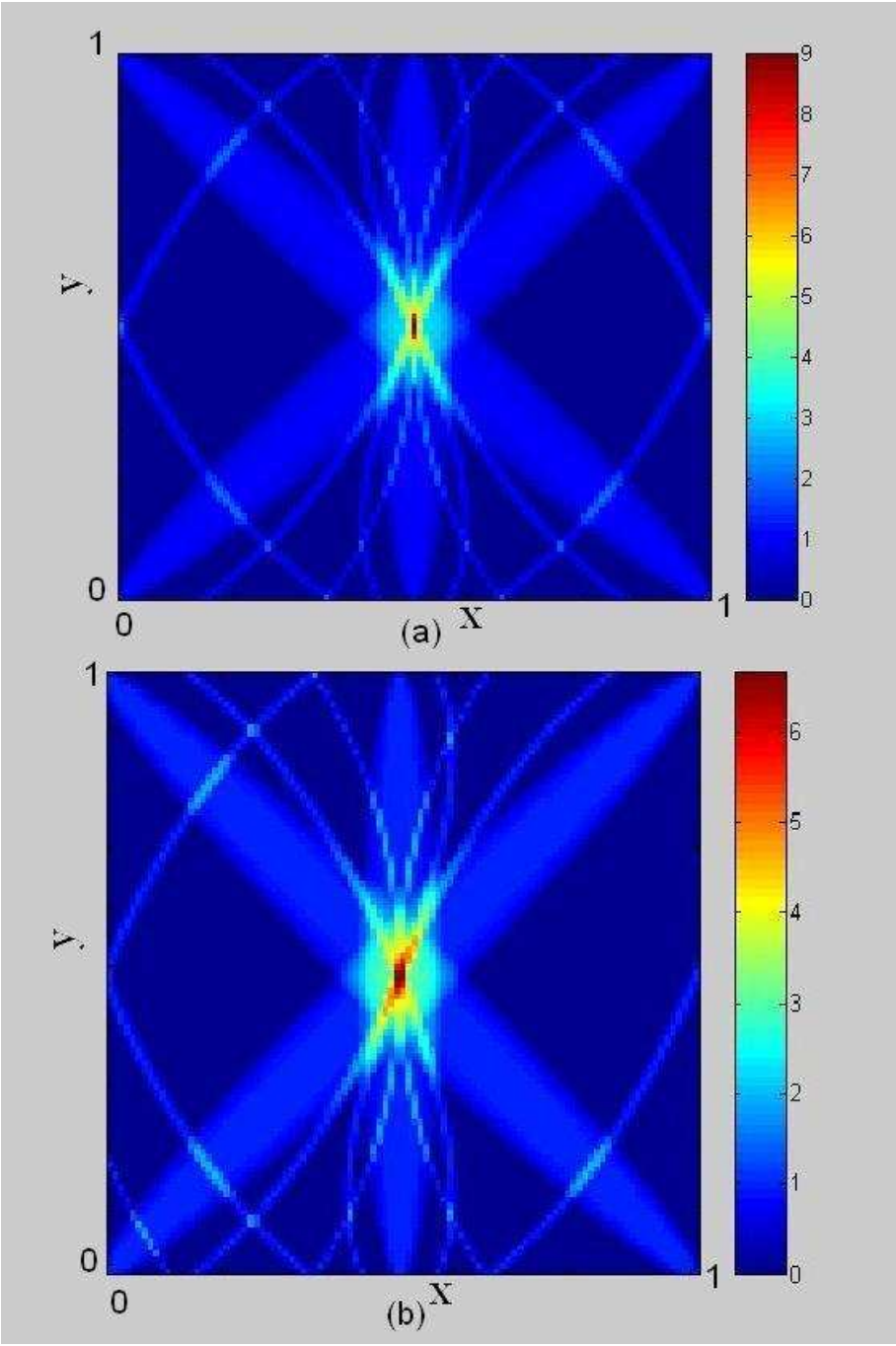


Figure M.3: (a) S(x,y) for true multi-path distances (b) S(x,y) for noisy multi-path distance measures

Appendix N

Synchronization and Distance Estimation in a Sensor Network

To complete the localization problem, we are going to study the synchronization problem in a sensor network. Our scheme is a centralized linear estimation scheme which achieves optimality if the noises are modelled as additive iid Gaussian noises.

The performance of the localization scheme that we proposed is highly dependent on the accuracy of the distance measurements. Two kinds of noises could occur in the estimation of distances. One is the measurement noises, which is often assumed to be iid Gaussian, the other is the *system error* from the offsets of the sensor clocks which can be treated as the *system parameters*. In a UWB system, if a sensor can both send and receive signals, then we can estimate the offsets of the sensor clocks *and* distances simultaneously.

Assume that sensor $i, i = 1, 2...N$ has an clock offset T_i, i.e. the clock of sensor i is T_i when the universal clock is 0. Notice that since we have *no* universal clock in the sensor network, the only thing matters is the *relative* offset.

Define

$$S = \{(i, j)|\text{sensor i and j can communicate to each other}, i < j\}$$

. We write $S = \{(i_1, j_1), ...(i_{|S|}, j_{|S|})\}$.We have the following 2 equations:

$$\begin{aligned} \frac{d_{ij}}{c} - T_i + T_j &= \varepsilon_{ij} - E_{ij} \\ \frac{d_{ji}}{c} - T_j + T_i &= \varepsilon_{ji} - E_{ji} \end{aligned} \tag{N.1}$$

Where E_{ij} is the estimated time difference at sensor j. $d_{ij} = d_{ji}$ is the distance between sensor i, j,ε_{ij} is the estimation error which can be assumed to be iid Gaussian r.v. $\sim N(0, \sigma^2)$, c is the speed of light.

So now we have a bunch of linear equations, unknown variables are $d_{ij}, (i,j) \in S, T_i, i = 1, 2...N$, but the linear equation system is singular. This can be easily proved by the fact that $x_{ij} = 0\ \forall i, j,\ \ T_i = c, \forall i$ is in the zero space of the linear equation system in Eqn.N.1. Now think of the N dimensional space R^N, the solutions of $T_i, i = 1, ..N$ is a point in R^N. Essentially all the lines perpendicular to the $N-1$ dimensional plane $\sum_{i=1}^{N} T_i = 0$ form an equivalent class, where every point on the line are equivalent. So we add one more constraint $\sum T_i = 0$ and do the estimation in the subspace $\sum T_i = 0$. Here the form of the constraint could be something like $T_1 = 0$ or $T_3 + T_7 = 5$. But our choice is the only symmetric linear constraint. With this additional constraint we can figure out *one single* solution in an equivalent *class*.

Now we have a linear equation system as following:

$$A\overrightarrow{x} = \begin{pmatrix} \overrightarrow{b} + \overrightarrow{\varepsilon} \\ 0 \end{pmatrix} \tag{N.2}$$

Where $\overrightarrow{x} = (T_1, ...T_N, d_{i_1 j_1}, ...d_{i_{|S|} j_{|S|}})^T$ is the unknown vector of dimension $N + |S| \times 1$. $\overrightarrow{b} = (E_{i_1 j_1}, E_{j_1 i_1}, ...E_{i_{|S|} j_{|S|}}, E_{j_{|S|} i_{|S|}})^T$ is the time difference estimation vector of dimension $2|S| \times 1$. $\overrightarrow{\varepsilon} = (\varepsilon_{i_1 j_1}, \varepsilon_{j_1 i_1}, ...\varepsilon_{i_{|S|} j_{|S|}}, \varepsilon_{j_{|S|} i_{|S|}})^T$ is the noise vector of dimension $2|S|$. A is a $(2|S|+1) \times (N + |S|)$ dimensional matrix. The $2k-1, 2k$ th, $k = 1, 2, ..., |S|$, row vector of A is a_{2k-1}, a_{2k} (each one is of dimension $1 \times (N + |S|)$) :

$$\begin{pmatrix} a_{2k-1} \\ a_{2k} \end{pmatrix} \overrightarrow{x} = \begin{pmatrix} \frac{d_{ij}}{c} - T_i + T_j \\ \frac{d_{ij}}{c} - T_j + T_i \end{pmatrix} \tag{N.3}$$

The $2|S| + 1$ th row vector of A is simply $(1, 1, ..., 1)$.

Notice that $\overrightarrow{\varepsilon}$ is a zero mean gaussian random vector with autocorrelation $\sigma^2 I_{2|s|}$

The optimal estimation of x is simply

$$\hat{\overrightarrow{x}} = (A^T A)^{-1} A^T \begin{pmatrix} \overrightarrow{b} \\ 0 \end{pmatrix} \tag{N.4}$$

A second thought on this problem leads to the following solution which is more computational efficient. Notice that d_{ij} which is equal to d_{ji} only appears in two equations. From Eqn.N.1, and given the noises are iid gaussian, the optimal estimation of d_{ji} is simply :

$$\hat{d_{ji}} = \frac{-c(E_{ij} + E_{ji})}{2} \tag{N.5}$$

It is independent of the estimation of T_i's. This is the optimal estimation of T_i, so it must be the same as the solution in Eqn.N.4. Substitute the estimation of d_i into Eqn. N.1, we have the following linear equations:

$$\begin{aligned} -T_i + T_j &= \frac{\varepsilon_{ij} - \varepsilon_{ji}}{2} + \frac{E_{ji} - E_{ij}}{2} \\ -T_j + T_i &= \frac{\varepsilon_{ji} - \varepsilon_{ij}}{2} + \frac{E_{ij} - E_{ji}}{2} \end{aligned} \tag{N.6}$$

As we can see, those two equations are identical. Thus one is redundant. We will use the second one of the two in the further analysis. Now we have a linear equation system as following:

$$G\overrightarrow{y} = \begin{pmatrix} \overrightarrow{h} + \overrightarrow{\varepsilon'} \\ 0 \end{pmatrix} \tag{N.7}$$

Where $\overrightarrow{y} = (T_1, ...T_N)$ is the unknown vector of dimension N. $\overrightarrow{h}$ is a $|S|$ dimensional vector of the observations, $\overrightarrow{h} = (\frac{E_{i_1 j_1} - E_{j_1 i_1}}{2}, ... \frac{E_{i_{|S|} j_{|S|}} - E_{j_{|S|} i_{|S|}}}{2})^T$. $\overrightarrow{\varepsilon'} = (\frac{-\varepsilon_{i_1 j_1} + \varepsilon_{j_1 i_1}}{2}, ... \frac{-\varepsilon_{i_{|S|} j_{|S|}} + \varepsilon_{j_{|S|} i_{|S|}}}{2})^T$ is the noise vector of dimension $|S|$, where $\overrightarrow{\varepsilon'} \sim N(0, \frac{\sigma^2}{2} I_{|S|})$. G is a $(1 + |S|) \times N$ dimensional matrix. The k th, $k = 1, 2, ..., |S|$, row vector of G is simply $(0, 0, ...0, 1, 0, ..., 0, -1, 0, ...0)$, i.e. the i_k th element is 1, the j_k th element is -1. And the optimal (also linear) estimation of $\overrightarrow{y}$ is:

$$\hat{\overrightarrow{y}} = (G^T G)^{-1} G^T \begin{pmatrix} \overrightarrow{h} \\ 0 \end{pmatrix} \tag{N.8}$$

From Eqn.N.7, we know that $\hat{\overrightarrow{y}} = \overrightarrow{y} - (G^T G)^{-1} G \overrightarrow{\varepsilon'}$. The estimation error is a linear transform of a zero mean white gaussian vector, which is still a zero mean gaussian vector. Now suppose in the sensor network, every sensor pair i, j can communicate to each other, i.e. $|S| = \frac{N(N-1)}{2}$. Then we have some interesting result :

$$G^T G = N I_N \tag{N.9}$$

Where I_N is the $N \times N$ identical matrix. So the estimation error of $\overrightarrow{y}$ is a N dimensional zero mean gaussian vector with autocorrelation:

$$\begin{aligned} E((\overrightarrow{y} - \hat{\overrightarrow{y}})((\overrightarrow{y} - \hat{\overrightarrow{y}}))^T) &= E((G^T G)^{-1} G^T \begin{pmatrix} \overrightarrow{\varepsilon'} \\ 0 \end{pmatrix} (\overrightarrow{\varepsilon'}^T, 0) G (G^T G)^{-1}) \\ &= N^{-1} I_N G^T \frac{\sigma^2}{2} \begin{pmatrix} I_{\frac{N(N-1)}{2}} & 0 \\ 0 & 0 \end{pmatrix} G N^{-1} I_N \\ &= \frac{\sigma^2}{2N^2} \begin{pmatrix} N & -1 & \dots & -1 \\ -1 & N & \dots & -1 \\ \vdots & & \ddots & \vdots \\ -1 & & \dots & N \end{pmatrix} \end{aligned} \tag{N.10}$$

From the above discussion we conclude this section. In a sensor network with N nodes, the variance of the clock-offset estimation is decreasing at rate N^{-1}, if the transmission power per sensor is constant.