Ranging and Positioning in Wireless Sensor Networks

A DISSERTATION
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Seshan Srirangarajan

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Professor Ahmed H. Tewfik, Adviser

August 2008

Acknowledgements

I would like to take this opportunity to express my gratitude to my adviser Professor Ahmed H. Tewfik, for his guidance, encouragement and advice throughout the course of my graduate studies. His insights, observations and suggestions have contributed immensely to the success of the research presented in this dissertation.

I am thankful to Professor Zhi-Quan (Tom) Luo for initiating the work presented in Chapter 4 on SOCP relaxation and for serving on the preliminary oral examining committee. I would also like to thank the members of my final defense examining committee, Professor Nihar Jindal, Professor Mostafa Kaveh and Professor Yongdae Kim, for the time and effort spent by them. Their comments and suggestions have helped improve this dissertation.

I would also like to gratefully acknowledge the support received from Honeywell International under the Honeywell University Affiliate Program during the first year of this research. Comments and suggestions provided by Mr. Ramakrishna Budampati and Mr. Patrick Gonia of Honeywell during this period have also contributed immensely to this research. I also thank Mr. Anoop Mathur, Dr. Soumitri Kolavennu and Mr. Steve Huseth of Honeywell for giving me the opportunity to intern with the Wireless Technologies group at the Honeywell Research Labs in Minneapolis. This provided me valuable experience of working in an industry research lab and also helped me in my own research.

I am also grateful for the company of and the numerous discussions I had with my officemates Dr. Alain Tchagang, Fikri Goksu, Vikrham Gowreesunker, Dr. Faisal Shah, Dr. Firat Ince, and other fellow graduate students. I am glad to have got the opportunity to work with my friend Dr. Shah. I have learnt a lot from him and have thoroughly enjoyed working with him. The research presented in Chapter 5 of this dissertation would not have been possible without his collaboration. I would also like to acknowledge the help and support I received from my friend Ms. Shen Dong.

I would also like to thank Professor Larry Kinney for his support and encouragement, and the opportunities he provided to help develop my teaching skills. I am also thankful to Professor David Lilja for the invaluable support he provided towards the end of my doctoral program. I also thank the Department of Electrical and Computer Engineering and the University of Minnesota for giving me the opportunity to pursue graduate studies.

And finally, but not the least, I would like to express my deepest gratitude to my parents. I am grateful for their love and support all through these years, and for the values they inculcated

in me. They have not only made this doctoral degree possible but have also enabled me become a better human being.

This dissertation is dedicated to

MY PARENTS

Ranging and Positioning in Wireless Sensor Networks

ABSTRACT

Ranging and positioning in wireless sensor networks refers to the ability to determine the positions of all nodes in a sensor network using the known positions of a few nodes called reference nodes and pairwise distance or range estimates between neighboring nodes. This is also known as the sensor network localization problem. In this thesis we first present two time-of-arrival based localization algorithms for indoor quasi-static environments based on statistical modeling of the ultra-wideband multipath channel. A model of the multipath channel in the form of the signal return and noise characterization is derived, and utilized to distinguish signal components from noise. The first localization algorithm uses multiple (ranging) signal receptions at each reference node, to differentiate between line-of-sight and non-line-of-sight components, and to accurately estimate the position of the line-of-sight component in the received multipath signal. The location is estimated through a mathematical programming problem formulation. The second localization algorithm employs a time-of-arrival based algorithm to obtain pseudo range estimates which are then used in a spatial domain quasi-maximum likelihood method for location estimation. Using a synthesized bandwidth of 2 GHz, a 4-bit analog-to-digital converter and with 5-10 dB signal-to-noise ratio, location estimation with close to sub-meter accuracy is achieved. Furthermore, the associated range estimation error does not increase with increase in the transmitter-receiver range.

We next present a distributed solution of the sensor network localization problem based on second-order cone programming relaxation. This algorithm is independent of the ranging technique being used and is computationally more efficient than most contemporary approaches, and scalable to networks with thousands of nodes. We show that the nodes can estimate their positions based on local information. Unlike previous approaches, we also consider the effect of inaccurate reference node positions. In the presence of reference node position errors, the localization is performed in three steps. First, the unlocalized nodes estimate their positions using information from their neighbors. In the second step, the reference nodes refine their positions using relative distance information exchanged with their neighbors and finally, the previously unlocalized nodes refine their position estimates. We demonstrate the convergence of the algorithm numerically. The simulation

results, shown for both uniform and irregular network topologies, illustrate the robustness of the algorithm to reference node position and distance estimation errors.

We also present the prototype implementation of a directional beacon based positioning algorithm using radio frequency signals. This algorithm allows each unlocalized node to compute its position with respect to a set of reference nodes which are equipped with rotating directional antenna. The directional beacon based algorithm eliminates the need for strict synchronization between the reference nodes and the unlocalized node. In contrast to time-of-arrival based positioning algorithms that rely on signal propagation time and bandwidth, the directional beacon based algorithm depends on the width of the antenna beampattern and the rotational speed of the directional antenna. We will show that these parameters can be optimized in a low cost solution while providing good position estimates. The system implementation is based on the GNU Radio software platform and the Universal Software Radio Peripheral as the hardware component. To deal with obstructed line-of-sight scenarios, we do not rely purely on the received signal strength and instead formulate a least squares problem to estimate the line-of-sight component in a multipath environment. These signal processing techniques yield a more accurate estimation of the bearing of the unlocalized node with respect to each of the reference nodes. We demonstrate the ability to obtain unlocalized node position estimates with sub-meter accuracy by transmitting a narrowband signal of 1 KHz bandwidth in the 2.4 GHz band.

Finally, event detection scenarios in sensor networks are considered. The goal in these network deployments is to detect certain critical or emergency conditions with minimum possible delay. We propose a heuristic based sensor selection and a sequential detection procedure that significantly improves the detection speed, measured in terms of the number of measurements needed for detection. In the proposed model, the fusion center selects one sensor at a time for measurement while maximizing a greedy heuristic. Instead of collecting a fixed number of measurements, the fusion center collects one measurement at each time step, until by some sequential decision rule the collection stops and a decision is made. The sequential detection procedure significantly outperforms a non-sequential (or fixed sample size) detector in that it always needs fewer measurements on average to achieve the same detection performance. In addition, we derive a simplified heuristic under the Gaussian probabilistic model. It is seen that the simplified heuristic performs as good as or slightly better than the greedy heuristic. The greedy heuristic based sensor selection provides a general framework for probabilistic models where a simplified heuristic is difficult to obtain.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Advances in low-power circuit design, simple yet reasonably efficient wireless communication equipment, reduced manufacturing costs and relative ease of deployment has made wireless communication technology ubiquitous. Ranging and location awareness are desirable features in a wireless network, with applications in asset tracking, locating people in emergency situations and robotics. Ranging refers to the ability to determine the distance between two neighboring nodes and location awareness refers to the knowledge of a nodes' position.

Recent years have also seen wireless sensor networks become extremely popular with the large-scale deployment of these networks with thousands of sensor nodes [3]. These networks typically combine wireless communication components, minimal computation capabilities and some sensing of the physical environment into a network that can be deeply embedded into the physical environment. All these components together in a single device form a sensor node. Some of the application areas for sensor networks are industrial automation (process control), military (real-time monitoring of troop movements), utilities (automated meter reading), building control and environmental monitoring [4].

1.1 Motivation

Sensor nodes measure some physical quantity(s) at a given position. In most applications, the data reported by the sensors is relevant only if tagged with the accurate location of the sensor nodes. Thus knowledge of the node positions becomes imperative. Using nodes equipped with Global

Positioning System (GPS) is a costly option. The location awareness feature is also emphasized in the IEEE 802.15.4a wireless personal area networks [5]. The principle interest of this standard is to provide communications and high precision ranging/location capability with 1 m accuracy or better. Position location in wireless personal area networks (WPANs) has found a number of applications ranging from commercial and residential (tracking people in assisted-living places and assets in a manufacturing facility) to public safety and military (tracking fire fighters and soldiers during their missions) [6].

The sensor network localization problem can be stated as follows. Assuming knowledge of the positions of some nodes and some pairwise distance measurements, determine the position of all nodes in the network. We will refer to nodes whose positions are known a priori as the reference nodes (RN) and nodes whose positions are unknown as the unlocalized nodes (UN). The sensor network localization problem typically consists of two main sub-problems: (i) the ranging sub-problem, to determine the distance (or range) between two neighboring nodes and (ii) the positioning sub-problem, to determine the position or location of the nodes given some pairwise distances.

This thesis examines several problems related to positioning in wireless networks with specific focus on wireless sensor networks. In Chapter 3 we present localization algorithms based on statistical modeling of the ultra-wideband physical layer channel. In Chapter 4 we present a distributed solution of the positioning sub-problem. This algorithm, based on second-order cone programming relaxation, is independent of the ranging technique used to obtain the pairwise distances. Chapter 5 presents the prototype implementation of a directional beacon-based positioning algorithm and signal processing solutions for the analysis of the experimental data. In Chapter 6 we focus on the problem of selecting sensors for measurement in event detection scenarios so as to achieve faster detection while meeting certain performance criterion.

1.2 Contributions and Thesis Organization

1.2.1 Localization Algorithms for UWB Multipath Environments

The use of ultra-wideband (UWB) communications at the physical layer of wireless systems allows for very precise ranging due to the high time resolution of UWB signals. In indoor environments the presence of reflectors and obstructions between the transmitter and receiver results in multiple

copies of the transmitted signal being received at the receiver. This phenomenon is known as multipath propagation. The high time resolution of UWB signals aids in mitigating the effect of multipath propagation. The signal received following the direct path from the transmitter to the receiver is referred to as the line-of-sight (LOS) component. The signals received following reflection, scattering etc are referred to as the non-line-of-sight (NLOS) components.

In Chapter 3 we present two time-of-arrival (TOA) based localization algorithms for indoor quasi-static environments based on statistical modeling of the ultra-wideband multipath channel. A statistical model of the multipath channel in the form of the signal return and noise characterization is derived. The first algorithm uses multiple (ranging) signal receptions to distinguish signal components (LOS/NLOS) from noise and accurately estimate the LOS component. The second algorithm fuses the range and position estimation phases in such a way that the geometric relationship between the TOAs aids in the range estimation as well. Using a bandwidth of 2 GHz, a 4-bit analog-to-digital converter and with 5-10 dB signal-to-noise ratio, location estimation with close to sub-meter accuracy is achieved. It will also be shown that the associated range estimation error does not increase with increase in the transmitter-receiver range [7, 8, 9].

1.2.2 Distributed Sensor Network Localization using SOCP Relaxation

In Chapter 4 we focus on the positioning sub-problem. The distance measurements between neighboring nodes can be obtained via time of arrival (TOA) [10, 11], time difference of arrival (TDOA) [12], angle of arrival (AOA) [13], received signal strength (RSS) [14] or other techniques [15, 16]. However due to the resource constraints on the sensor nodes the distance measurements are inaccurate or noisy. In addition, the reference node positions may be inaccurate even when determined with the use of GPS or other techniques. Most approaches in the literature do not account for inaccuracies in the reference node positions. The computational efficiency and scalability of the algorithms are important considerations in addition to the localization accuracy. A number of methods, based on minimizing some global error function, have been explored to account for the measurement uncertainties. It is observed that the computational complexity varies based on the optimization model chosen.

We present a distributed algorithm, based on the second-order cone programming (SOCP) relaxation, to determine the position of all nodes in the network given a few reference nodes and some pairwise distance measurements. This algorithm is independent of the ranging technique used

to obtain pairwise distances between neighboring nodes. An extensive simulation study, on uniform and irregular network topologies, will be presented demonstrating the robustness of the proposed algorithm to reference node position and distance estimation errors. The performance gains, in terms of problem size reduction and computational efficiency, are achieved without sacrificing localization accuracy [17, 18].

1.2.3 Implementation of a Directional Beacon based Positioning Algorithm

In Chapter 5 we present the implementation of a directional beacon based positioning algorithm using radio frequency (RF) signals. The existing localization systems can be broadly classified into two categories. The first category consists of systems which develop a signaling system and support infrastructure focused primarily on the positioning and tracking application. Systems in the second category tend to use existing wireless network infrastructure with minimal modifications to locate unlocalized nodes. The first category systems are designed for a desired positioning accuracy and tend to use expensive, dedicated devices for the purpose. Examples of such systems include GPS. The second category of systems, in the absence of dedicated devices, use smart algorithms to overcome the low accuracy of the measured quantities. We demonstrate a system that fits in this second category.

In the proposed implementation each UN computes its position with respect to a set of RNs equipped with rotating directional antenna. This algorithm does not need synchronization between RNs and the UN. We demonstrate the ability to obtain UN position estimates with sub-meter accuracy by transmitting a narrowband signal of 1 KHz bandwidth in the 2.4 GHz band [19, 20]. This implementation will be shown to be resilient to timing and synchronization errors.

1.2.4 Sensor Selection and Event Detection using Heuristic based Sequential Hypothesis Testing

The development of extremely small sensing devices has led to the use of large scale networks of these devices in numerous applications. Sensor nodes have enabled us to monitor/detect environments that have not been reachable until now. Sensor networks are typically designed to monitor physical environments for days, months or even years. To obtain a complete picture of

the environment it is desirable to collect information from all sensor nodes in the network. On the other hand, sensor networks have energy constraints that do not allow taking measurements from many sensors and transmitting these measurements reduces the network lifetime. Thus, in order to extend the network lifetime it is essential to carefully select the sensor nodes from which information will be collected. Several algorithms have been proposed in the literature for sensor selection.

In Chapter 6 sensor networks used for event detection scenarios are considered. The goal in these network deployments is to detect certain critical events or emergency conditions with minimum possible delay. We propose a heuristic based sensor selection and a sequential detection procedure that significantly improves the detection speed, measured in terms of the number of measurements needed for the detection.

Chapter 2

Related Work

2.1 Localization in UWB Multipath Environments

Ranging and location estimation in wireless networks has received considerable attention from the research community recently. Some survey articles on the research area and its applications are [15, 16, 21, 4]. Traditional localization techniques use one or more of the following measures: received signal strength (RSS) indicators [14], time of arrival (TOA) [10, 11], time difference of arrival (TDOA) [12], or angle of arrival (AOA) [13]. Techniques based purely on signal strength are prone to inaccuracies and large variances in position estimates [6, 22].

TOA/TDOA techniques using ultrasound signals have also been explored. These provide limited range and need additional hardware. Ultra-wideband (UWB) signals have been employed in position location systems [10, 11, 23, 24]. The use of ultra-wideband (UWB) communications allows for very precise ranging due to the high time resolution of UWB signals. However the presence of multipath components in the received signal due to non-line-of-sight (NLOS) propagation makes the problem of estimating the line-of-sight (LOS) position and determining the range especially challenging. The LOS time delay estimation and hence positioning accuracy is inversely proportional to the bandwidth of the signals and inversely proportional to the square root of the observation time [25]. Most of the work reported in the literature is based on assumptions that the line-of-sight (LOS) signal is always present [10, 11]. In [10] it is assumed that the LOS component is the earliest arrival and a threshold is used in a window of the received signal to estimate the LOS component. The results showed that the estimation error increases rapidly with the transmitter

(TX) - receiver (RX) range. These approaches, despite using bandwidths in excess of 1 GHz, do not achieve the sub-meter (< 1 m) ranging accuracy that some applications require.

Many approaches have been proposed in the literature that attempt to find some distinct properties of NLOS range measurements to distinguish them from LOS measurements [26, 27, 28]. In [26] it is observed that the NLOS range measurements have greater variance than the LOS measurements. Reference [27] models NLOS and LOS range measurements as Gaussian random variables, and derives the theoretical framework for nonparametric and parametric hypothesis tests. In [28], the authors present an algorithm to detect LOS measurements when NLOS error is intermittently present in a time series of range measurements. These attempts and our own work on NLOS identification suggest that using pure statistical characteristics to distinguish NLOS measurements from LOS measurements is a difficult problem. Location estimation algorithms for NLOS propagation have been studied in the literature. In [29] the property that NLOS errors are always positive is used to estimate the location by adding some constraints or penalty function.

2.2 Sensor Network Localization using Optimization Techniques

Sensor network localization has been an area of active research in recent years with a large number of current and future applications. Most localization systems estimate the node positions using some kind of range or distance information between nodes. However, some systems such as [21] perform localization using connectivity information. Such systems depend on a high density of reference nodes (RNs) and result in relatively low positioning accuracy.

It will be shown in Chapter 4 that the localization problem in its original form is a non-convex optimization problem with many local minima. Doherty et al. in [30] formulate the localization problem as a feasibility problem with convex radial constraints. However, this method requires centralized computation which is not suitable for large-scale networks. Shang proposed a distributed localization method MDS-MAP(P, R) based on multi-dimensional scaling (MDS) [2]. This method builds for each node a local map of the small sub-network in the node's vicinity and then merges these local maps to form a global map followed by a refinement step. This method needs only a few RNs (or in the absence of RNs generates a relative map) and partly overcomes the drawback associated with centralized computation. However, the construction of local maps for

each node results in enormous amount of redundant computation as it is seen that most local maps are not used in building the global map. MDS-MAP(P, R) is not entirely suitable for large networks as the cost of refining and merging the local maps grows faster than linear due to the larger maps being manipulated. Also the cost of refining the global map, a single global optimization step, becomes dominant for large networks.

Costa et al. [1] apply distributed weighted MDS (dwMDS) to the sensor network localization problem and formulate the problem using a general form of the cost function we use in Chapter 4. They solve the minimization problem using majorizing functions.

Biswas and Ye solve the problem using the semidefinite programming (SDP) relaxation [31]. The SDP relaxation approach can solve small problems effectively. The authors report a few seconds of PC execution time for a 50 node network. They have also proposed two techniques to improve the accuracy of the SDP solution [32]. The first technique adds a regularization term to the objective function to force the SDP solution to lie close to a low dimensional subspace of R^d and the second technique improves the SDP estimated solution using a gradient-descent method. However, the number of constraints in the SDP model is $O\left(n^2\right)$, where n is the number of nodes in the network

Most SDP solvers can handle problems with at most 100 variables, while sensor networks typically have 100's of nodes resulting in problem dimensions in the 10,000's. To overcome this difficulty, Biswas and Ye proposed a distributed method for solving the SDP [33]. In this iterative distributed scheme, the RNs are first partitioned into many clusters according to their physical locations. A sensor is assigned to a cluster if the sensor has a direct link to one of the RNs. Then semidefinite programs are solved independently for each cluster. The nodes whose position becomes known are used to iteratively locate the remaining unlocalized nodes (UNs). The authors report a few minutes of PC execution time for a network with 4000 nodes. But, since the clustering is done based on geographic locations [34], each cluster may have only partial connection information for the border nodes if these have connections with multiple clusters. Thus nodes on the border of each cluster may not get positioned accurately [35].

We consider the second-order cone programming (SOCP) relaxation due to its simpler structure and the potential to be solved faster. The SOCP relaxation for the localization problem was first studied by Tseng [36]. It has been shown in [36] that even though the SOCP relaxation is weaker than the SDP relaxation, it can accurately position a large percentage of the sensors up to the

square root of the distance error. The localization approach presented in Chapter 4 enables the SOCP relaxation problem to be solved in a completely *distributed* fashion. Each UN executes the localization algorithm independently using distance information to the RNs and nodes with which it is directly linked.

A number of existing approaches consider the distance (or range) estimation errors [37, 38], however most do not consider the inaccuracy in RN positions which is also a significant source of error. The RNs are typically positioned using GPS or by means of surveying by humans. Civilian GPS accuracy is limited to about 15m while surveying is prone to human observation errors. In Section 4.4, we demonstrate the ability of the distributed SOCP approach to provide good localization accuracy even in the presence of significant errors in the RN positions.

2.3 Range-Free Sensor Network Localization Approaches

There have been investigations into range-free techniques for sensor network localization. These are attractive as a cost-effective alternative to range-based schemes but rely on high density of nodes and tend to offer lower localization accuracy. UNs use primarily connectivity information for inferring proximity to a set of RNs. In the Centroid localization scheme [21], UN localizes itself to the centroid of its proximate RNs. In APIT [39], each UN decides its position based on the possibility of being inside or outside of a triangle formed by any three RNs within its communication range. Spotlight system [40] creates well controlled (in time and space) events in the network while the sensor nodes detect and timestamp these events. From the spatio-temporal knowledge for the created events and the temporal information provided by sensor nodes, nodes' spatial information can be obtained. In the StarDust system [41], sensor nodes are equipped with optical retro-reflectors. An aerial device projects light towards the deployed sensor network, and records an image of the reflected light. An image processing algorithm is developed for obtaining the locations of sensor nodes. The Lighthouse system [42] uses a parallel light beam, that is emitted by a RN which rotates with a certain period. The UN detects the light beam for a period of time, which is dependent on the distance between it and the light emitting device.

2.4 Directionality based Position Location Algorithms

The TOA/TDOA based techniques need large bandwidths and long observations times which might be difficult to meet in some real world applications. TOA/TDOA techniques using radio frequency (RF) signals are sensitive to timing estimation errors. Small timing errors of the order of micro-seconds result in very large position errors because of the high propagation speed (3×10^8 m/s) of the electromagnetic waves. AOA information has been used earlier in the literature for determining the position of a node [43, 44, 45]. For AOA techniques, the error in time delay estimation translates into error in angle estimation which results in significantly smaller position error. The proposed implementation of the directional beacon based algorithm results in a position estimation error of only 0.5 m with a 500 msec error in delay estimation (refer Section 5.5 for further discussion on this matter). Thus, directionality based techniques can provide good accuracy with relatively inexpensive hardware.

McGillem and Rappaport [44] were one of the earliest to propose the use of AOA information for positioning and navigation along with a system implementation demonstrating the technique. They used infrared beacons with a rotational optical receiving system to obtain angular measurements using beacons. Nasipuri presented a directionality based positioning scheme in [45]. Later in [46] they proposed a system implementation using rotating optical beacon generators and sensor nodes equipped with photo sensors. Shah and Tewfik in [43] presented an enhanced positioning scheme based on directional beacons using the time of earliest arrival for detecting the LOS component. Although many directionality based positioning techniques have been described in the literature, very few have presented practical system implementations, and to our knowledge none based on radio signals has been presented. A positioning system based on radio signals allows the use of the existing transceiver circuitry on wireless devices for positioning. However, the hostile wireless propagation environment introduces distortion in the radio signals and poses unique challenges to the system design. Smart algorithms to address these issues are presented in Chapter 5.

2.5 Sensor Selection and Event Detection Schemes

A survey of various sensor selection schemes is presented in [47]. Information theoretic approaches to sensor management are presented in [48]. An information-driven sensor query approach is proposed in [49]. Sensor selection method based on mutual information is presented in [50]. An

entropy-based heuristic approach is proposed in [51] which greedily selects the next sensor to reduce overall uncertainty. Bounds on the performance of information theoretic measures, based on the notion of submodularity, are obtained in [52, 53]. It has been noted that most of the information theoretic measures for sensor selection are computationally expensive. A statistical approach to decide which sensors to use in answering a query was proposed in [54].

Chapter 3

Localization Algorithms for UWB Multipath Environments

In this chapter we present two novel localizations algorithms. The first algorithm solves the non-line-of-sight (NLOS) identification problem by using multiple signal receptions in a maximum likelihood estimation framework to differentiate between LOS, NLOS components and noise. The ranging algorithm is based on statistical modeling of the multipath channel. The multipath channel is modeled based on the IEEE 802.15.3a channel model 3 (CM3). The signal (LOS or NLOS) and noise components are characterized during an initial calibration phase, which is used to distinguish signal components from noise. Multiple signal receptions at each reference node (RN) are used to differentiate between LOS and NLOS components, and accurately estimate the LOS position in the received signal.

The mathematical programming approach presented here assumes that all the range estimates are LOS and uses two bounds (lower and upper) on each range estimate. The bounds are determined by the Cramer-Rao bound on the estimation error. Infeasibility of the constraints helps to identify NLOS estimates. This method will be shown to give more accurate position estimates than the previously reported approaches. Simulations also show that the range estimation error with this approach does not increase with the range.

It is seen that traditional localization methods can be decomposed into two steps. In the first step, the TOA (or TDOA) measurements associated with the visible RNs are obtained. In the

second step, a location algorithm is implemented, fusing the measurements obtained in the first step into a position estimate. It is seen that the geometric relationship between the TOAs (or TDOAs) is logically exploited in the second step to obtain the position estimate but is not used to assist in the TOA estimation of the first step [55].

The second algorithm we present in this chapter fuses the range (or TOA) estimation and location estimation phases in such a way that the geometric relationship of the TOAs also aids in the range estimation. This lowers the probability of reporting NLOS range estimates. This additional information is available while estimating the range and improves the localization accuracy dramatically compared to the traditional approaches. At the same time the proposed algorithm is computationally efficient.

The ranging algorithm, using the statistical characterization of signal (LOS/NLOS) and noise components, obtains multiple *pseudo* range estimates. One of these estimates will be due to the LOS component while others will be due to NLOS components or noise. The range estimates from all visible RNs are combined in a spatial-domain quasi-maximum likelihood (QML) estimation technique to arrive at the final location estimate.

The approaches presented here achieve better results by exploiting the knowledge of the channel statistics. These statistics are determined during an initial calibration phase and thereafter used during the localization phase. It will also be shown that the calibration phase is sufficient to gather all the necessary channel information and no other prior information about the channel is needed. Suitable modifications enable these algorithms to be used in low bandwidth (BW) and multi-band systems.

3.1 Localization Model

We refer to the nodes whose positions are known a priori as the reference nodes (RN) and nodes whose positions are to be estimated as the unlocalized nodes (UN). Consider a setup in which a number of UNs and RNs communicate with each other for estimating locations of the UNs. The RNs and UNs can be time-synchronized using techniques such as [56]. In the absence of such time synchronization, the range to a reference node is calculated using the Two-Way Time Transfer (TWTT) method, originally developed to permit precise synchronization and range estimation between satellites and ground stations. In our implementation of the TWTT, reference node A

sends a message to unlocalized node B along with the time at which it was sent. After a turnaround time T, B sends a message back to A along with the time at which it was sent and the arrival time of the message it received from A. Both A and B determine the arrival times of the message they receive by determining the time of earliest arrival corresponding to the message. Using the message transmission times and the earliest arrival times of the messages at both A and B, A can determine the time of flight between A and B. This information is sent to a central location where it is fused with similar information from other RNs to produce a location estimate of B.

The UNs and/or RNs transmit a short duration Gaussian monopulse (the ranging pulse) for the purpose of ranging and location estimation. The transmit signal, after going through the multipath channel, is input to a matched filter receiver. The matched filter output is subjected to thresholding to detect local peaks. A threshold is chosen based on the desired error performance and the estimated signal-to-noise ratio (SNR). This system model is depicted in Fig 3.1. Fig 3.2 shows the matched filter output in the presence of Gaussian noise. The waveform has significant multipath components and the signal peak does not occur at the leading edge of the waveform. A calibration phase, explained in the next section, aids in differentiating noise from LOS/NLOS peaks and multiple measurements are used to distinguish LOS from NLOS components.

In multi-band communication systems, the whole bandwidth is divided into several sub-bands. In each time interval, a signal is transmitted in one of the sub-bands. We assume that the UNs and RNs transmit a short duration Gaussian monopulse with a bandwidth of 528 MHz. Signals from 4 sub-bands are combined to give a virtual large bandwidth (2 GHz) signal using the technique in [57]. Signal in each sub-band is sampled at 1 GHz and then upsampled by a factor of 4 to give an effective sampling rate of 4 GHz. Higher sampling rates are achieved via processing in the digital domain. The sample rate for the waveform in Fig 3.2 is 20 GHz.

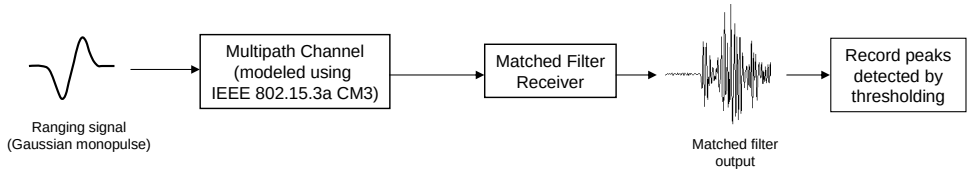


Figure 3.1: Block diagram of the proposed UWB ranging and localization system.

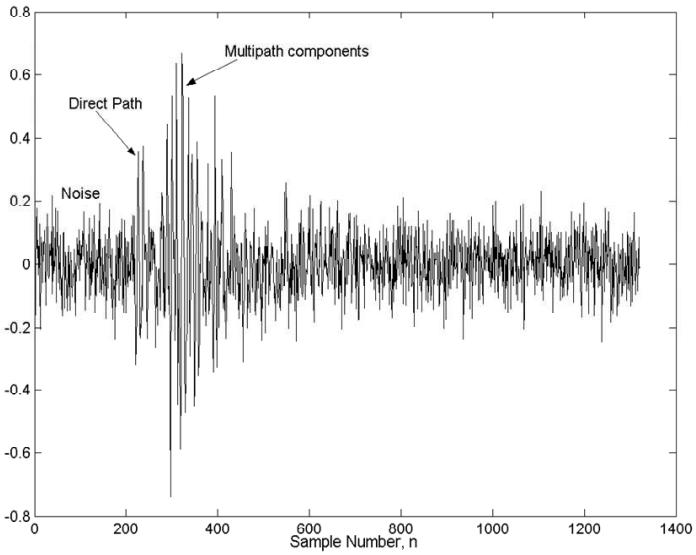


Figure 3.2: Matched-filter output in the presence of Gaussian noise.

3.2 Statistical Modeling: Offline Calibration Phase

In the calibration phase, two sets of training runs are carried out. The first set of training is done in an ideal noise free or virtual high SNR environment achieved by averaging several measurements over a short time interval. Alternatively, we can use a reliable channel model, such as the IEEE 802.15.3a. Here we report results using the IEEE 802.15.3a channel model 3 (CM3) which is based on NLOS (4-10 m) channel measurements reported in [58, 59]. Similar UWB channel models have been proposed in [60, 61, 62]. In a high SNR environment the local peaks detected will either be due to the LOS or NLOS components. It is reasonable to assume that under high SNR conditions and over a set of simulations, the first detected peak would be due to the direct path or the LOS and all other peaks are due to NLOS components. The measured peak strengths are normalized by the strength of the global peak in the output of the matched filter. Global peak is defined as the largest peak detected in a single received ranging pulse. Histograms are estimated for the strengths of the LOS and NLOS peaks using these measurements.

The second set of training runs are carried out in the presence of Gaussian noise but in the absence of a transmit signal, i.e., under noise-only conditions. The procedure outlined above is followed to estimate a histogram for the relative strength of the noise peaks. The histograms obtained from simulations performed using 250 different channel realizations are shown in Fig 3.3. It is seen

that the relative strength of the signal (LOS/NLOS) peaks follows an exponential distribution, whereas that of the noise peaks follows a lognormal distribution. By normalizing these histograms to unit area we obtain probability density functions for the relative strengths of the signal and noise peaks, which we denote by $f_{signal}(\rho)$ and $f_{noise}(\rho)$, respectively.

$$f_{signal}(\rho) = \frac{1}{\beta} \exp\left(-\frac{\rho}{\beta}\right)$$
$$f_{noise}(\rho) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{\ln^2\left(\frac{\rho}{\rho_0}\right)}{2\sigma^2}\right) \tag{3.1}$$

The histograms for the time difference between the location of the local peaks and the global peak in the matched filter output are also estimated, shown in Fig. 3.4. These results agree with the IEEE 802.15.3a channel model and illustrate an alternative approach for estimating these characteristics in the absence of a channel model. This characterization of the LOS/NLOS and noise peaks is used in the two ranging and location estimation algorithms which are described in the next two sections.

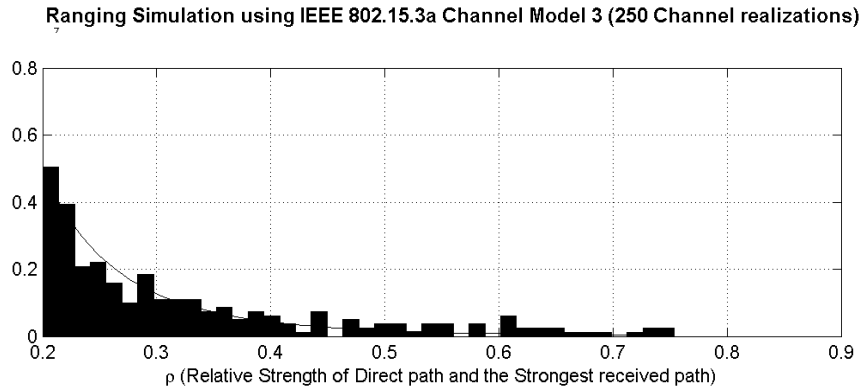
3.3 Ranging and Location Estimation Algorithm using Multiple Signal Receptions

3.3.1 Overview of the Ranging Approach

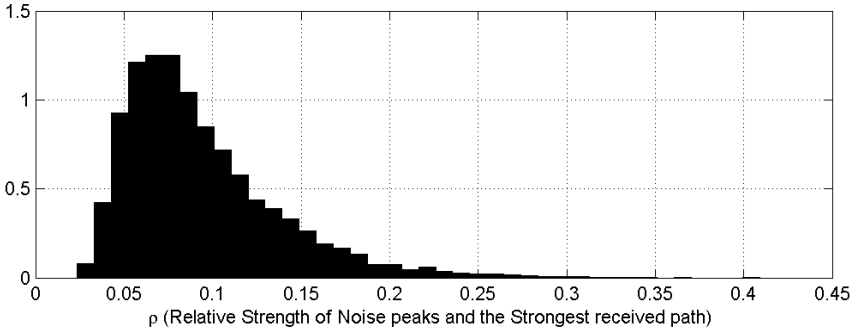
We assume all the nodes in the network are quasi-static or slow moving. Each UN transmits multiple copies of the ranging pulse at fixed intervals of time. All RNs within the radio range of a given UN record these multiple signal receptions to be used for range estimation. We detect the received signal peaks due to multiple pulses by thresholding and estimate the LOS component as the earliest peak across all receptions (the underlying assumption). Using multiple signal receptions helps to average over noise and increases the effective signal power and SNR, resulting in improved ranging accuracy.

3.3.2 Identifying Candidate LOS Component

To estimate the position of the LOS component, the time axis is divided into a number of small time bins. Each detected peak in a time bin is also referred to as a return. For each time bin two

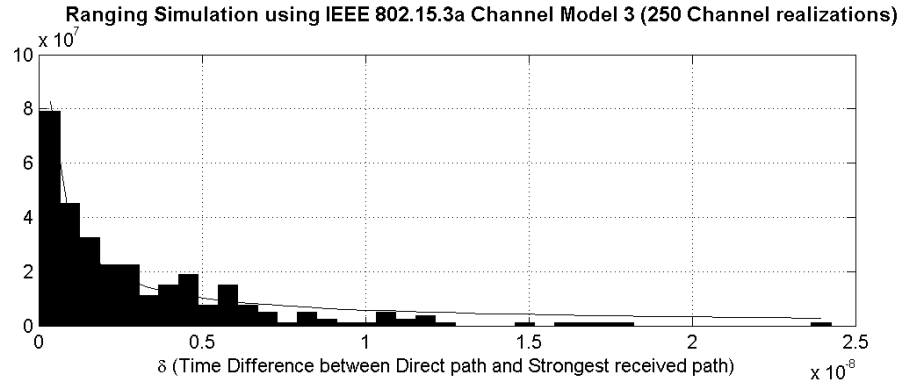


(a)

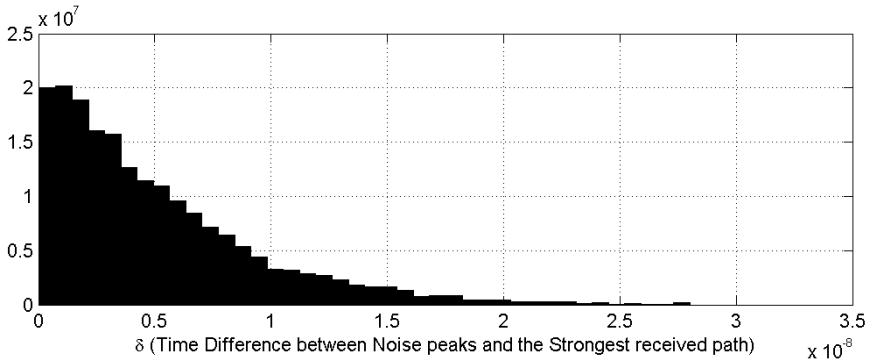


(b)

Figure 3.3: Relative strength (ρ) distribution for the: (a) signal (LOS/NLOS) peaks and (b) noise peaks.



(a)



(b)

Figure 3.4: Time difference (δ) distribution for the: (a) signal (LOS/NLOS) peaks and (b) noise peaks.

hypotheses are proposed: (i) the bin contains returns due to noise or (ii) the returns are due to a LOS or NLOS component. The likelihood of the peaks coming from the LOS or NLOS distribution is calculated looking across all of the collected signal receptions and using the histogram for the relative strengths of the LOS/NLOS components. Similarly, for each bin the likelihood of the returns being due to noise is calculated using the histogram for the relative strength of the noise peaks. Based on which likelihood function turns out to be larger, each bin is estimated to contain either a LOS or NLOS component or noise. The earliest time bin that has a higher likelihood of containing a signal (LOS or NLOS) component is chosen as the estimate of the direct path or LOS signal position.

The likelihood function under each hypothesis is evaluated as a product of the probabilities of the returns in a bin, across all the collected signal receptions, being either due to LOS/NLOS components or due to noise. The likelihood function evaluation procedure is shown in Figs. 3.5 and 3.6. Consider the case where we use three signal receptions for ranging at each node. We record all the (normalized) peaks detected in the three receptions in a given time bin. For each detected peak, the relative strength is calculated, say ρ_1. The probability of the peak being due to noise, $P(\rho_1)$, is found by locating ρ_1 on the relative strength distribution (x-axis) for the noise peaks and reading the corresponding value on the y-axis. This computation is shown in Fig 3.6. The product of these probabilities for all peaks in a given bin, gives the likelihood of the bin containing a noise peak. If a time bin does not contain any return, the probability of the return being less than the threshold is used to evaluate the likelihood function.

$$P_{noise}(t_i) = \prod_{j=1}^{S} P(\text{peak detected in time bin } i \text{ in signal } j \text{ is due to noise})$$

where S is the number of signal receptions collected at each RN.

Similar computation is done to obtain the likelihood associated with the time bin containing a signal return (LOS or NLOS peak) denoted by $P_{signal}(t_i)$. The earliest time bin t_i with $P_{signal}(t_i) \geq P_{noise}(t_i)$ is chosen as the LOS position estimate. Note that the peak chosen as the LOS estimate may actually correspond to a NLOS component. The location algorithm described next takes this possibility into account while estimating the location.

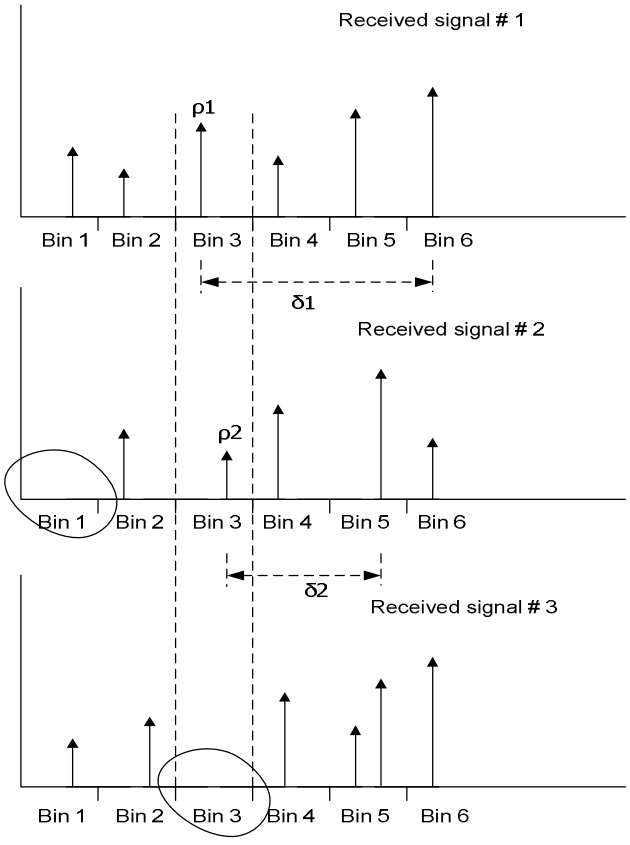


Figure 3.5: Illustration of Likelihood evaluation.

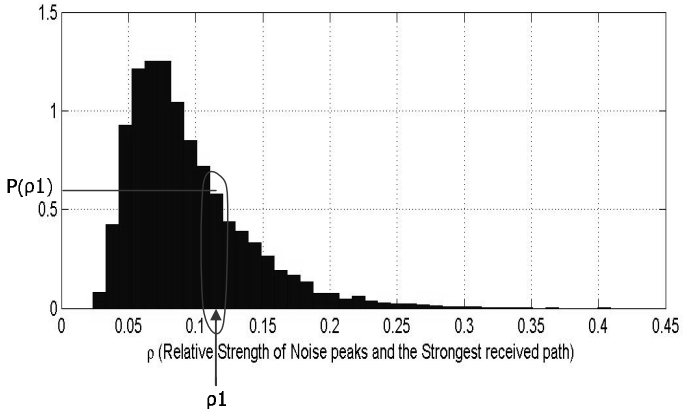


Figure 3.6: Illustration of Likelihood function evaluation.

3.3.3 Location Estimation Algorithm

In our localization set-up, range estimates to a given UN from a number of other nodes are used for location estimation. Each range estimate provides a circle centered on the corresponding node, on which the UN lies. In the absence of measurement error, the position of the UN is given by the intersection of the circles derived from the range estimates. However due to measurement error, the estimates correspond to circular rings whose width depends on the estimation error.

Let (x,y) be the unlocalized nodes' position, (x_i,y_i) the position of the i^{th} RN and r_i the range estimate from the i^{th} RN to the UN. The range estimation error (ϵ_r) is chosen to be a multiple of the Cramer-Rao lower bound (CRLB) on the range estimation error variance [57].

For each range estimate r_i:

$$(r_i - \epsilon_r)^2 \leq (x_i - x)^2 + (y_i - y)^2 \leq (r_i + \epsilon_r)^2 \tag{3.2}$$

where $i = 1, \ldots, N$. Let $K_i = x_i^2 + y_i^2$ and $R = x^2 + y^2$. Then (3.2) can be rewritten as:

$$(r_i - \epsilon_r)^2 - K_i \leq -2x_i x - 2y_i y + R \leq (r_i + \epsilon_r)^2 - K_i.$$

In matrix form,

$$h_1 \leq GZ \leq h_2$$

where

$$h_1 := \begin{bmatrix} (r_1-\epsilon_r)^2 - K_1 \\ (r_2-\epsilon_r)^2 - K_2 \\ \vdots \\ (r_N-\epsilon_r)^2 - K_N \end{bmatrix}, h_2 := \begin{bmatrix} (r_1+\epsilon_r)^2 - K_1 \\ (r_2+\epsilon_r)^2 - K_2 \\ \vdots \\ (r_N+\epsilon_r)^2 - K_N \end{bmatrix},$$

$$G := \begin{bmatrix} -2x_1 & -2y_1 & 1 \\ -2x_2 & -2y_2 & 1 \\ \vdots & \vdots & \vdots \\ -2x_N & -2y_N & 1 \end{bmatrix} \text{ and } Z := \begin{bmatrix} x \\ y \\ R \end{bmatrix}.$$

Thus the location estimation can be formulated as a constrained minimization that tries to find the maximum-likelihood (ML) location estimate assuming all the range estimates to be LOS [63].

$$\begin{aligned} \min \quad & \{(h_2 - GZ)^T \Psi^{-1} (h_2 - GZ)\} \\ \text{subject to} \quad & h_1 \leq GZ \leq h_2 \end{aligned} \tag{3.3}$$

where

$$\Psi = 4c^2 BQB,$$

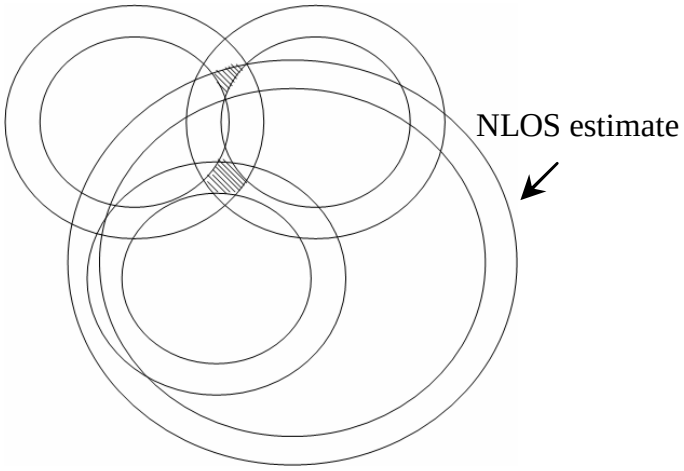
$$B = \text{diag}\{r_1, \ldots, r_N\},$$

Q is the noise covariance matrix and c is the signal propagation speed.

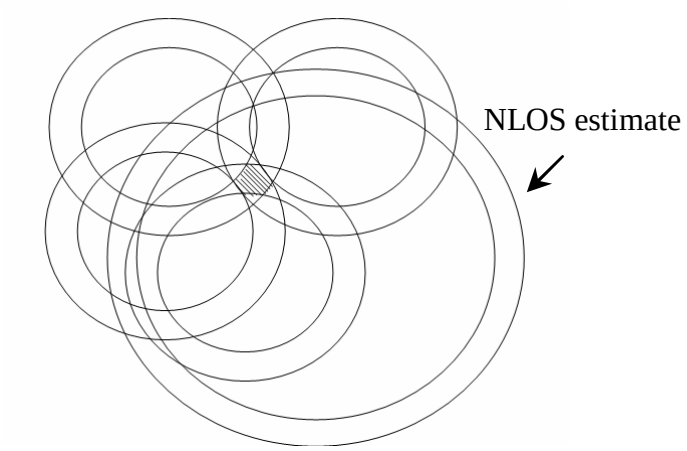
Infeasibility of the constraints will imply that one or more of the range estimates are NLOS and need to be dropped from the location estimation procedure. It is seen that if at most M estimates can be NLOS, then at least $(M+1)$ or more accurately $\max(M+1, 4)$ LOS estimates are needed to obtain an unambiguous location estimate. Figs. 3.7(a) and 3.7(b) illustrate the case with $M = 1$; four estimates give more than one possible location estimate, hence 5 range estimates are needed to resolve this ambiguity.

3.4 Spatial Domain Quasi-Maximum Likelihood Location Estimation Algorithm

Each node that needs to be localized transmits a ranging pulse. All RNs (and previously localized nodes) which receive the transmission, record the received pulse. Each of these RNs (or nodes)



(a) 1 NLOS and 3 LOS range estimates; resulting in location ambiguity.



(b) 1 NLOS and 4 LOS range estimates; location ambiguity resolved.

Figure 3.7: Location Estimation with 1 NLOS range estimate ($M = 1$).

executes the first phase of the localization algorithm to identify the *pseudo* range estimates. These range estimates are referred to as *pseudo* since only one estimate is due to the LOS component while the others are due to NLOS components or noise. The pseudo range estimates are then communicated to a central processing node along with the position of the node that measured them. At the central node this information is fused with similar information from other RNs (or nodes) to produce an estimate of the UN location. This localization model employs a centralized framework where most of the computation is done at a central node while minimal processing is done at the UNs. This is desirable when UNs are resource constrained and can only perform a minimal amount of computation. In a distributed framework, the RNs transmit the ranging pulse and the UN computes its location by estimating the pseudo ranges corresponding to each RN.

3.4.1 Identifying Pseudo Range Estimates

The ranging pulse transmitted by a UN is received by all RNs within its radio range. At each RN, a threshold is chosen to detect peaks in the received signal. The detected peaks are recorded in terms of their normalized signal strength ($\rho_{j,i}$) and the time difference ($\delta_{j,i}$) between the detected peak and the global peak in the received signal, where i refers to the i^{th} RN and j refers to the j^{th} detected peak. The time difference is used to estimate the pseudo range ($r_{j,i}$) due to each of the detected peaks (as if each of the peaks were due to the LOS component). Let $t_{peak,i}$ denote the time stamp of the global peak in the received signal at the i^{th} RN, then:

$$r_{j,i} = c \cdot (t_{peak,i} - \delta_{j,i}). \tag{3.4}$$

Each pseudo range estimate gives a circle centered on the corresponding RN with radius $r_{j,i}$ on which the UN could lie (refer Fig. 3.8).

3.4.2 Location Estimation Algorithm using Pseudo Range Estimates

The localization procedure using pseudo range estimates is described below:

Step 1. At each RN, detect peaks in the received signal by setting a threshold; record the pair ($\rho_{j,i}$,$\delta_{j,i}$) for each detected peak.

Step 2. Using (3.4) estimate $r_{j,i}$ for each detected peak.

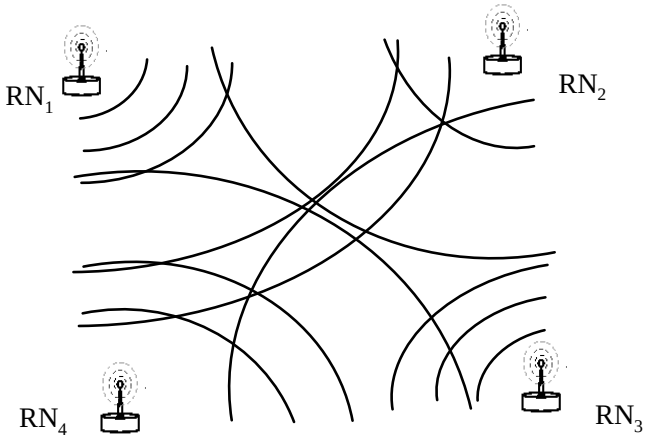


Figure 3.8: Pseudo range estimates from 4 RNs to an unlocalized node; the UN is located at the intersection of the LOS ranges.

Step 3. Each RN transmits the recorded information along with its location information to a central processing location.

Step 4. After information from all RNs has been received at the central location, the *area of interest*, where the UN could possibly exist, is divided into a grid of cells. (Note: *Area of interest* is determined based on the known locations of the RNs and the largest pseudo range estimate from each RN. This is reasonable since each pseudo range gives how far the UN could be located from the RN). The cell size is determined by the smallest resolvable time interval during the pseudo range identification phase.

Step 5. Represent the grid of cells as a matrix. The matrix is populated such that the entry in each cell represents the likelihood of the cell containing a LOS range estimate. This computation is explained next.

Likelihood Matrix Computation

Let $y_i(t) = x(t) + n_i(t)$ be the received signal at the i^{th} RN, where $x(t)$ represents the transmitted signal and $n_i(t)$ the Gaussian noise. We assume that noise is independent for each RN and for each received pulse. Let $\rho_{j,i}$ be the relative strength of the j^{th} peak, mapped to a cell, due to the signal received at the i^{th} RN. The mapping to a cell is done based on the pseudo range $r_{j,i}$

corresponding to the peak $\rho_{j,i}$. Then, the likelihood of the peak being due to signal or noise is given by $f(\text{signal}/\rho_{j,i})$ or $f(\text{noise}/\rho_{j,i})$, respectively.

$$f(\text{signal}/\rho_{j,i}) = \frac{f(\rho_{j,i}/\text{signal}).f(\text{signal})}{f(\rho_{j,i})} = \frac{f_{signal}(\rho_{j,i}).f(\text{signal})}{f(\rho_{j,i})}$$
$$f(\text{noise}/\rho_{j,i}) = \frac{f(\rho_{j,i}/\text{noise}).f(\text{noise})}{f(\rho_{j,i})} = \frac{f_{noise}(\rho_{j,i}).f(\text{noise})}{f(\rho_{j,i})}. \tag{3.5}$$

To compute the UN location, we need to combine measurements from all RNs. There are two possibilities to consider. These correspond to two models for the path loss from the UN to each of the RNs, and the likelihood of any peak being due to signal or noise: (i) We may consider the path losses to be correlated since they are dominated by a deterministic loss that is a function of the RNs and UN geometry. In this case, extensive calibration would be needed to model each of the individual path losses which is not practical. (ii) We may alternatively consider the path losses to be random and independent due to the arbitrary placement and dynamic structure of the obstructions between the UN and each of the RNs. In this case, we can assume that the path losses are independent realizations drawn from the same distribution. We use the latter model as a reasonable approximation in the presence of multipath propagation.

In each cell peaks from a number of RNs each with strengths $\rho_{j,i}$ ($i = 1, \ldots, N$ and $j = 1, \ldots, N_i$, where N_i are the number of peaks detected at the i^{th} RN) are reported. If each of these is an independent observation, the overall likelihood function would be a product of the individual likelihoods. Otherwise, the product of the individual likelihoods is not the overall likelihood. Nevertheless, it provides a reasonable cost function that we will maximize below. We refer to this cost function as a quasi-likelihood function.

Intuitively, each range estimate $r_{j,i}$ can be represented by a circle around the RN on a map of the network. The intersection of the circles from all the visible RNs would give a grid of possible locations for the UN. The position with the maximum likelihood (or quasi-likelihood) value is chosen as the UN location estimate.

Let $A_{measured,i}$ represent the matrix of likelihood values for the i^{th} RN. Since each pseudo range estimate comes from a peak in the received signal, the likelihood of it being due to the LOS signal component is related, directly to the likelihood of the peak coming from the signal (LOS/NLOS) distribution, and inversely to the likelihood of the peak being from the noise distribution. For each detected peak with relative strength, say ρ_i (the second subscript on ρ is omitted for the sake of simplicity), the likelihood of it being due to noise $f_{noise}(\rho_i)$, is obtained from the noise relative

strength distribution. Locate ρ_i on the x-axis of the noise distribution and the corresponding value on the y-axis gives $f_{noise}(\rho_i)$. This computation is similar to that shown in Fig. 3.6. The likelihood $f_{signal}(\rho_i)$ is calculated in a similar manner using the signal relative strength distribution. We denote $f_{signal}(\rho_i)$ and $f_{noise}(\rho_i)$ assigned to the matrix cell $(h,\ k)$ by $f_{signal}(\rho_i, h, k)$ and $f_{noise}(\rho_i, h, k)$, respectively.

Using (3.5) we obtain:

$$A_{measured,i}(h,k) = \frac{f(\text{signal}/\rho_i, h, k)}{f(\text{noise}/\rho_i, h, k)} = \frac{f_{signal}(\rho_i, h, k)}{f_{noise}(\rho_i, h, k)} \tag{3.6}$$

where i is the RN index and $(h,\ k)$ are used to index the matrix or cell entries. Each of the matrices $A_{measured,i}$ indicates where the UN is most likely to be present. Thus by overlaying each of these matrices, one over the other, on the cell grid would give the overall likelihood distribution:

$$A_{overall}(h,k) = \prod_{i=1}^{N} A_{measured,i}(h,k). \tag{3.7}$$

The cell with the maximum valued entry when mapped to the area of interest gives the location estimate (x_s,y_s) of the unlocalized node.

$$(x_s, y_s) \Leftrightarrow (h_s, k_s) = \underset{(h,k)}{\text{argmax}} \left[\prod_{i=1}^{N} \left(\frac{f_{signal}(\rho_i, h, k)}{f_{noise}(\rho_i, h, k)} \right) \right]. \tag{3.8}$$

3.5 Simulation Results

Extensive simulations were carried out using 250 different channel realizations based on the IEEE 802.15.3a channel model 3 (CM3). The transmit signal or ranging pulse is a Gaussian monopulse with a center frequency of 2 GHz and a bandwidth of 3.14 GHz. Noise is assumed to be independent for each signal reception. A 4-bit analog-to-digital converter (ADC) is used in the receiver circuitry and the SNR reported includes the quantization noise effects due to the ADC.

3.5.1 Location Estimation Algorithm using Multiple Signal Receptions

Following the approach in Section 3.3, the UN transmits multiple copies of the ranging pulse and each RN in its radio range records these multiple signal receptions. Each RN then obtains the range estimate to the UN and finally this range information from all RNs is used to estimate the UN location. The use of multiple signal receptions for range estimation reduces the noise variance and helps to eliminate large estimation errors. In Fig. 3.9 we plot the range estimation error using

three signal receptions as a function of the transmitter (TX) - receiver (RX) distance. It is seen that the range estimation error does not increase significantly with the TX-RX distance. TX-RX distances for the simulations ranged from 0-8 m.

The approach in [10] looks for the LOS in a small window of the received signal. Estimating the window size becomes extremely difficult at long ranges due to the complex LOS blockage and this results in large estimation errors. The previous approaches relied on the signal strength for LOS estimation. Since the signal strength decreases with increase in the range which lowers the SNR with a resultant increase in the range estimation errors. In our approach, we are able average out noise and achieve a higher effective SNR by looking at multiple signal receptions. This approach thus works well even for larger TX-RX distances.

In Fig. 3.10, the range estimation error resulting from the approach using multiple signal receptions (with $S = 3$) is compared with the method where only a single signal reception is used. The comparison is shown for different values of the SNR. For 10-15 dB SNR, the average and rms range estimation errors are smaller than 0.2 m using three signal receptions.

Fig. 3.11 is a histogram of the range estimation error showing the number of range errors of different magnitudes. The peak vertical bar in the histogram plot corresponds to around 220 range estimates with an error of less than 0.1 m. It is also seen that there are relatively very few large errors.

Next consider the situation illustrated in Fig. 3.7(b). The unlocalized node position is estimated using 5 range estimates:

$$\begin{aligned}
\text{Estimated ranges} &= \{4.8900, 0.2850, 1.9200, 0.1950, 3.5400\} \text{ m} \\
\text{True ranges} &= \{3.4350, 0.2700, 1.9050, 0.2400, 3.5050\} \text{ m} \\
\text{Location Estimate} &= (-0.0174, -0.0100) \text{ m} \\
\text{True Location} &= (0.0, 0.0) \text{ m}.
\end{aligned}$$

The location estimation algorithm (3.3) found the constraints due to the estimated ranges to be infeasible and thus an attempt was made to solve with 4 constraints. Eliminating the constraint due to the first estimate, which is a NLOS estimate, results in a feasible solution. Thus, even if the ranging algorithm reports NLOS estimates, the localization algorithm is able to identify and eliminate them while estimating the location. The sub-meter ranging and localization accuracy achieved here is better than other approaches, using similar bandwidths, reported in the literature [11, 10].

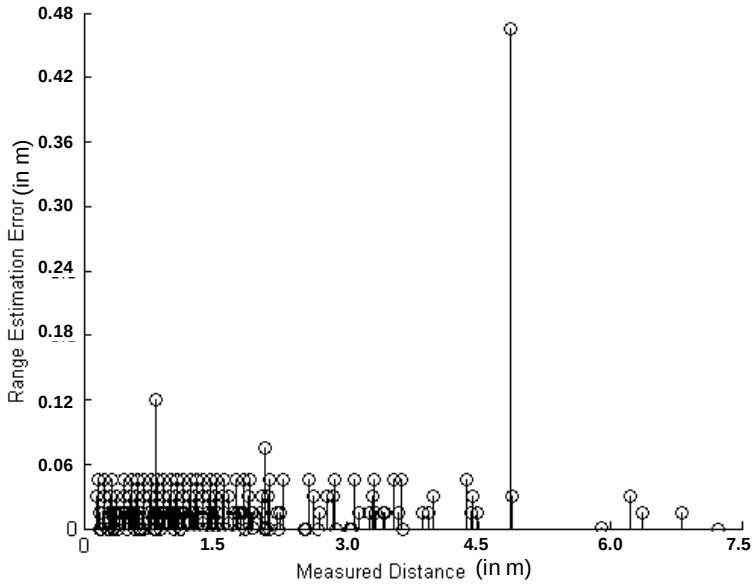


Figure 3.9: Range estimation error vs. TX-RX distance using multiple signal receptions.

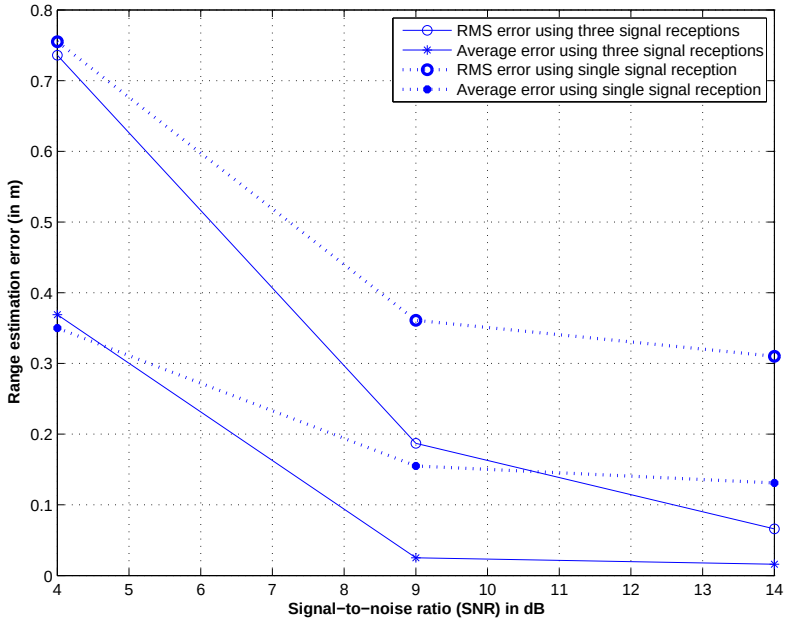


Figure 3.10: Range estimation error vs. SNR using multiple signal receptions.

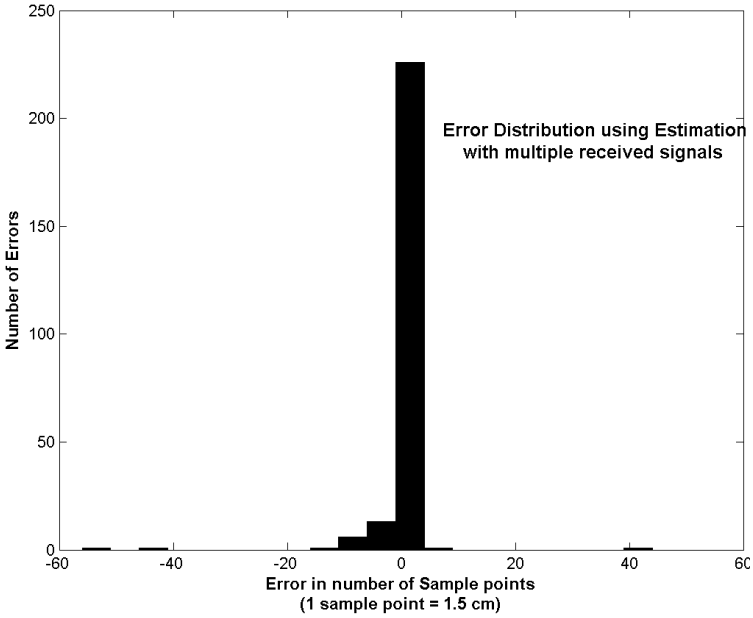


Figure 3.11: Range estimation error histogram.

3.5.2 Spatial Domain Quasi-Maximum Likelihood Location Estimation Algorithm

The simulations in this sub-section use pseudo range estimates from 5 RNs for estimating the UN location. The RN coordinates are randomly generated on a square grid and the SNR is fixed at 10 dB. This localization experiment is repeated 50 times. The simulation results from 5 such trials, each using independent channel realizations, are shown in Fig. 3.12. The true UN location is (0, 0) m. Solid lines indicate the error between the estimated and true UN locations.

Fig. 3.13 shows the average and rms position estimation error, based on all 50 trials, as a function of the SNR. It is seen that the localization accuracy improves with SNR and there is no significant degradation in the accuracy even when the SNR is reduced to 5 dB.

The spatial domain QML approach does not try to locate the LOS position in each received signal. Rather we combine the pseudo range estimates from a number of RNs and the final location estimate corresponds to the cell with the maximum likelihood of containing a signal component. This works well because it is highly unlikely that a majority of the RNs would report NLOS estimates which overlap at a given point (or cell).

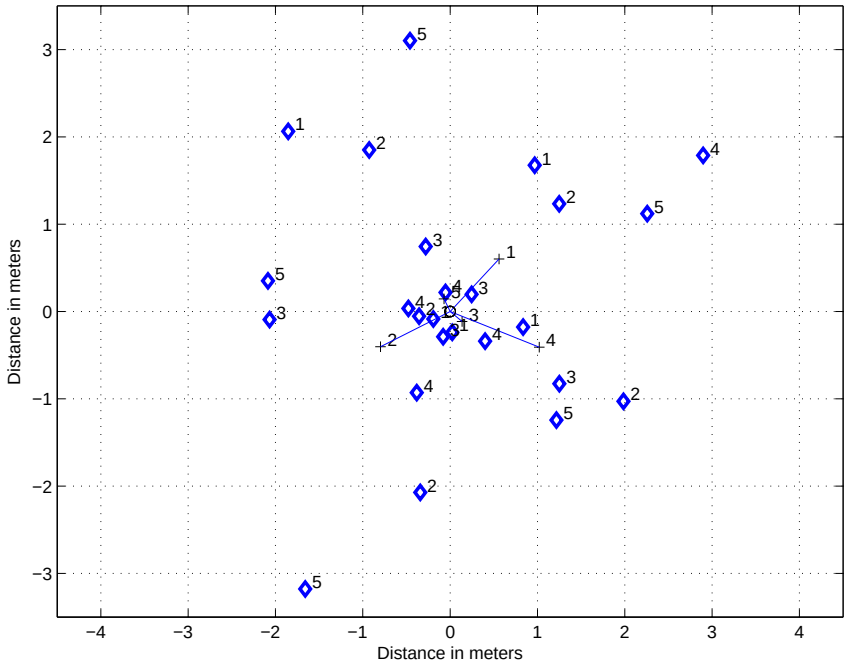


Figure 3.12: Position estimation results from 5 trials using the spatial domain QML location estimation algorithm. True locations of the RN ($\diamond$) and UN ($\circ$) along with the estimated UN locations ($+$). Solid lines indicate the error between the estimated and true UN locations. The node locations for each of the 5 trials are numbered distinctly.

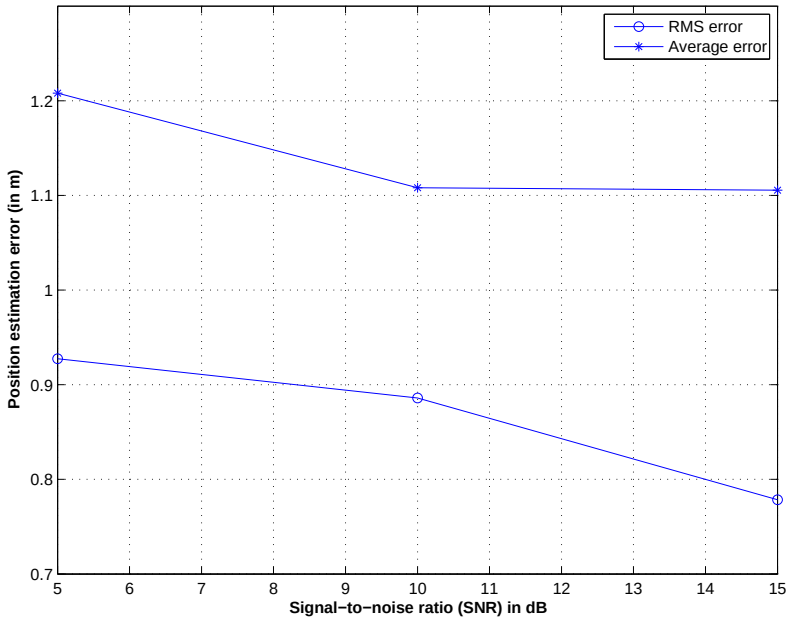


Figure 3.13: Position estimation error vs. SNR using spatial domain QML location estimation algorithm.

3.6 Conclusion

In this chapter we presented two localization algorithms based on statistical modeling of the UWB multipath channel in terms of signal (LOS/NLOS) and noise characterization. The first algorithm uses multiple signal receptions for range estimation. Simulation results have shown that this approach gives sub-meter ranging accuracy even for large TX-RX ranges. The accuracy achievable is limited only by the size of the time bin chosen and the SNR. The bin size is related to the time resolution achievable with the ranging signal being used, which in turn depends on its bandwidth. In the absence of sufficient bandwidth and/or SNR, collecting more number of signal receptions for each range estimate and increasing the recording time for the received signal can improve the ranging and localization accuracy.

The second algorithm, based on spatial domain quasi-maximum likelihood estimation, fuses the range estimation and location estimation phases such that the geometric relationship of the TOAs aids the range estimation phase as well. This approach while being computationally efficient has shown the ability to provide close to sub-meter localization accuracy. Again the accuracy achievable is limited only by the cell size chosen and the SNR. The cell size is related to the time resolution achievable with the ranging signal being used, which in turn depends on its bandwidth. Also, by not trying to directly estimate the LOS position in each of the received signals, we overcome the difficult problem of NLOS identification.

Chapter 4

Distributed Sensor Network Localization using SOCP Relaxation

In this chapter we present a distributed approach based on second-order cone programming (SOCP) for solving the sensor network localization problem. It is assumed that the pairwise range or distance estimates are available (obtained using a suitable ranging method). Here we focus on the positioning sub-problem and thus the distributed solution presented here is independent of the ranging method used. The distributed SOCP approach is computationally efficient and scalable to large network sizes. In addition, we demonstrate the ability to handle distance (or range) estimation errors and errors in the reference node (RN) positions. In the presence of distance estimation errors the approach consists of a single step wherein each unlocalized node (UN) determines its position by executing the localization algorithm independently using distance information to the RNs and other nodes with which it is directly linked (i.e., which are within its communication range). If in addition to the distance estimation errors, the RN positions also have errors then the approach consists of three steps: using the local distance information and inaccurate RN positions each UN estimates its position. Then, the RNs execute the localization algorithm using position information from their neighboring nodes and the associated distance information to refine their positions. Finally, the previously UNs re-execute the localization algorithm to refine their position

estimates.

One of the significant advantages of our approach is that it is fully distributed and converges to an optimal (or near-optimal) solution. As a result of the distributed nature of the solution, the problem dimension at each node is a linear function of only the number of neighbors of the node. There is no significant increase in the computational effort per node even in large networks (for a given node connectivity level), whereas most existing methods result in an exponential increase in the computation time with network size. This makes the distributed SOCP approach suitable for large-scale networks with thousands of nodes. As we will demonstrate, the performance gains are achieved without sacrificing localization accuracy. We demonstrate the convergence of our algorithm numerically.

4.1 Sensor Network Localization: Problem Formulation

The sensor network localization problem is mathematically formulated as follows. Consider n distinct points (or nodes) in R^d ($d \geq 1$). Given the positions of the last $(n-m)$ points (or RNs) $x_{m+1}, ..., x_n$ and the Euclidean distance d_{ij} between neighboring points (or nodes) i and j where $(i,j) \in A$. A is the neighbor set defined as $A = \{(i,j) :\| x_i - x_j \| \leq RadioRange\}$ [1], we need to estimate the positions of the first m points (or UNs). In the presence of distance estimation errors, this can be formulated as the following non-convex minimization [64]:

$$\min_{x_1,...,x_m} \sum_{(i,j)\in A} | \, \|x_i - x_j\|^2 - d_{ij}^2 \, | \tag{4.1}$$

where $\| \cdot \|$ denotes the Euclidean norm. If the distance estimation errors are assumed to be independent and normally distributed, it can be shown that (4.1) is closely related to the maximum likelihood estimation of the node positions [65].

The original problem (4.1) can be reformulated in convex form using relaxation techniques. As a first step, (4.1) can equivalently be written as:

$$\min_{x_1,...,x_m, y_{ij}} \sum_{(i,j)\in A} | \, y_{ij} - d_{ij}^2 \, | \;\; \text{s.t. } y_{ij} = \|x_i - x_j\|^2, \forall (i,j) \in A.$$

Relaxing the equality constraints to "greater than or equal to" inequality constraints yields the

[1] The set A is undirected: $(i,j) = (j,i), \forall (i,j) \in A$

following convex problem:

$$\min_{x_1,\ldots,x_m,y_{ij}} \sum_{(i,j)\in A} |\, y_{ij} - d_{ij}^2 \,| \;\text{ s.t. } y_{ij} \geq \|x_i - x_j\|^2, \forall (i,j) \in A \tag{4.2}$$

which is an SOCP. The problem in (4.2) can equivalently be written as:

$$\begin{aligned} \min_{x_1,\ldots,x_m,y_{ij},t_{ij}} & \sum_{(i,j)\in A} t_{ij} \\ \text{s.t. } & y_{ij} \geq \|x_i - x_j\|^2 && \forall (i,j) \in A \\ & t_{ij} \geq |\, y_{ij} - d_{ij}^2 \,| && \forall (i,j) \in A. \end{aligned} \tag{4.3}$$

One approach would be to solve the SOCP problem as a global minimization over the entire network. However, due to the problem sizes encountered in sensor networks solving the SOCP relaxation *globally* might be computationally demanding as shown in [36]. The distributed approach has the advantage that the algorithm can be executed on the sensor nodes themselves thus removing the need to relay all the information to a central processor. A parallel or distributed algorithm is preferred in sensor networks which we propose next.

4.2 Distributed SOCP Localization Algorithm

In a distributed algorithm, implemented over multiple processors, the algorithm is divided into "phases". During each phase, every processor must execute a number of computations that depend on the results of the computations of other processors in previous phases. However, the timing of computations at any one processor during a phase can be independent of the timing of the computations at other processors within the same phase. All interactions between processors take place at the end of the phases. Such distributed algorithms are also called *synchronous*. Here we show how the SOCP relaxation for the sensor network localization problem can be formulated as a synchronous distributed algorithm [66].

We can approximately reformulate (4.2) as:

$$\min_{x_1,\ldots,x_m,y_{ij}} \sum_{(i,j)\in A} \{|\, y_{ij} - d_{ij}^2 \,| + I_-(f_{ij}(x,y))\}$$

$$\text{where } f_{ij}(x,y) = \|x_i - x_j\|^2 - y_{ij} \text{ and } I_-(u) = \begin{cases} 0 & u \leq 0, \\ \infty & u \geq 0. \end{cases}$$

Here I_- is the indicator function which can be approximated by the *logarithmic barrier* function and the problem reduces to:

$$\min_{x_1,\ldots,x_m,y_{ij}} \sum_{(i,j)\in A} \{|\, y_{ij} - d_{ij}^2 \,| - (1/t)\log(y_{ij} - \|x_i - x_j\|^2)\}. \tag{4.4}$$

It is seen that the objective function in (4.4) is separable. For each $i \in \{1, ..., m\}$, the objective function depends only on the positions of the neighboring nodes and the pairwise distance measurements between them. This enables the objective function to be decomposed and the minimization can then be carried out at each node x_i using local information. Each UN will be able to independently solve this minimization using information from its neighboring nodes and RNs.

Let $N_A(i) = \{j : (i,j) \in A\}$ be the neighbor set for node x_i. Using the separability observation, (4.3) can be solved independently over the m UNs x_i $(i = 1, \ldots, m)$, where each node uses information $(x_j,\, d_{ij})$ from its neighboring nodes x_j, $j \in N_A(i)$. The information exchange between nodes occurs at the end of each iteration (or phase). Thus (4.3) decomposes to the following distributed formulation:

$$\begin{aligned} \min_{x_i, y_{ij}, t_{ij}} & \sum_{j\in N_A(i)} t_{ij} \\ \text{s.t. } & y_{ij} \geq \|x_i - x_j\|^2 && \forall j \in N_A(i) \\ & t_{ij} \geq |\, y_{ij} - d_{ij}^2 \,| && \forall j \in N_A(i). \end{aligned}$$

This can be written in standard SOCP form as:

$$\begin{aligned} \min_{x_i, y_{ij}, t_{ij}} & \sum_{j\in N_A(i)} t_{ij} \\ \text{s.t. } & \left(\frac{y_{ij} + t_i^{'}}{2}\right)^2 \geq \left(\frac{y_{ij} - t_i^{'}}{2}\right)^2 + \|x_i - x_j\|^2 && \forall j \in N_A(i) \\ & t_{ij} \geq |\, y_{ij} - d_{ij}^2 \,| && \forall j \in N_A(i) \\ & t_i^{'} = 1. \end{aligned} \tag{4.5}$$

The distributed SOCP algorithm consists of a phase where each UN estimates its position using information from the neighboring nodes and solving the SOCP (4.5). In an iterative distributed scheme, this would be followed by a communication phase wherein each node exchanges its position estimate with its neighbors. These iterations could be repeated after fixed intervals of time or when any new information becomes available at a node. It should be noted that the algorithm uses

information from neighboring RNs as well as other neighbor nodes to position a given node. Thus to obtain a non-trivial position estimate each node needs at least 3 neighbors (for 2-D localization) with position estimates, as opposed to the more stringent requirement of having 3 RNs in the neighborhood that many triangulation/trilateration schemes impose.

If the RN positions are inaccurate, the distributed SOCP approach will consist of three steps: after the UNs estimate their positions based on the inaccurate RN positions and distance information, the RNs solve the local SOCP (4.5) using position information from their neighboring nodes and the associated distance information to refine their positions. As we will show, this second step results in a significant improvement in the positioning accuracy of the inner RNs. Finally, another iteration of the local SOCP (4.5) over the previously unlocalized nodes further refines their position estimates.

Let $n_i(=|N_A(i)|)$ represent the number of neighbors of the node x_i. SOCP (4.5) has $2n_i + 3$ variables, $2n_i$ conic constraints and 1 equality constraint. In sensor networks, due to the relatively short radio range of the sensors, the *number of neighbors* of a given node is a small fraction of the total number of nodes in the network (i.e., $n_i \ll n$). Thus the distributed SOCP approach (4.5) results in significantly smaller problem sizes than approaches proposed in the literature. The SOCP problem (4.5) can be efficiently solved in practice by interior point methods. Here we use SeDuMi [67] to solve this problem[2].

4.3 Localization with Accurate Reference Node Position Information

The experiments in this section assume that accurate RN position information is available. In this setting, the localization problem is solved by executing the distributed SOCP algorithm at each of the UNs. We assess the average-case performance on networks with uniform topology as well as irregular topology. For each parameter setting, the algorithm is run on 5 randomly generated examples.

We randomly generate the true positions of the UNs and RNs $x_1^t, \ldots, x_n^t$ according to a uniform distribution on the unit square $[-0.5, 0.5]^2$ and noisy distance measurements d_{ij} are generated by

[2]Refer to Appendix A for details on how to rewrite SOCP (4.5) in the SeDuMi form

adding normally distributed measurement noise to the true distance. Specifically:

$$d_{ij} = \|x_i^t - x_j^t\| \cdot \max\{0, 1 + \epsilon_{ij} \cdot nf_d\} \quad \forall (i,j) \in A$$

$$A = \{(i,j) : \|x_i^t - x_j^t\| \leq RadioRange\}$$

where ϵ_{ij} is a normal random variable $\mathcal{N}(0,1)$ representing measurement noise, $RadioRange \in (0,1)$ represents the radio range of the nodes and $nf_d \in [0,1]$ is the noise factor (standard deviation of the distance error in percentage) for the distance measurements. For a standard deviation of 10% in the distance estimation error we set $nf_d = 0.10$.

We wrote the code in Matlab to solve the SOCP relaxation. Our code calls SeDuMi (Version 1.1) [67], a C implementation of a predictor-corrector primal-dual interior point method for solving SDP/SOCP. The simulations were carried out on a PC with 2.53 GHz Pentium 4 processor and 1 GB RAM running Matlab 7.0.1 (R14).

To check the positioning accuracy of our algorithm, we define the average error as:

$$\overline{err} = \frac{\sum_{i=1}^{m} \|x_i - x_i^t\|}{m}.$$

First we consider the uniform topology. Fig. 4.1 shows the results using the SOCP algorithm for a randomly generated 500 node network with $nf_d = 0.05$. It is seen that there is a close match between the estimated and true positions for nodes which lie in the convex hull of their neighbors. The average positioning error is 0.032 (21.3% of the *RadioRange*). The estimated positions become less accurate as we move towards the boundary.

Fig. 4.2 shows the effect of the percentage of RNs (p) on the average positioning error ($\overline{err}$) for different network sizes with similar node connectivity levels. Increasing p from 12 to 15% lowers the average error. The variation in the average error across network sizes (for a given p) is most likely due to the small differences in the node connectivity levels.

The positioning accuracy improves significantly with increase in the node connectivity level. Fig. 4.3 shows that the average error decreases steadily as *RadioRange* is increased from 0.08 to 0.20. For larger radio ranges the error tends to reach a lower bound determined by the distance estimation errors.

Irregular topologies are more difficult than uniform topologies. The results for an irregular topology, namely a C-shaped network, with 300 nodes and $nf_d = 0.05$ are shown in Fig. 4.4. The average positioning error is 0.048 (32% of the *RadioRange*).

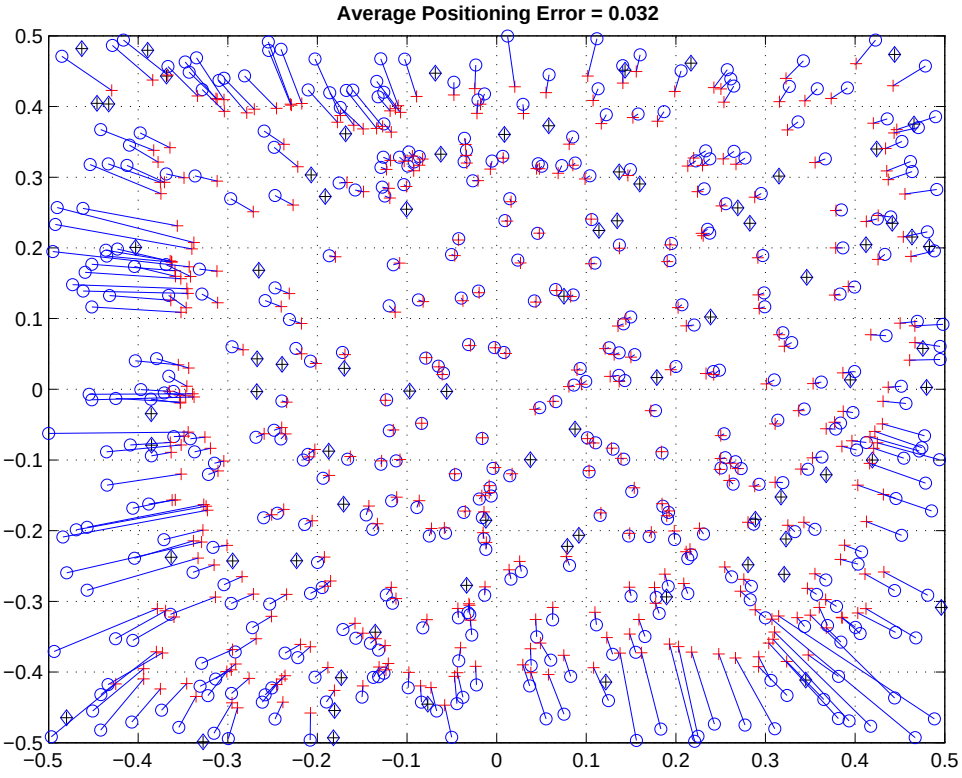


Figure 4.1: Distributed SOCP results for Uniform topology: $n = 500$, $RadioRange = 0.15$, $p = 0.15$ and $nf_d = 0.05$. $\overline{err} = 0.032$ and $err_{max} = 0.232$. True positions of the UNs ($\circ$) and RNs ($\diamond$) along with the estimated node positions ($+$). The solid lines indicate the error between the estimated and true UN positions.

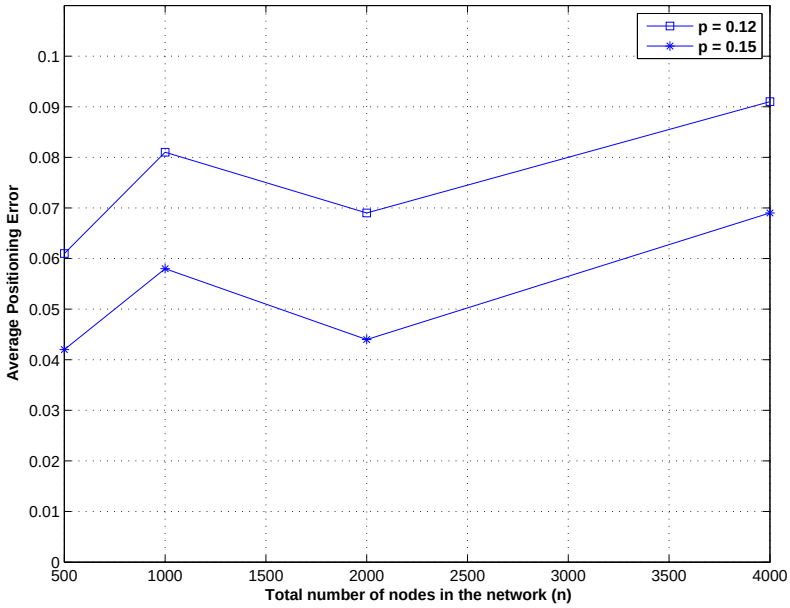


Figure 4.2: Average positioning error as a function of the network size (n) for two different percentages of RNs. ($RadioRange = 0.10$, $nf_d = 0.05$ and average node connectivity ≈ 30)

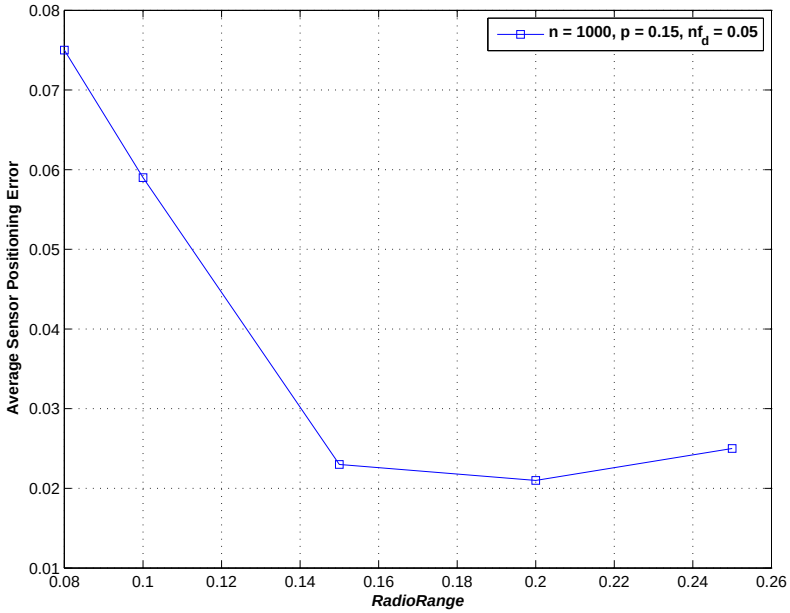


Figure 4.3: Average positioning error as a function of $RadioRange$. ($n = 1000$, $p = 0.15$ and $nf_d = 0.05$)

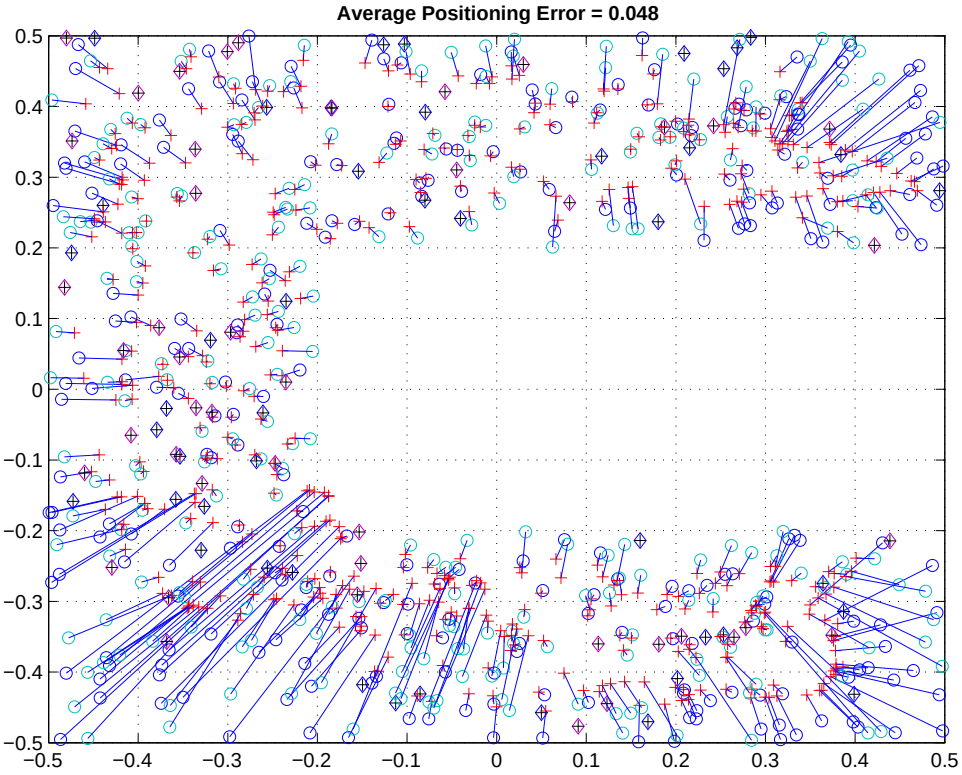


Figure 4.4: Distributed SOCP results for irregular (C-shaped) topology: $n = 300$, $RadioRange = 0.15$, $p = 0.15$ and $nf_d = 0.05$. $\overline{err} = 0.048$ and $err_{max} = 0.448$. True positions of the UNs ($\circ$) and RNs ($\diamond$) along with the estimated node positions ($+$). The solid lines indicate the error between the estimated and true UN positions.

Table 4.1 lists the test cases used to understand the computational complexity of the distributed SOCP algorithm. *RadioRange* is chosen such that the average node connectivity is about the same across all test cases. Table 4.1 lists the input parameter values, the cardinality of A (denoted by $|A|$), the average node connectivity, the typical SOCP (4.5) dimension and the computation time per node (excluding the time needed for computing the relative distances d_{ij} and communication or message exchanges). Comparison with the SOCP dimensions reported by Tseng [36] for similar network sizes reveals that the dimensions in Table 4.1 are smaller by at least two orders of magnitude and are still in the realm of problem sizes which can be handled efficiently, even for a network with 4000 nodes. The SOCP dimension, which depends on the number of neighboring nodes, does not increase with the network size for a given average node connectivity. As a result, the per node computational burden is significantly reduced.

Table 4.2 presents test cases used with the MDS-MAP(P, R) algorithm [2]. The cost of refining and merging the local maps, and the optional global refinement step in the MDS-based algorithm becomes dominant for large networks ($n > 300$). Hence, network sizes for these tests are smaller than those for the distributed SOCP algorithm. However *RadioRange* is chosen such that the node connectivity is approximately the same as for the test cases in Table 4.1.

From Tables 4.1 and 4.2, it is seen that the MDS-MAP(P, R) algorithm requires slightly more than three times the computational effort needed for the distributed SOCP algorithm, even for relatively small network sizes. Thus the distributed SOCP algorithm significantly improves the computational efficiency without sacrificing localization accuracy.

4.4 Localization with Reference Node Position Errors

The experiments in this section consider the inaccuracy in the RN positions in addition to the distance estimation errors. The goal is to localize the UNs while reducing the adverse impact of the RN position errors on the positioning accuracy. One way to achieve this is to solve SOCP (4.5) simultaneously at each of the UNs and RNs. We ran a few simulation test cases using this approach, but the results did not converge in each of those cases. Hence we propose a three-step distributed approach. In the first step, each UN estimates its position using distance information from its neighbors. In the second step, the RNs use information from their neighbors to refine their positions. We also observed that the second step aids in refining only those RN positions which

Test Case	n	*RadioRange*	p	$\|A\|$	Avg. Node Connectivity	SOCP dimension (Typical)	CPU time per node (in sec)
1	500	0.15	0.12	15104	30.2	123×181	1.24
2	500	0.15	0.15	15054	30.2	181×124	1.19
3	1000	0.10	0.12	28368	28.4	115×169	1.20
4	1000	0.10	0.15	28158	28.2	115×169	1.18
5	2000	0.08	0.12	73566	36.8	151×223	1.30
6	2000	0.08	0.15	76706	38.4	155×229	1.29
7	4000	0.05	0.15	119082	29.8	123×181	1.03

Table 4.1: Distributed SOCP: Input parameters for the test cases, corresponding SOCP (4.5) dimensions and CPU times. (p gives the percentage of reference nodes, and Noise figure $nf_d = 0.05$ for all test cases).

Test Case	n	*RadioRange*	p	Avg Node Connectivity	CPU time per node (in sec)
1	100	0.35	0.050	29.6	4.03
2	200	0.25	0.025	32.9	4.57
3	300	0.20	0.017	30.9	4.35
4	500	0.15	0.010	30.8	4.07

Table 4.2: MDS-MAP(P, R): Input parameters for the test cases and CPU times. (p gives the percentage of reference nodes, and Noise figure $nf_d = 0.05$ for all test cases).

are within the convex hull of their neighbors (or in the interior of the network). Thus the RN refinement step is applied only for RNs in the interior of the network. Finally, we repeat one more iteration of the distributed SOCP algorithm over the previously unlocalized nodes using the refined RN positions. It is seen that the refined RN positions improve the UN positioning significantly.

The simulation setting is similar to that in section 4.3 except for the following differences. The noisy distance measurements and inaccurate RN positions are generated as:

$$d_{ij} = \|x_i^t - x_j^t\| \cdot \max\{0, 1 + \epsilon_{ij} \cdot nf_d\} \quad \forall (i,j) \in A$$

$$x_i = x_i^t \cdot \max\{0, 1 + \epsilon_{ij} \cdot nf_a\} \quad \forall i = (m+1), \ldots, n.$$

where nf_a and $nf_d \in [0, 1]$ are the noise factors for RN positions and distance measurements, respectively. Simulations in this section were carried out on a PC with 3 GHz Pentium 4 processor and 2 GB RAM running Matlab 7.2.0 (R2006a).

We experimented with the noise factors (nf_d and nf_a) to understand their effect on the positioning accuracy. Fig. 4.5 shows the variation of the average error ($\overline{err}$) with increasing nf_d. The distributed SOCP algorithm handles distance errors as large as 20% gracefully with a small degradation in positioning accuracy. MDS-MAP(P, R) algorithm gives better accuracy using the $O(n^3)$ global refinement step and shows similar degradation. It should be noted that MDS-MAP(P), the MDS-MAP algorithm without global refinement, shows loss in performance compared to MDS-MAP(P, R). MDS-MAP(P) also uses more computations than the distributed SOCP due to the merging and refinement of local maps. It is observed that the singular value decomposition (SVD) step of the MDS-MAP algorithms takes progressively longer to converge as the distance errors increase. Localization systems based on received signal strength (RSS) measurements regularly encounter distance estimation errors of $15 - 20\%$. The distributed SOCP approach is thus robust to large distance errors while being computationally efficient.

Fig. 4.6 shows the effect of the noise factor nf_a on the average error. The distributed SOCP algorithm is robust to RN position errors with results comparable to the MDS-MAP(P, R) algorithm. The degradation in the positioning accuracy is not significant even in the presence of 30% error in the RN positions. MDS-MAP(P, R) algorithm which typically works with very few RNs, needs more RNs to compensate for the errors introduced by the inaccurate RN positions.

In Section 4.3 we showed the improvement in the positioning accuracy achievable as a result of increasing the radio range for all the nodes. Since increasing the radio range of all nodes is

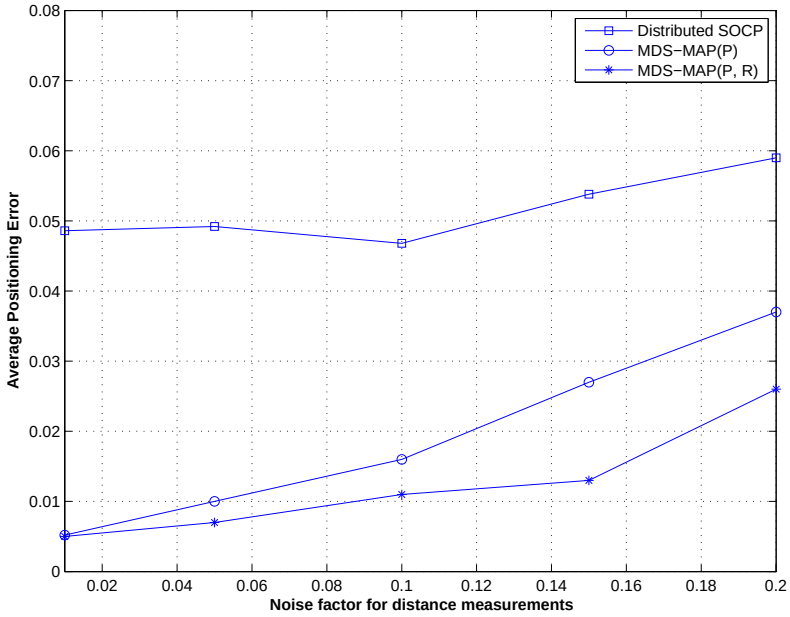


Figure 4.5: Average positioning error as a function of the Noise Factor nf_d. ($n = 500$, $RadioRange = 0.15$, $p = 0.15$ and $nf_a = 0.10$)

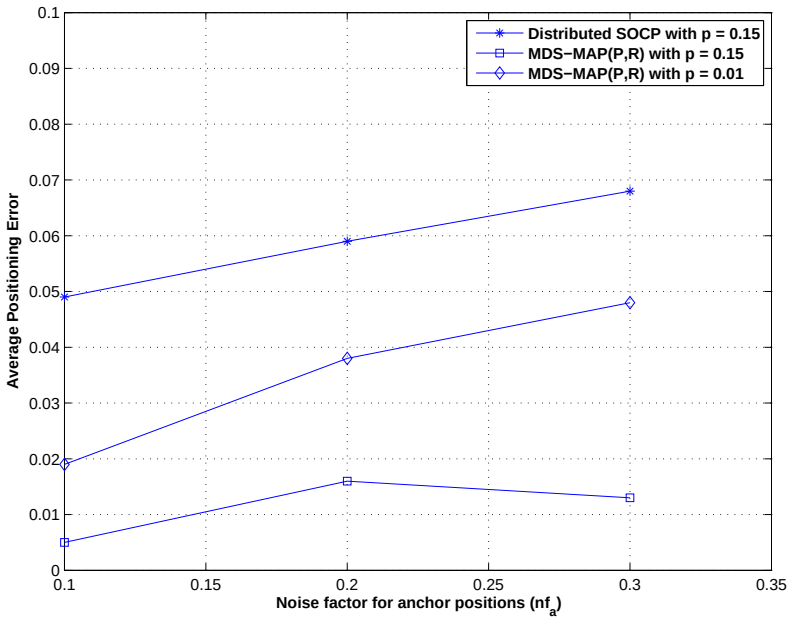


Figure 4.6: Average positioning error as a function of nf_a and p. ($n = 500$, $RadioRange = 0.10$ and $nf_d = 0.01$)

typically not feasible due to the resource constraints, increasing the radio range for the RNs is more reasonable. Also, in reality RNs tend to have more transmit power and a longer range. Fig. 4.7 illustrates the effect of increasing the radio range of the RNs while keeping the range of the other nodes fixed. Increasing the radio range of the RNs from 0.10 to 0.15 results in a very significant improvement in the UN positioning accuracy, but further increase in the RN radio range does not improve the positioning. This behavior can be explained as follows. During position estimation for a UN, we use information from only those nodes and RNs which are within its radio range. We do not account for the fact that some RNs are able to transmit to a UN even though the UN cannot transmit to those RNs (due to asymmetric radio ranges). In effect, we use information from only those nodes which have a bidirectional link with the UN. So the increased range of the RNs only aids in refining the position estimates of the RNs when they communicate with other RNs, which improves the positioning accuracy of the UNs, but with diminishing return. Further improvement in positioning accuracy can be expected by using information from all RNs which can transmit to a given UN.

We now revisit the irregular C-shaped network and use longer ranges for the RNs than the other nodes in the network. Radio range of the RNs is labeled as $RadioRangeRef$ and the radio range of all other nodes is labeled as $RadioRangeSensor$. Fig. 4.8 shows the positioning result when the radio range is 0.20 for the RNs and 0.15 for all other nodes. The average positioning error is 0.031 (20.6% of the $RadioRangeSensor$), a 10% improvement over the accuracy achieved with the radio range set to 0.15 for both RNs and other nodes. Fig. 4.9 shows the results with the RN range increased to 0.25 resulting in an average positioning error of 0.025 (16.7% of the $RadioRangeSensor$).

4.5 Tracking a Mobile Sensor Node

In this section we consider the scenario where the goal is to track a mobile sensor node in an indoor environment using a few high-power and long range RNs. This arises in situations such as tracking fire-fighters, who are on a rescue mission, inside a building on fire. The fire-fighters will have wearable sensors which can be tracked using high-power transmitters placed outside the building.

To make the experiments more realistic, we include a fading coefficient (f), which represents

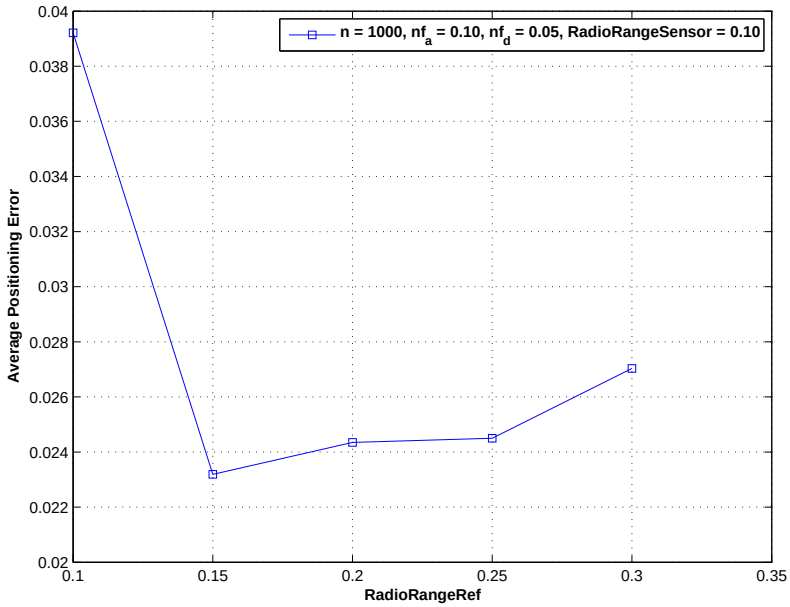


Figure 4.7: Average UN positioning error as a function of the radio range of the reference nodes ($RadioRangeRef$). ($n = 1000$, $p = 0.15$, $nf_d = 0.05, nf_a = 0.10, RadioRangeSensor = 0.10$)

the fraction of all RNs which cannot be heard by the mobile node at any given time. This models the obstructions encountered in indoor environments which limit the number of RNs that can be heard at any point.

The RNs are placed equidistantly at the boundary of a $[-0.5, 0.5]^2$ square grid. We use 8 RNs for the experiments. The tracking results of our algorithm under different parameter settings are shown in Figs. 4.10 and 4.11. In Fig. 4.10 the distance estimates have $\pm 15\%$ error standard deviation ($nf_d = 0.15$) whereas in Fig. 4.11 $nf_d = 0.10$. There is no significant degradation in the results as the error standard deviation in the RN positions was increased from 5% to 15% ($nf_a = 0.05$ to 0.15). Thus we kept nf_a fixed at 0.10 for all the tracking results shown here. The estimated tracks are fairly accurate with up to 15% error standard deviation in the distance estimates. Distance estimates with more than 20% error standard deviation begin to degrade the results. The various error metrics for these experiments are shown in Table 4.3.

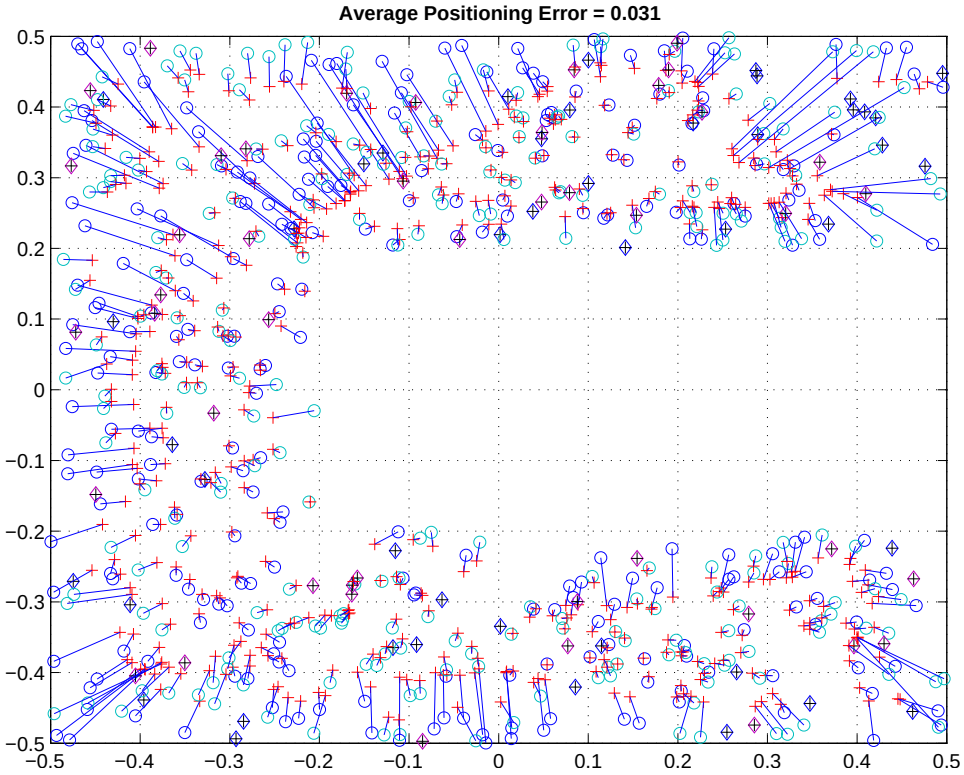


Figure 4.8: Distributed SOCP results for irregular (C-shaped) topology: $n = 300$, $RadioRangeSensor = 0.15$, $RadioRangeRef = 0.20$, $p = 0.15$ and $nf_d = 0.05$. $\overline{err} = 0.031$ and $err_{max} = 0.444$. True positions of the UNs ($\circ$) and RNs ($\diamond$) along with the estimated node positions ($+$). The solid lines indicate the error between the estimated and true UN positions.

Test Run	nf_d	f	Number of RNs Heard	Average Error	Error Standard Deviation	Maximum Error
1	0.15	0.50	4	0.094	0.043	0.182
2	0.15	0.40	5	0.075	0.026	0.152
3	0.10	0.50	4	0.061	0.038	0.169
4	0.10	0.40	5	0.044	0.029	0.117

Table 4.3: Simulation parameters for the tracking results. f is the fading coefficient, total number of RNs $= 8$, $nf_a = 0.10$ for all test runs.

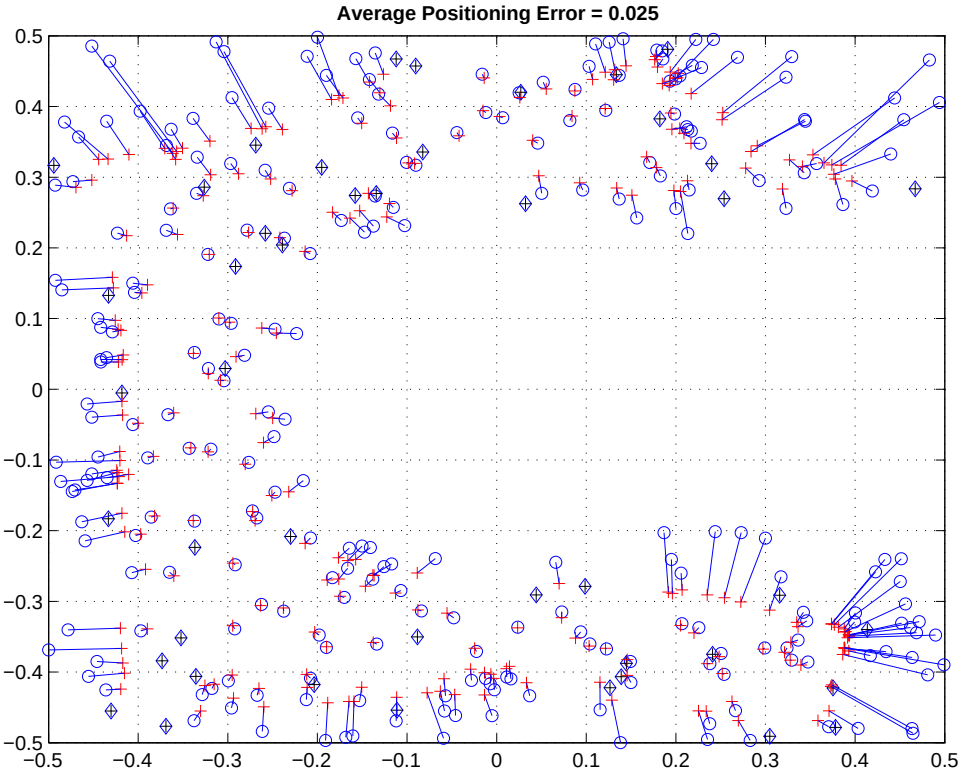


Figure 4.9: Distributed SOCP results for irregular (C-shaped) topology: $n = 300$, $RadioRangeSensor = 0.15$, $RadioRangeRef = 0.25$, $p = 0.15$ and $nf_d = 0.05$. $\overline{err} = 0.025$ and $err_{max} = 0.429$. True positions of the UNs ($\circ$) and RNs ($\diamond$) along with the estimated node positions ($+$). The solid lines indicate the error between the estimated and true UN positions.

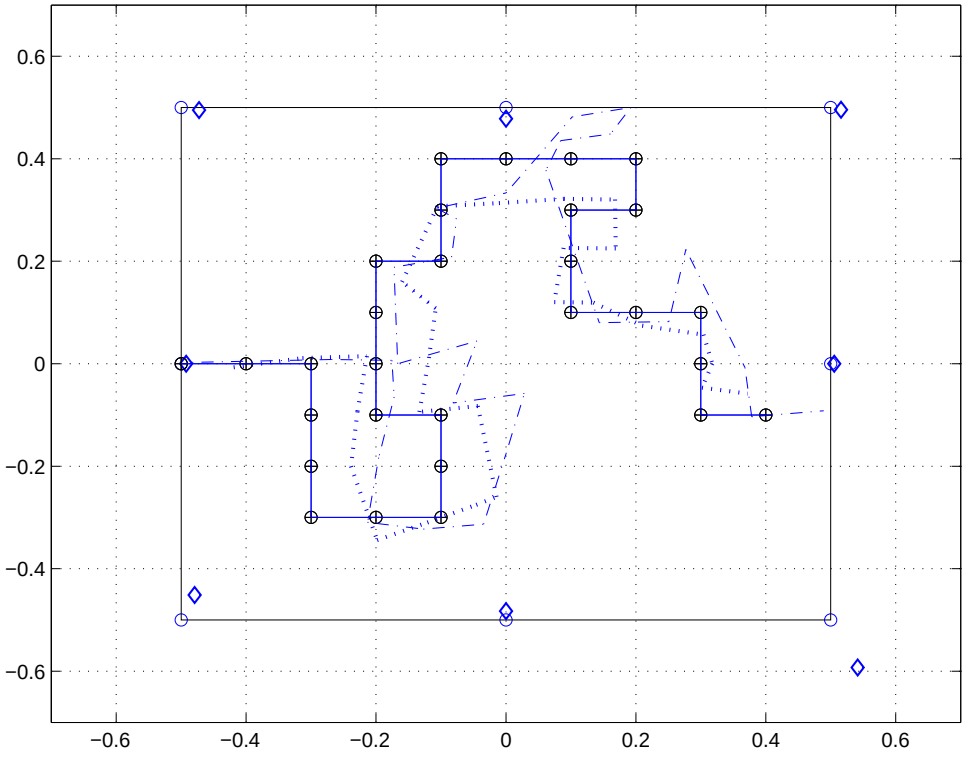


Figure 4.10: Tracking Results: Circles ($\circ$) on the $[-0.5, 0.5]^2$ square grid represent the true RN positions, the diamonds ($\diamond$) represent the inaccurate RN positions used for the experiments. $\oplus$ indicates mobile node positions along the actual path. Solid lines indicate the actual path followed by the mobile node. Estimated paths are indicated by dash-dot lines for test run 1 and dotted lines for test run 2. (Test Run 1: $nf_d = 0.15$, $f = 0.50$. Test Run 2: $nf_d = 0.15$, $f = 0.40$)

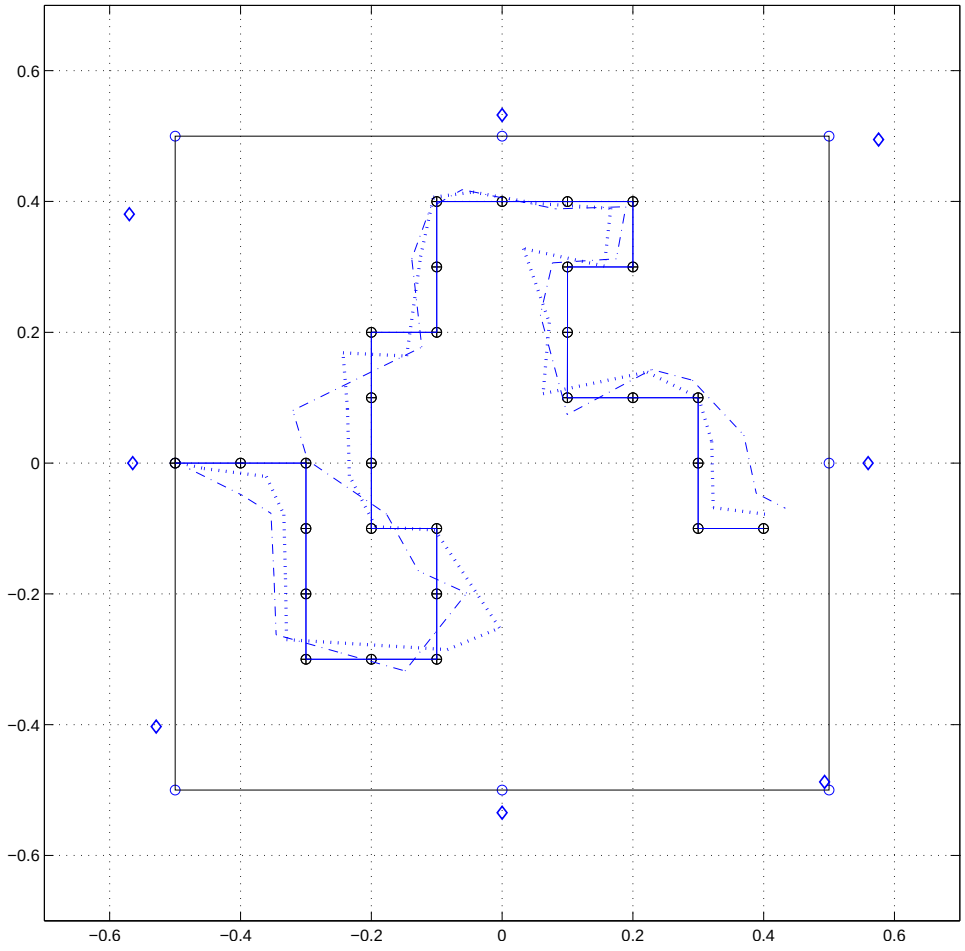


Figure 4.11: Tracking Results: Circles ($\circ$) on the $[-0.5, 0.5]^2$ square grid represent the true RN positions, the diamonds ($\diamond$) represent the inaccurate RN positions used for the experiments. $\oplus$ indicates mobile node positions along the actual path. Solid lines indicate the actual path followed by the mobile node. Estimated paths are indicated by dash-dot lines for test run 3 and dotted lines for test run 4. (Test Run 3: $nf_d = 0.10$, $f = 0.50$. Test Run 4: $nf_d = 0.10$, $f = 0.40$)

4.6 Asynchronous Distributed Algorithm

In this section we consider asynchronous execution of the distributed SOCP algorithm. The motivation for this is to understand the convergence properties of the algorithm under asynchronous execution, and to compare the time to convergence and communication penalty with its synchronous counterpart. We also demonstrate a variation of the asynchronous algorithm that allows the distributed SOCP approach to be used with fewer RNs.

4.6.1 Synchronous vs. Asynchronous

Asynchronous algorithms lack the notion of phases and coordination between the different processors is less strict. To simulate asynchronous execution for the SOCP algorithm, we randomly pick the nodes which will update their positions during any given iteration. Each node localizes itself based on whatever information happens to be available from its neighbors at that time; including some information which may not have been updated for the last few iterations.

Table 4.4 presents a representative comparison between the synchronous and asynchronous executions of the algorithm. It is seen that the asynchronous algorithm needs about three times as many iterations to achieve the same positioning accuracy as the synchronous version. Despite this fact, the computational time per node (excluding the time for communication) increases by a much smaller factor. The communication requirements for the asynchronous version might exceed those for the synchronous execution due to the larger number of iterations resulting in more message exchanges between nodes. Note that the tradeoff between computation time and the number of messages assumes that there are no queuing delays affecting the computation. Considering the results in Table 4.4 it can be said that the asynchronous algorithm converges at about the same rate as the synchronous algorithm. This agrees with the analysis of other well-known algorithms such as the Bellman-Ford algorithm.

4.6.2 Localization with fewer RNs using the asynchronous algorithm

One of the advantages of a distributed iterative localization algorithm is the ability to use the nodes positioned in one iteration as *pseudo* RNs in the following iterations. The distributed SOCP algorithm makes use of this implicitly. Here we outline a procedure to make explicit use of this, thus allowing the algorithm to localize the nodes using fewer RNs.

	Total Number of Iterations	Computational time per node (in sec)	Average Position Error	
			Before RN Refinement	After RN Refinement
Synchronous	20	1.91	0.0436	0.0232
Asynchronous	60	2.67	0.0447	0.0213

Table 4.4: Comparison of Synchronous and Asynchronous Algorithm Execution. Simulation parameters: $n = 1000$, $nf_d = 0.05$, $nf_a = 0.10$, $p = 0.15$, $RadioRangeRef = 0.15$, $RadioRangeSensor = 0.10$.

In a variation of the asynchronous algorithm, instead of randomly choosing nodes which update their positions in a given iteration, the position update will be based on the availability of at least three neighboring nodes which have obtained an estimate of their position.

Fig. 4.12 shows the positioning results for a network with $n = 500$ nodes and 5% RNs (or 25 RNs). The RNs are placed on a uniform grid to ensure good coverage. It is seen that the results converge giving an average positioning error of 0.0346 (or 21.6% of the $RadioRangeSensor$). This illustrates the ability of the distributed SOCP approach to localize nodes with fewer RNs, without loss of positioning accuracy.

Finally, in Table 4.5 we provide a representative comparison of the distributed SOCP, dwMDS [1] and MDS-MAP algorithms [2] in terms of performance and complexity, on networks with a low percentage of RNs. Some of the parameter values are estimated based on the data presented in [2, 1]. dwMDS and distributed SOCP are similar in computational complexity, however the distributed SOCP method provides better accuracy. The results using MDS-MAP algorithms, despite using $O(n^3)$ operations, do not differ significantly from the distributed SOCP algorithm.

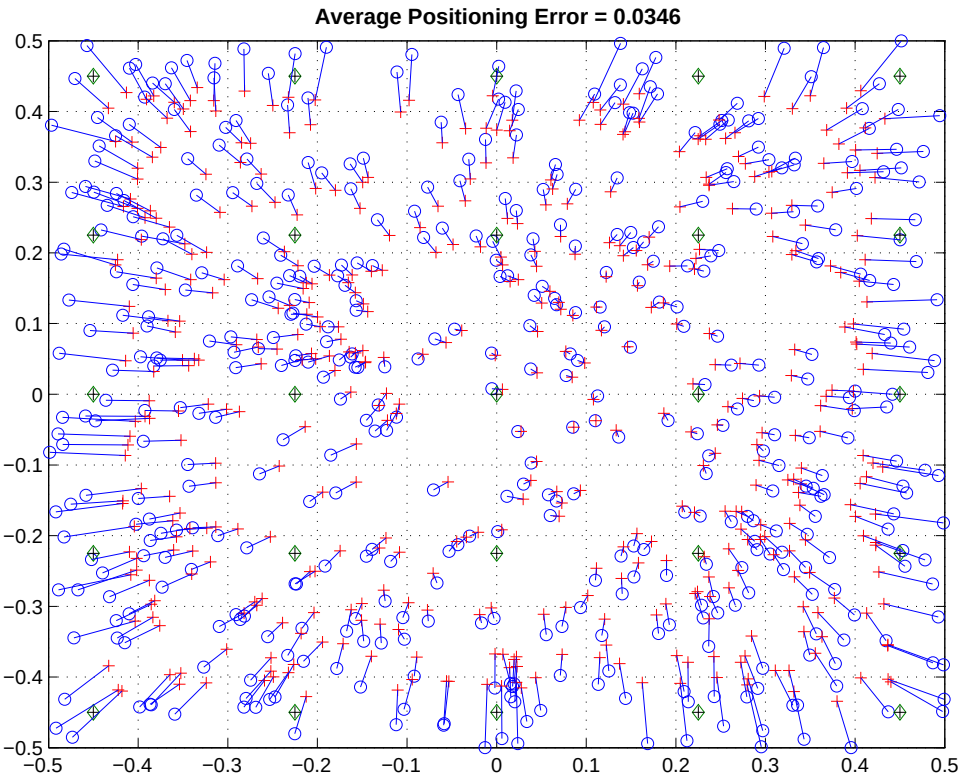


Figure 4.12: Asynchronous distributed SOCP using low percentage of RNs: $n = 500$, $p = 0.05$, $RadioRangeRef = 0.16$, $RadioRangeSensor = 0.16$, $nf_d = 0.10$ and $nf_a = 0$. Positioning error $\overline{err} = 0.0346$. True positions of the UNs ($\circ$) and RNs ($\diamond$) along with the estimated node positions ($+$). The solid lines indicate the error between the estimated and true UN positions.

	n	nf_d	p	*RadioRange*	Avg Node Connectivity (n_c)	Position Error (% *RadioRange*)			Computational Complexity
						Mean	Median	RMS	
MDS-MAP(C)†	44	0.30	0.09	0.43	–	–	–	71.7%	$O\left(n^3\right)$
dwMDS †	44	0.30	0.09	0.43	–	–	–	41.3%	$O\left(nn_cL\right)$
	44	0.30	0.09	0.61	–	–	–	26.7%	
Dist. SOCP	44	0.30	0.09	0.43	13.0	27.6%	29.2%	30.7%	$O\left(nn_cL\right)$
	44	0.30	0.09	0.61	24.6	24.1%	23.2%	26.4%	
MDS-MAP(C)‡	200	0.05	0.05	0.15	12.2	–	17%	–	$O\left(n^3\right)$
MDS-MAP(P, R)‡	200	0.05	0.05	0.15	12.2	–	6%	–	$O\left(nn_c^3\right) + O\left(n^3\right)$
Dist. SOCP	200	0.05	0.08	0.22	21.4	11.6%	10.5%	12.7%	$O\left(nn_cL\right)$

Table 4.5: Comparison of different localization algorithms on random Uniform networks. L represents the number of iterations needed for the algorithm to converge (typically a small number). † Experimental data and results from [1]. ‡ Simulation results from [2].

4.7 Conclusion

The proposed distributed algorithm based on SOCP relaxation solves the sensor network localization problem, in the presence of inaccuracies in the RN positions and distance measurements, with significant computational savings and without sacrificing positioning accuracy. An extensive numerical study of the algorithm under different scenarios has been presented. This method is also able to improve positioning of the RNs which are in the convex hull of their neighbors. The asynchronous version of the algorithm also shows good convergence properties and allows localization with fewer RNs.

Chapter 5

Implementation of a Directional Beacon based Positioning Algorithm

In this chapter we discuss the implementation of a directional beacon (DB) based positioning algorithm using narrowband radio frequency (RF) signals. We develop a signal processing framework to work with the low-accuracy experimental data obtained using low-cost hardware. The positioning algorithm allows each unlocalized node (UN) to compute its position with respect to a set of reference nodes (RNs) which are equipped with rotational directional antenna. The system implementation is based on the GNU Radio software platform and uses Universal Software Radio Peripheral (USRP) as the hardware component. Even though the technique needs some modification at the RNs in the form of a rotational directional antenna, we show that the UNs do not need hardware modifications. In contrast to positioning algorithms that rely on the bandwidth and propagation time of the transmit signal, the positioning algorithm used here depends on the width of the antenna beampattern and the rotational speed of the directional antenna. Our low-cost implementation using directional antenna and stepper motor provides sub-meter positioning accuracy. Another important advantage of this algorithm is that it does not require any synchronization between the RNs and the UN. However, the RNs are assumed to be synchronized. We use intelligent techniques such as maximum likelihood (ML) based amplitude estimation and least

squares based line-sf-sight (LOS) time-delay estimation, to estimate the bearing of the UN with respect to each of the RNs in the presence of multipath components due to reflection etc. We also demonstrate a technique to combine received signal data from multiple transmit antenna rotations to improve the estimation accuracy. These techniques allow us to obtain enhanced position estimates with sub-meter accuracy.

5.1 Localization Principle, Signal Model and Positioning Algorithm

5.1.1 Localization Principle and System Model

This section summarizes the localization principle that is the basis for the directional beacon (DB) based positioning algorithm [43]. We refer to the nodes whose positions are known *a priori* as the reference nodes (RNs) and nodes whose positions are unknown as the unlocalized nodes (UNs). Consider a wireless network that contains three reference nodes RN-1, RN-2 and RN-3. The RNs can be located at arbitrary but known positions. For simplicity, we consider that the RNs are located at the corners of a rectangular field as shown in Fig. 5.1. We further assume that the co-ordinate system origin is at RN-3. Now, consider the situation where a unlocalized node Q joins the network and needs to determine its coordinates (x, y) with respect to the RNs.

We now describe the DB based algorithm which will allow node Q to determine its coordinates. This algorithm requires each RN to be equipped with a rotating directional antenna. The hardware implementation of the rotating antenna is discussed in the next section. The localization principle is based on observing the times when node Q receives the different beacon signals, and estimating its angular bearings and location with respect to the RNs by triangulation [44, 45]. It is necessary that the transmissions from different RNs are distinguishable at Q. This may be achieved, for example, by using different frequencies or coded sequence of pulses for each beacon.

If the times at which Q receives beacons from RN-1, RN-2 and RN-3 are t_1, t_2 and t_3, respectively, the bearings of Q (refer to Fig. 5.1) can be obtained as:

$$\begin{aligned} \alpha &= \phi_1 - \omega(t_2 - t_1) \\ \beta &= \phi_2 - \omega(t_3 - t_2) \end{aligned} \tag{5.1}$$

where ω is the angular speed of the rotating directional beam in rad/s, and ϕ_1 and ϕ_2 are the

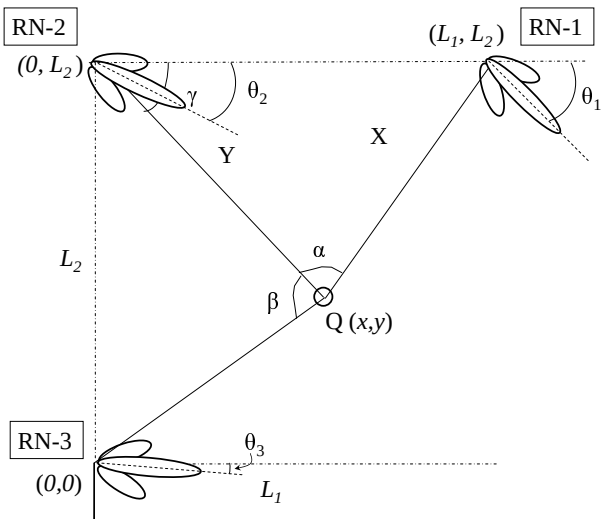


Figure 5.1: Arrangement of the reference nodes and the coordinate system.

constant angular separation between the directional beams of the RNs. ϕ_1 and ϕ_2 can be expressed in terms of the initial angular positions of each of the RN antenna beams ($\theta_i, i = 1, 2, 3$ shown in Fig. 5.1). Thus, $\phi_1 = \theta_1 - \theta_2$ and $\phi_2 = \theta_2 - \theta_3$. From (5.1) it is clear that absolute timings are not required as we are dealing with time differences in calculating the bearings. The coordinates (x, y) can be obtained by making use of the geometry of the RNs. Using simple trigonometry, it can be shown that the coordinates (x, y) of the node Q can be computed as

$$x = \frac{L_2 \cos\gamma}{\sin\beta} \cos(\beta - \gamma)$$
$$y = \frac{L_2 \cos\gamma}{\sin\beta} \sin(\beta - \gamma) \tag{5.2}$$

where

$$\gamma = \tan^{-1}\left(\frac{L_2 \cot\beta - L_1}{L_1 \cot\alpha - L_2}\right)$$

The symmetric arrangement of the RNs in Fig. 5.1 leads to a simple relation in (5.2). However, the localization principle used to arrive at (5.2) is valid for any arrangement of the RNs.

5.1.2 Signal Model

Each of the RNs (RN-1 to RN-3) is assumed to transmit a continuous time signal with a complex low-pass equivalent represented as $p(t)$. Then the low-pass equivalent of the received signal at the

UNs is given by

$$r(t) = z(\varphi)p(t) + \nu(t) \tag{5.3}$$

where $\nu(t)$ is the additive noise and $z(\varphi)$ is the amplitude profile of the received signal which is a function of the angular position $\varphi = \omega t$ of the directional antenna. The amplitude profile $z(\varphi)$ includes the effect of path loss and the antenna beampattern. In the presence of reflectors, $z(\varphi)$ can be modeled as

$$z(\varphi) = \sum_{m=1}^{M} a_m s(\varphi - \psi_m) \tag{5.4}$$

where $s(\varphi)$ is the expected amplitude profile based on the actual beampattern of the directional antenna. In (5.4), we assume that M signal components (the LOS and multipath components) are received at the UN with amplitudes $\{a_m\}_{m=1}^{M}$ and angular shifts $\{\psi_m\}_{m=1}^{M}$ determined by the angle of departure of the signal components from the directional antenna.

In discrete time, the received signal samples can be expressed as

$$r(n) = z(n)p(n) + \nu(n) \tag{5.5}$$

where $r(n) := r(t)|_{t=nT_s}$, $z(n) := z(\varphi)|_{\varphi=n\omega T_s}$ and T_s is the sampling time. Thus, the amplitude profile of the received signal can be written as

$$z(n) = \sum_{m=1}^{M} a_m s(n - \tau_m) \tag{5.6}$$

where $s(n) := s(\varphi)|_{\varphi=n\omega T_s}$ and $\tau_m := \psi_m/\omega T_s$. The signal model in (5.6) highlights the differences between the DB based positioning algorithm and the conventional time-of-arrival (TOA) schemes. One important difference is in the interpretation of $s(n)$. In TOA schemes, $s(n)$ refers to the transmitted pulse in the time domain while in the DB based positioning algorithm $s(n)$ corresponds to the beampattern of the directional antenna. Thus, the antenna beam-width in the DB algorithm is analogous to the pulse-width in TOA schemes. Similarly, where the accuracy of TOA schemes is dependent on the pulse-width (or signal bandwidth in the frequency domain) and the propagation speed, the positioning accuracy of the DB algorithm depends on the beampattern and rotational speed ω of the antenna. A quantitative discussion of these issues in presented in Section 5.5.4.

5.1.3 Directional Beacon based Positioning Algorithm

The directional beacon based positioning algorithm requires detection of the LOS component to mark the time instant when the transmitting beam is aligned with the receiver. Different

approaches can be used to detect the LOS component. For instance, the scheme in [45] searches for the maximum of the received signal strength in order to mark the times. Due to the very nature of indoor multipath propagation environment, the received signal consists of signal copies due to reflection and scattering in addition to the direct LOS component. We rely on the earliest signal component (instead of the strongest component) to detect the LOS component. This works well for both clear LOS and obstructed LOS scenarios.

For the system shown in Fig. 5.1, each RN broadcasts its known position and the initial angular position of its antenna beam. Using this information the unlocalized node Q executes the following steps of the positioning algorithm to localize itself:

Step 1. RN-1 transmits a signal continuously while its antenna is being rotated at a constant angular speed (ω).

Step 2. The received signal $r(t)$ at the UN consists of multiple copies of the transmit signal due to multipath propagation. The discrete-time samples $r(n)$ of the received signal are stored for further analysis.

Step 3. The amplitude profile $z(n)$ of the received signal at the UN is estimated (refer Section 5.3).

Step 4. The time shift t_1 associated with the LOS signal component in $z(n)$, due to RN-1, is estimated (refer Section 5.4).

Step 5. Steps 1-4 are repeated for RN-2 and RN-3 to obtain t_2 and t_3, respectively. Finally, the coordinates of the UN are obtained using (5.1) and (5.2).

5.2 Prototype Implementation

In this section, we describe the hardware and software implementation of the directional beacon based positioning algorithm. Our implementation is based on a software defined radio (SDR) platform. We use the GNU radio software and its hardware companion, the Universal Software Radio Peripheral (USRP). The SDR set-up allows us the ease and flexibility to design and transmit user-defined waveforms. In addition, the RF transceiver circuitry is fairly inexpensive. The rotating beacons are generated using a directional antenna coupled to a stepper motor. Further details are

described in the following subsections. Figs. 5.2 and 5.3 illustrate the block level set-up at the transmitter and receiver, respectively.

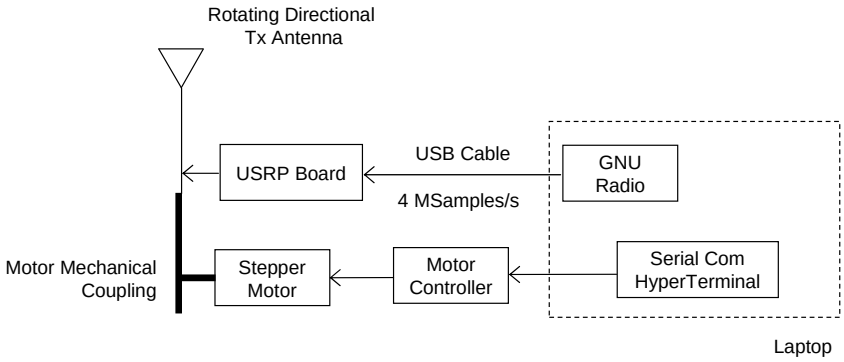


Figure 5.2: Hardware setup for the transmitter (or RN) with rotating directional antenna.

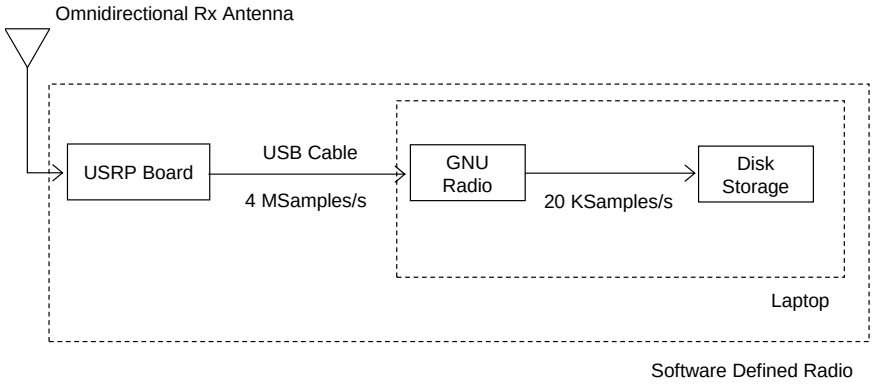


Figure 5.3: Hardware setup for the receiver (or UN).

5.2.1 Hardware Platform

The RF front end for the software radio comprises the USRP boards from Ettus Research [68]. The USRP consists of a daughter board (RFX2400), capable of transmitting and receiving RF signals in the 2.4-2.5 GHz band. The USRP also contains a motherboard that includes the universal serial bus (USB) interface, a field-programmable gate array (FPGA) to implement high-speed baseband processing and analog-to-digital converters (ADCs) and digital-to-analog converters (DACs). There are four on-board ADCs each with a 12-bit resolution sampling at 64MSamples/s and four

14-bit DACs each operating at 128MSamples/s.

The rotating beacons are generated by mounting a directional antenna on the stepper motor. The directional antenna TLX2415, is a 16 element linear antenna array from Telex with a main lobe half-power beam-width of 30° [69]. A bipolar stepper motor 8718L-02S, from Lin Engineering, is used to rotate the antenna with a constant angular speed [70]. The stepper motor is chosen to provide a torque of 900 oz-in (6.355 N.m) sufficient to rotate the antenna. For the experiments, a narrowband signal with a bandwidth of 1 KHz is used as the transmit signal and the motor is set to rotate the antenna at 0.47 rpm ($\omega = 0.049$ rad/s).

5.2.2 Software Platform

The software component of the SDR set-up is called the GNU radio which is implemented as an open public license software [71]. GNU radio provides a library of blocks for radio transmission and reception. These blocks are glued together using the Python scripting language. We use gr_block and usrp_sink_c to continuously transmit a single frequency tone. The receiver is implemented using the usrp_source_c block that captures the data from the RF front end and writes it to a file.

5.2.3 Experimental Setup

The experiments were carried out in an indoor fieldhouse at the University of Minnesota Recreation Center. The test area was a rectangular field measuring 55.14 m by 43 m. Snapshots of the experimental set-up are shown in Figs. 5.4 and 5.5. The RNs were placed at three corners of the rectangular field and the UN whose position is to be determined was placed inside the field.

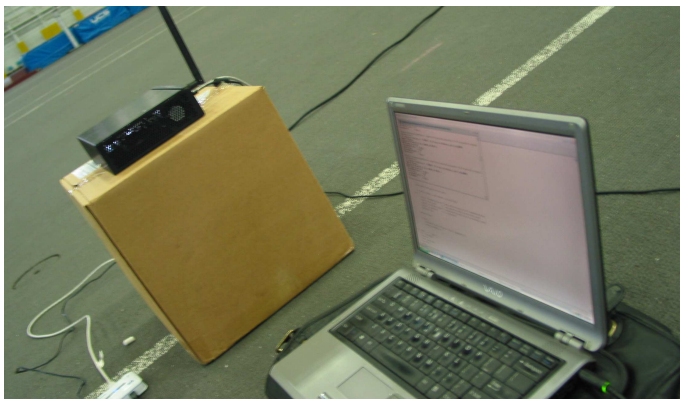


Figure 5.4: Experimental Setup: The unlocalized node whose position is to be determined.

Figure 5.5: Experimental Setup: The reference node equipped with directional antenna.

5.2.4 Data Collection and Processing

Steps 1-4 (from Section 5.1.3) are repeated for each of the RNs. The received signal at the UN is downconverted to a complex baseband signal. The I and Q components of the complex signal are passed through their respective channels. Each channel path has an ADC sampling at 64 MSamples/s. To reduce the burden on the USB interface, the ADC output is downsampled by 32 resulting in a 2 MSamples/s data stream. The digital samples from the I and Q channels are packed together and passed to the GNU radio software through the USB interface. This results in an effective data rate of 4 MSamples/s across the USB. Since each data sample is represented using 4 Bytes, the data rate across the USB is 16 MB/s. The samples of the received signal are finally written to a file. To keep file sizes manageable, we store only the first 32 samples out of every 6400 samples to the hard disk.

After the data has been pre-processed, as explained above, the collected data undergoes a number of post-processing steps as illustrated in Fig. 5.6. The details of some of the statistical signal processing algorithms are discussed in the next section. The collected data is first sliced to separate the data from each RN and from multiple antenna rotations at each RN. Next the data undergoes serial-to-parallel conversion into blocks of size N_b samples. Each data block is then subjected to a signal detection test, the results of which are used for further grouping of the data blocks. This aids in the estimation of the received signal amplitude profile. The estimated amplitude profile is used in a least squares (LS) framework for multipath time-delay estimation. Finally, the time delay estimates from all RNs are combined to obtain the bearing and position

coordinates of the unlocalized node.

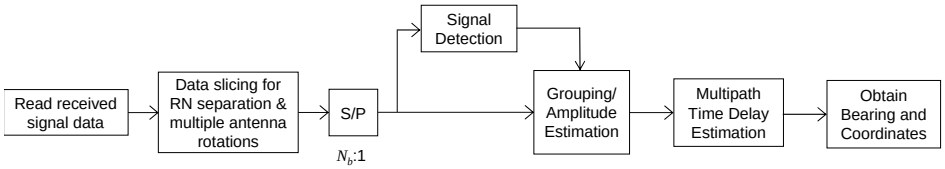


Figure 5.6: Post processing of the experimental data using statistical signal processing techniques.

5.3 Signal Detection and Enhanced Amplitude Estimation

5.3.1 Signal Detection

Using the GNU radio software, we generate a baseband discrete time signal of the form $p(n) = \sin(2\pi f_o\, n + \theta)$ where $f_o \in [0, 1]$ is the normalized digital frequency and θ is the phase. The discrete-time samples of the transmit signal are passed to the USRP which produces an analog signal of 1 KHz bandwidth. The RFX2400 daughter board on the USRP modulates the analog signal to the 2.4-2.5 GHz RF band.

Due to practical constraints, the data transfer over the USB interface to the USRP is intermittent resulting in intermittent signal transmission. We model this scenario, at the receiver, as two hypotheses: H_1 represents the case where the data is successfully transferred to the USRP resulting in a successful transmission (and reception) and H_0 represents the case where we assume that the data transfer to the USRP failed resulting in only noise being received. Thus, the received signal during the l^{th} data burst under the two hypotheses can be written as

$$\begin{aligned} H_1 &: r_l(n) = z\sin(2\pi f_o\, n + \theta_l) + \nu_l(n) \\ H_0 &: r_l(n) = \nu_l(n) \qquad\qquad \text{for} \quad n = 0, \cdots, N_b - 1 \end{aligned} \tag{5.7}$$

where z represents the amplitude of the received tone that is assumed to be constant over a burst of N_b samples. This assumption is invariably true due to the high sampling rates of the order of 10^6 samples/s. The amplitude z of the received signal includes the effect of path loss and the antenna beampattern. As a first step we would like to detect whether a given burst of N_b samples belongs to H_1 or H_0. The ML estimation of the amplitude z for a given data burst containing N_b

samples is given by [72, pg. 195]

$$\hat{z} = \frac{2}{N_b} \left| \sum_{n=0}^{N_b-1} r_l(n) e^{-j2\pi f_o n} \right|. \tag{5.8}$$

We then set a threshold Z_T to decide between the two hypotheses such that $\hat{z} \underset{H_1}{\overset{H_0}{\lessgtr}} Z_T$. The threshold Z_T is empirically set to 5-10% of the maximum value of the received signal. The received signal samples (or data bursts) which were classified under H_1 are used for further processing, as outlined in the next subsection.

5.3.2 Enhanced Amplitude Estimation

In order to improve the quality of the ML estimate obtained from (5.8) it is necessary to increase the number of samples used for the estimation. The ML estimate approaches the minimum variance unbiased estimate as $N_b \to \infty$. Since the number of samples in a single burst is fixed at N_b[1], the quality of the estimate can be enhanced by estimating the amplitude over multiple consecutive data bursts. Let us assume we have L consecutive signal bursts (bursts satisfying hypothesis H_1)[2]. In [19] this is formulated as the ML amplitude estimation of a constant frequency sinusoid with unknown initial phase.

The vector of received signal samples during the l^{th} burst can be expressed as

$$\mathbf{r}_l = z\mathbf{H}_l\,\boldsymbol{\theta}_l + \boldsymbol{\nu}_l \quad \text{for} \quad l = 1, \cdots, L \tag{5.9}$$

where $\mathbf{r}_l := [r_l(0) \ \cdots \ r_l(N_b-1)]^T$, $\boldsymbol{\theta}_l := [\cos\theta_l \quad \sin\theta_l]^T$ and

$$\mathbf{H}_l := \begin{bmatrix} 0 & 1 \\ \sin 2\pi f_o & \cos 2\pi f_o \\ \vdots & \vdots \\ \sin(N_b-1)2\pi f_o & \cos(N_b-1)2\pi f_o \end{bmatrix}.$$

Concatenating the L consecutive bursts

$$\mathbf{r} = \mathbf{H}\,\boldsymbol{\theta} + \boldsymbol{\nu}. \tag{5.10}$$

where $\mathbf{r} := \begin{bmatrix}\mathbf{r}_1^T \ \cdots \ \mathbf{r}_L^T\end{bmatrix}^T$, $\boldsymbol{\theta} := z[\boldsymbol{\theta}_1^T \ \cdots \ \boldsymbol{\theta}_L^T]^T$ and $\mathbf{H} := \text{diag}(\mathbf{H}_1 \ \cdots \ \mathbf{H}_L)$. Also note that, $\boldsymbol{\theta}^T\boldsymbol{\theta} = Lz^2$.

[1] In our set-up $N_b = 32$ since only 32 consecutive samples out of 6400 are stored to the disk.
[2] L has a maximum limit such that assuming the same amplitude for these bursts is reasonable.

Using the result from [19, Chapter 6], the enhanced ML estimate of the amplitude $\hat{z}$ using L consecutive bursts is given by

$$\hat{z} = \sqrt{\frac{\hat{\boldsymbol{\theta}}^T \hat{\boldsymbol{\theta}}}{L}} \tag{5.11}$$

where $\hat{\boldsymbol{\theta}} = \left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\,\mathbf{r}$.

5.4 Estimating the LOS Component in Multipath Environment

This section describes the procedure for estimating the LOS component in a multipath environment using the least squares error criterion. Let $\hat{z}(n)$ represent the estimated amplitude of the n^{th} received signal burst obtained using the procedure outlined in Section 5.3. Sample values of $\hat{z}$ corresponding to the parts of the received signal which were classified as H_0 are obtained through linear interpolation of the neighboring samples. Using the multipath model for $z(n)$ presented in (5.6), we can write

$$\hat{z}(n) = \sum_{m=1}^{M} a_m s(n-\tau_m) + \eta(n) \tag{5.12}$$

where $\eta(n)$ represents the error between the estimate $\hat{z}(n)$ and the true profile $z(n)$ given by (5.6). It is important to note that the samples of $\hat{z}(n)$ in (5.12) are spaced at an interval larger than T_s. Let us assume we have N samples[3] of $\hat{z}(n)$ in a single rotation of the antenna that are used to estimate $\mathbf{a} = [a_1 \cdots a_M]^T$ and $\boldsymbol{\tau} = [\tau_1 \cdots \tau_M]^T$. We can write (5.12) in matrix-vector form as

$$\hat{\mathbf{z}} = \mathbf{S}(\boldsymbol{\tau})\mathbf{a} + \boldsymbol{\eta} \tag{5.13}$$

where $\hat{\mathbf{z}} = [\hat{z}(0) \cdots \hat{z}(N-1)]$ and $\mathbf{S}(\boldsymbol{\tau})$ is an $N \times M$ matrix with $[\mathbf{S}(\boldsymbol{\tau})]_{n,m} = s(n-\tau_m)$. The least squares (LS) error criterion for the model in (5.13) becomes

$$J(\boldsymbol{\tau}, \mathbf{a}; n) = ||\hat{\mathbf{z}} - \mathbf{S}(\boldsymbol{\tau})\mathbf{a}||^2 \tag{5.14}$$

where $||.||$ represents the l_2 norm. The minimization of (5.14) poses two problems [73, 74]: 1) the cost function in (5.14) is non-linear in $\boldsymbol{\tau}$, and 2) the estimates $\{\hat{\tau}_m\}_{m=1}^{M}$ can only take values that are integer multiples of the sampling time period and hence limit the resolution of the estimate.

[3] In our set-up, $N = 10000$, the number of samples of $\hat{z}(n)$ in one rotation of the antenna.

To overcome these problems we consider the equivalent frequency domain model of (5.12) which can be expressed as

$$\tilde{z}(k) = \tilde{s}(k) \sum_{m=1}^{M} a_m e^{-j2\pi k\tau_m/N} + \tilde{\eta}(k) \quad \text{for} \quad k = 0, \cdots, N-1 \tag{5.15}$$

where $\tilde{z}(k) = \sum_{n=0}^{N-1} \hat{z}(n)e^{-j2\pi nk}$ represents the N-point discrete Fourier transform (DFT) of $\{\hat{z}(n)\}_{n=0}^{N-1}$. Similarly, $\tilde{s}$ is the DFT of $\{s(n)\}_{n=0}^{N-1}$. Re-writing (5.15) into an equivalent matrix-vector form we obtain

$$\tilde{\mathbf{z}} = \tilde{\mathbf{G}}(\boldsymbol{\tau})\mathbf{a} + \tilde{\boldsymbol{\eta}} \tag{5.16}$$

where $\tilde{\mathbf{z}} := [\tilde{z}(0) \cdots \tilde{z}(N-1)]^T$ and $\tilde{\mathbf{G}}(\boldsymbol{\tau}) := \tilde{\mathbf{S}}_D\mathbf{H}(\boldsymbol{\tau})$ with $\mathbf{H}(\boldsymbol{\tau})$ being an $N \times M$ matrix of the form

$$\mathbf{H}(\boldsymbol{\tau}) := \begin{bmatrix} 1 & \cdots & 1 \\ e^{-j2\pi\tau_1/N} & \cdots & e^{-j2\pi\tau_M/N} \\ \vdots & \ddots & \vdots \\ e^{-j2\pi(N-1)\tau_1/N} & \cdots & e^{-j2\pi(N-1)\tau_M/N} \end{bmatrix}$$

and $\tilde{\mathbf{S}}_D := \text{diag}[\tilde{s}(0) \cdots \tilde{s}(N-1)]$. Now, the least squares error criterion for the frequency domain model in (5.16) can be formulated as

$$J(\boldsymbol{\tau}, \mathbf{a}; k) = ||\tilde{\mathbf{z}} - \tilde{\mathbf{G}}(\boldsymbol{\tau})\mathbf{a}||^2. \tag{5.17}$$

Since (5.17) is linear in $\mathbf{a}$ for a given $\boldsymbol{\tau}$, the least squares estimate for $\mathbf{a}$ is given by

$$\hat{\mathbf{a}} = \left[\tilde{\mathbf{G}}^{\mathcal{H}}(\boldsymbol{\tau})\tilde{\mathbf{G}}(\boldsymbol{\tau})\right]^{-1} \tilde{\mathbf{G}}^{\mathcal{H}}(\boldsymbol{\tau})\, \tilde{\mathbf{z}}. \tag{5.18}$$

Substituting $\hat{\mathbf{a}}$ from (5.18) into (5.17) reduces the LS error criterion to

$$J(\boldsymbol{\tau}; k) = ||\tilde{\mathbf{G}}^{\perp}(\boldsymbol{\tau})\, \tilde{\mathbf{z}}||^2 \tag{5.19}$$

where $\tilde{\mathbf{G}}^{\perp}(\boldsymbol{\tau}) = \mathbf{I}_N - \tilde{\mathbf{G}}(\boldsymbol{\tau}) \left[\tilde{\mathbf{G}}^{\mathcal{H}}(\boldsymbol{\tau})\tilde{\mathbf{G}}(\boldsymbol{\tau})\right]^{-1} \tilde{\mathbf{G}}^{\mathcal{H}}(\boldsymbol{\tau})$.

It has been shown that the error function in (5.19) is highly oscillatory or multimodal [73] with closely spaced multiple local minima which makes the minimization extremely difficult. A solution, suggested in [73, 74], is to allow the amplitude estimate $\hat{\mathbf{a}}$ to be complex which results in a smoother error function. The conjugate symmetry associated with $\tilde{\mathbf{z}}$ and $\tilde{\mathbf{S}}_D$ forces the amplitude estimates in (5.18) to be real. Thus, to permit $\hat{\mathbf{a}}$ to be complex, we need to consider the single

sided spectrum in the least squares error function (5.17). Hence we define the following using single-sided spectrums:

$$\tilde{\mathbf{z}}_p := [\tilde{z}(0) \cdots \tilde{z}(\frac{N}{2}-1)]^T,$$
$$\tilde{\mathbf{S}}_{Dp} := \text{diag}[\tilde{s}(0) \cdots \tilde{s}(\frac{N}{2}-1)] \qquad \text{and}$$
$$\mathbf{H}_p(\boldsymbol{\tau}) := \begin{bmatrix} 1 & \cdots & 1 \\ e^{-j2\pi\tau_1/N} & \cdots & e^{-j2\pi\tau_M/N} \\ \vdots & \ddots & \vdots \\ e^{-j2\pi(\frac{N}{2}-1)\tau_1/N} & \cdots & e^{-j2\pi(\frac{N}{2}-1)\tau_M/N} \end{bmatrix}.$$

We can now re-write the LS error function in (5.17) as

$$J_C(\boldsymbol{\tau}, \mathbf{a}_C; k) = ||\tilde{\mathbf{z}}_p - \tilde{\mathbf{G}}_p(\boldsymbol{\tau})\mathbf{a}_C||^2, \tag{5.20}$$

with $\tilde{\mathbf{G}}_p(\boldsymbol{\tau}) = \tilde{\mathbf{S}}_{Dp}\mathbf{H}_p(\boldsymbol{\tau})$ and the LS estimate of the complex amplitudes given by

$$\hat{\mathbf{a}}_C = \left[\tilde{\mathbf{G}}_p^{\mathcal{H}}(\boldsymbol{\tau})\tilde{\mathbf{G}}_p(\boldsymbol{\tau})\right]^{-1} \tilde{\mathbf{G}}_p^{\mathcal{H}}(\boldsymbol{\tau})\, \tilde{\mathbf{z}}_p. \tag{5.21}$$

Substituting $\hat{\mathbf{a}}_C$ into (5.20) leads to $J_C(\boldsymbol{\tau}; k)$ that is known to be a smoother function with far fewer oscillations [73]. However, the minimum of $J_C(\boldsymbol{\tau}; k)$ is not the true minimum of $J(\boldsymbol{\tau}; k)$. Adding a penalty term proportional to the imaginary part of $\mathbf{a}_C$ to $J_C(\boldsymbol{\tau}; k)$ allows us to control its smoothness [73]. The modified error function can be written as

$$J_C^{'}(\boldsymbol{\tau}; k) = ||\tilde{\mathbf{z}}_p^{'} - \tilde{\mathbf{G}}_p^{'}(\boldsymbol{\tau})\hat{\mathbf{a}}_C^{'}||^2 \tag{5.22}$$

where $\tilde{\mathbf{z}}_p^{'} := \begin{bmatrix} \text{Re}\{\tilde{\mathbf{z}}_p\} \\ \text{Im}\{\tilde{\mathbf{z}}_p\} \\ \mathbf{0} \end{bmatrix}$, $\tilde{\mathbf{G}}_p^{'}(\boldsymbol{\tau}) := \begin{bmatrix} \text{Re}\{\tilde{\mathbf{G}}_p(\boldsymbol{\tau})\} & -\text{Im}\{\tilde{\mathbf{G}}_p(\boldsymbol{\tau})\} \\ \text{Im}\{\tilde{\mathbf{G}}_p(\boldsymbol{\tau})\} & \text{Re}\{\tilde{\mathbf{G}}_p(\boldsymbol{\tau})\} \\ \mathbf{0} & \alpha_p\mathbf{I}_M \end{bmatrix}$ and $\hat{\mathbf{a}}_C^{'} := \begin{bmatrix} \text{Re}\{\hat{\mathbf{a}}_C\} \\ \text{Im}\{\hat{\mathbf{a}}_C\} \end{bmatrix}$.

The penalty term is controlled by α_p that allows us to trade-off between the smoothness of $J_C(\boldsymbol{\tau}; k)$ and its bias from the global minimum of $J(\boldsymbol{\tau}; k)$. We solve for $\hat{\boldsymbol{\tau}}$ by minimizing the error function in (5.22) after substituting for $\hat{\mathbf{a}}_C$ from (5.21).

5.4.1 LOS Estimation using Data from Multiple Repeated Antenna Rotations

Our experimental setup allows us to collect received signal samples at the UN for multiple repeated rotations of the transmit antenna at the RNs. Assuming the environment has not changed significantly between rotations, the data collected during each rotation will result in similar amplitude

profiles. If we collect N samples in each of the R rotations then the received signal vector of NR samples can be modeled as

$$\hat{\mathbf{z}}^{(R)} = \mathbf{S}^{(R)}(\boldsymbol{\tau})\mathbf{a} + \boldsymbol{\eta} \tag{5.23}$$

where $\hat{\mathbf{z}}^{(R)} = [\hat{z}(0) \cdots \hat{z}(NR-1)]^T$ and $\mathbf{S}^{(R)}(\boldsymbol{\tau})$ is an $NR \times M$ matrix with $\left[\mathbf{S}^{(R)}(\boldsymbol{\tau})\right]_{n,m} = s(\langle n \rangle_N - \tau_m)$, where $\langle \cdot \rangle_N$ represents modulo-by-N operation. For M multipaths, we have $\boldsymbol{\tau} = [\tau_1 \cdots \tau_M]^T$ and $\mathbf{a} = [a_1 \cdots a_M]^T$. Similar to (5.15), the frequency domain model for the data from R rotations of the antenna can be expressed as

$$\tilde{z}^{(R)}(k) = \tilde{s}^{(R)}(k) \sum_{m=1}^{M} a_m e^{-j2\pi k\tau_m/NR} + \tilde{\eta}(k) \quad \text{for} \quad k = 0, \cdots, NR-1 \tag{5.24}$$

where $\tilde{z}^{(R)}(k)$ and $\tilde{s}^{(R)}(k)$ represents the NR-point DFT of $\hat{\mathbf{z}}^{(R)}$ and $\{s(\langle p \rangle_N)\}_{p=0}^{NR-1}$, respectively. Transforming (5.24) into matrix-vector form, we obtain

$$\tilde{\mathbf{z}}^{(R)} = \tilde{\mathbf{G}}^{(R)}(\boldsymbol{\tau})\mathbf{a} + \tilde{\boldsymbol{\eta}} \tag{5.25}$$

where

$$\begin{aligned} \tilde{\mathbf{z}}^{(R)} &:= [\tilde{z}^{(R)}(0) \cdots \tilde{z}^{(R)}(NR-1)]^T, \\ \tilde{\mathbf{G}}^{(R)}(\boldsymbol{\tau}) &:= \tilde{\mathbf{S}}_D^{(R)} \mathbf{H}^{(R)}(\boldsymbol{\tau}), \\ \tilde{\mathbf{S}}_D^{(R)} &:= \mathrm{diag}[\tilde{s}^{(R)}(0) \cdots \tilde{s}^{(R)}(NR-1)] \end{aligned}$$

and $\mathbf{H}^{(R)}(\boldsymbol{\tau})$ an $NR \times M$ matrix with $\left[\mathbf{H}^{(R)}(\boldsymbol{\tau})\right]_{p,m} = e^{-j2\pi(p-1)\tau_m/NR}$. The LS estimates of $\mathbf{a}$ and $\boldsymbol{\tau}$ can be obtained by minimizing an LS error function of the form (5.22).

5.5 Experimental Results

5.5.1 Estimation of Amplitude Profile

The first step in the DB based positioning algorithm is to estimate the amplitude profile of the sinusoidal signal received at the unlocalized node Q. A typical plot of the received signal at node Q for one 90° rotation of the transmitting antenna at RN-2 is shown in Fig. 5.7. The ML estimate of the amplitude profile $\hat{z}(n)$, using (5.8) to estimate the amplitude of the sinusoid in a single burst of the received data, is shown in Fig. 5.8(a). To improve the estimated amplitude profile, we determine the instances of consecutive signal bursts and obtain an enhanced ML estimate of the

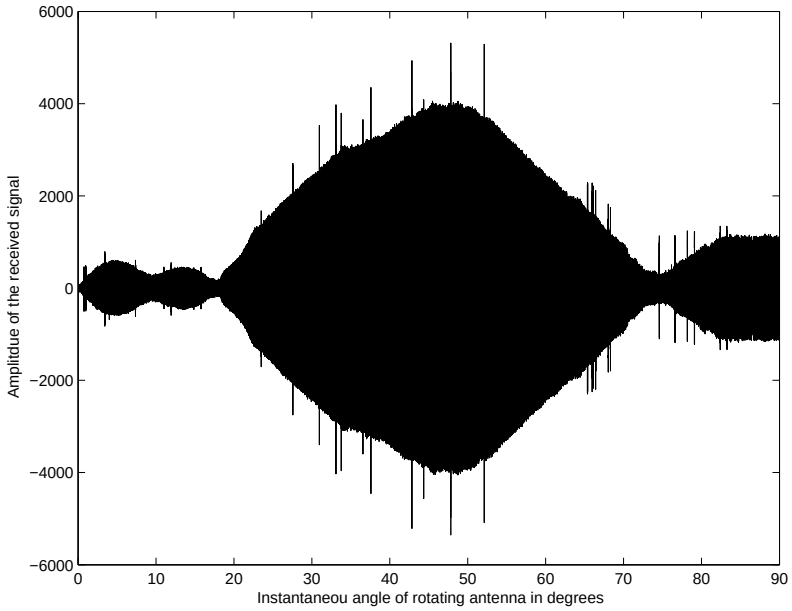


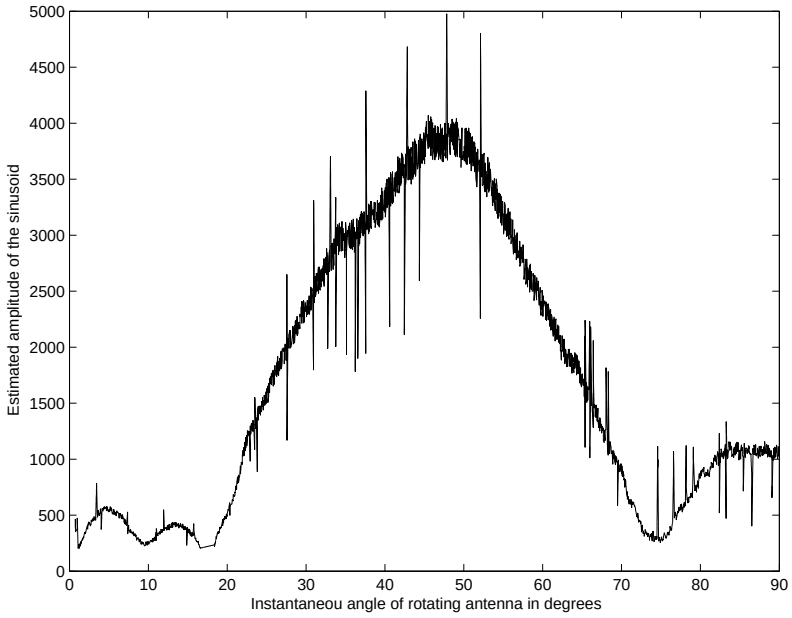
Figure 5.7: Received signal from 90° rotation of the RN-2.

sinusoid amplitude using (5.11). The enhanced amplitude profile is shown in Fig. 5.8(b). The use of consecutive bursts to estimate the amplitude reduces the effect of noise as seen by comparing Figs. 5.8(a) and 5.8(b).

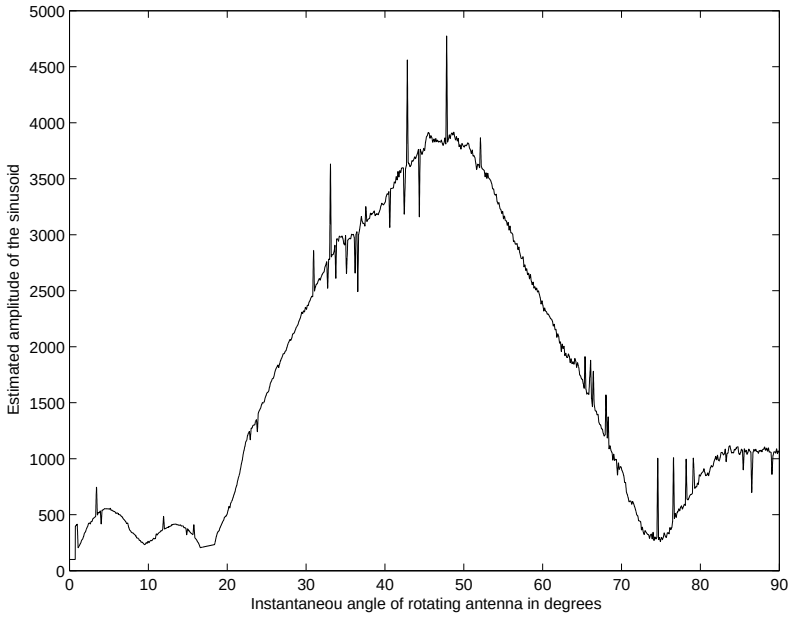
5.5.2 Estimation of the LOS Component

We use the algorithm described in Section 5.4 to estimate the LOS and multipath components in the received signal data. For simplicity, we use $M = 2$, i.e., we consider the LOS component and a single reflection or multipath component. In our experimental setup, the estimated amplitude profile $\hat{z}(n)$ consists of $N = 10000$ samples for 90° rotation of the directional antenna ($\varphi = 0$ to 90°). These samples are taken every 3.2 ms. In Fig. 5.9(a) a plot of the estimated amplitude profile $\hat{z}(n)$ for the received signal from RN-3 is shown. The antenna beampattern (or radiation pattern) for the azimuth angle ranging from 0 to 90°, which represents $s(n)$, is shown in Fig. 5.9(b).

To keep the required computations at a reasonable level, we downsample $\hat{z}(n)$ and $s(n)$ by 16 and use 1024-point DFT. For the penalty term in (5.22), we use $\alpha_p = 10^4$. MATLAB optimization routines (`fminunc`) are used to minimize (5.22) and obtain the time-delay estimate $\hat{\boldsymbol{\tau}} = [\hat{\tau}_1 \ \hat{\tau}_2]^T$. Substituting $\hat{\boldsymbol{\tau}}$ in (5.21) gives us the LS estimate of the complex amplitude $\hat{\mathbf{a}}_C = [\hat{a}_{C1} \ \hat{a}_{C2}]^T$ which comprises the LOS and multipath component amplitudes. For the received signal from RN-3, the

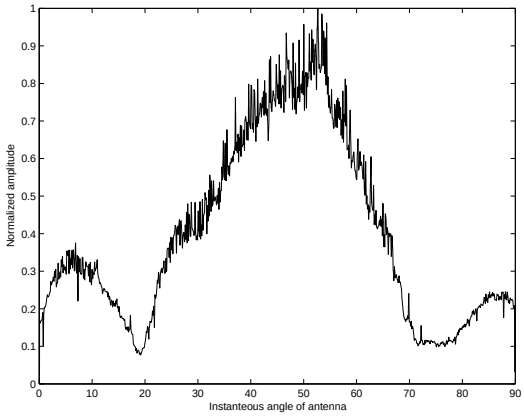


(a) ML amplitude estimation using single burst of $N_b = 32$ samples.

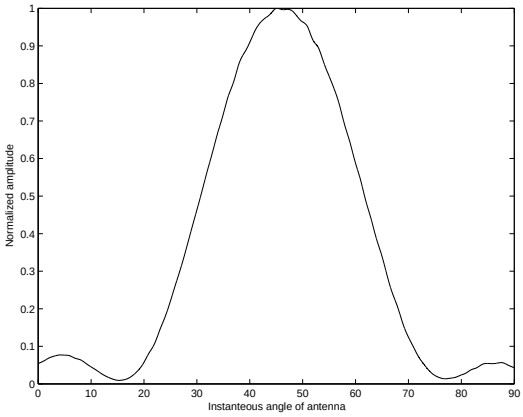


(b) ML amplitude estimation using multiple consecutive bursts.

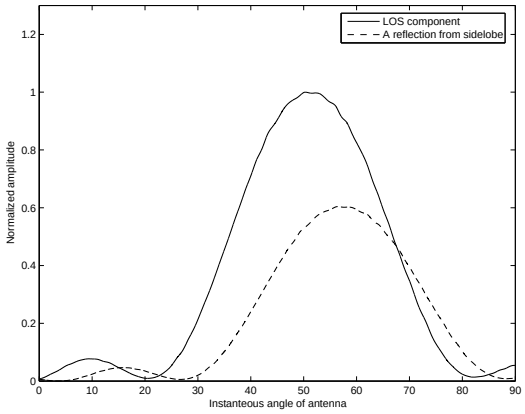
Figure 5.8: Estimated amplitude profile of the received signal from RN-2 (Fig. 5.7).



(a) Estimated amplitude profile of the received signal.



(b) Antenna Beampattern representing $s(n)$.



(c) Estimated LOS and multipath components.

Figure 5.9: Experimental results based on the received signal from RN-3.

estimated LOS and multipath components are shown in Fig. 5.9(c) with amplitudes $|\hat{a}_{C1}|$ and $|\hat{a}_{C2}|$, respectively. We estimate the earliest component (smaller of the τ_i's, $i = 1, 2$) as the LOS component which is true for both clear LOS and obstructed LOS scenarios. This procedure is repeated for the received signal from each RN to obtain the estimates for $t_i, i = 1, 2, 3$ in (5.1).

5.5.3 Position estimation and lower bound on estimation error variance

Based on the estimated LOS component for each RN, we obtain the corresponding time-delay estimates $t_i, i = 1, 2, 3$. Finally, we use (5.1) and (5.2) to estimate the position of the unlocalized node Q. For further improving the accuracy of the position estimate we use the technique outlined in section 5.4.1 for obtaining the time-delay estimates using multiple 90° rotations of the transmitting antenna at each RN. Since this increases the data record length used for estimation, we expect the signal-to-noise ratio (SNR) to increase and the estimation error variance to decrease as we add data from multiple antenna rotations. In Fig. 5.10, we plot the variance of the position estimation error as a function of the number of 90° antenna rotations. It is seen that with as few as eight 90° rotations of the transmit antenna, the estimation error variance goes down by 10 dB. Let $\mathcal{E}$ denote the position estimation error. We obtain a lower bound on the variance of the position estimation error, the details of which are given in Appendix B, as:

$$\mathrm{var}(\mathcal{E}) \geq \left(K_x^2 + K_y^2\right) \cdot \frac{1}{\frac{\xi}{N_o/2}\bar{F}^2} \tag{5.26}$$

It should be noted that this bound is not as tight as the Cramer-Rao lower bound (CRLB) and is rarely attainable, but is mathematically more tractable.

Using a single rotation of the antenna, the root mean square value of the position estimation error is 1.457 m. After using data from eight 90° rotations of the antenna the root mean square position error goes down to 0.362 m.

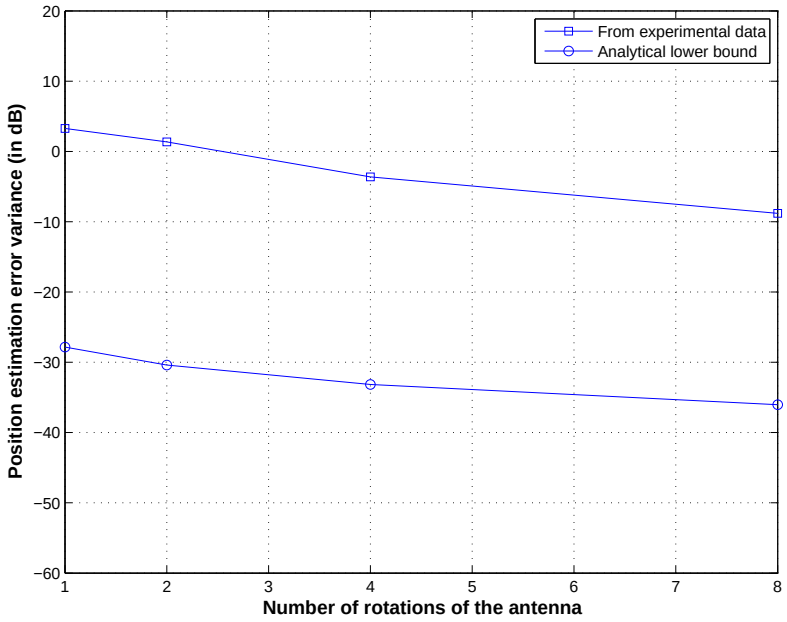


Figure 5.10: Position estimation error variance as a function of the number of rotations of the antenna.

5.5.4 Effect of Time Delay Estimation, Synchronization and Motor Speed Step Errors on Position Estimation

Following the notation used in Appendix B, we can express the estimation error in the x and y coordinates as:

$$\Delta\hat{x} = \frac{\partial g(t)}{\partial \gamma} \cdot \frac{\partial \gamma}{\partial \alpha} \cdot \omega \cdot \Delta t = K_x \cdot \Delta t$$
$$\Delta\hat{y} = \frac{\partial h(t)}{\partial \gamma} \cdot \frac{\partial \gamma}{\partial \alpha} \cdot \omega \cdot \Delta t = K_y \cdot \Delta t$$

where Δt is the time delay estimation error. Thus the position estimation error is:

$$\mathcal{E} = \sqrt{(\Delta\hat{x})^2 + (\Delta\hat{y})^2}. \tag{5.27}$$

On the other hand, for a TOA based technique, the position estimation error corresponding to a time delay estimation error of Δt is given by:

$$\mathcal{E}_{TOA} \approx \sqrt{(c\Delta t)^2 + (c\Delta t)^2} \tag{5.28}$$

where $c = 3 \times 10^8$ m/s is the speed of electromagnetic waves in free space.

Comparing (5.27) and (5.28) it is seen that the position error for directionality based techniques is proportional to the angular speed of the rotational antenna (ω) while for TOA techniques it is proportional to the signal propagation speed (c) which is many orders of magnitude larger. This comparison is also illustrated in Fig. 5.11. The directional beacon based method results in a positioning error of less than 0.5 m with up to 500 milliseconds error in time delay estimation for $\omega = 0.049$ rad/s. Fig. 5.11 also shows that the time needed for each position estimation using the directional beacon based method can be reduced significantly by increasing ω without significant loss in positioning accuracy. The positioning error remains nearly 8 orders of magnitude smaller than TOA schemes for similar time-delay estimation errors.

Similar analysis can be applied to the synchronization error between the reference nodes. The directional beacon based technique can tolerate synchronization errors of the order of milliseconds, which has been shown to be achievable in real applications [56], without significant positioning errors. Another source of error in the directional beacon based algorithm would be due to changes in the angular speed of the rotational antenna (ω). The stepper motor used in our experiments is rated for a step error of ± 1.08 arc minutes (or 0.018°). It can be shown that a 500 milliseconds error in time delay estimation is equivalent to a 1.4° step error over the duration of one position estimation and the rated step error is close to two orders of magnitude lower. This shows that the step error can be safely ignored as a source of error in our experimental set-up.

5.6 Conclusion

The prototype implementation for a directional beacon based positioning algorithm using RF signals was presented. Novel techniques for improving the position estimation accuracy using maximum likelihood amplitude estimation, least squares based time-delay estimation and combining data from multiple antenna rotations, were presented. It was also shown that the accuracy of the directional beacon based algorithm depends on the width of the antenna beampattern and the rotational speed of the directional antenna. We demonstrated, through experiments, the ability to obtain position estimation results with sub-meter accuracy. The robustness of the algorithm (and the implementation) to timing and synchronization errors was also demonstrated.

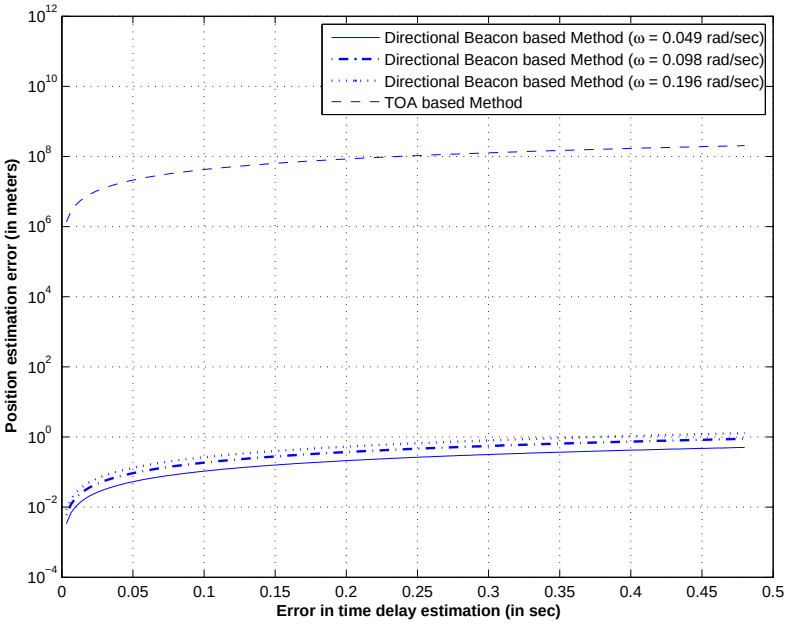


Figure 5.11: Comparison of the Directional Beacon and TOA based position estimation methods in terms of their sensitivity to time delay estimation error.

Chapter 6

Sensor Selection and Event Detection using Heuristic based Sequential Hypothesis Testing

In this chapter we address the sensor selection and event detection problems in applications where sensor networks are used for monitoring critical points or infrastructure and detecting abnormal or emergency conditions. In these applications, sensor nodes are used to automatically detect such conditions and enable a central node or the fusion center to take further action. Examples of such applications include detecting leaks in nuclear power plants, detecting heat sources in the context of forest fires etc. The goal in these network deployments is to detect these conditions (or events) with the minimum amount of delay. We propose a greedy heuristic based sensor selection and a sequential detection procedure that significantly improves the detection speed [75]. It will be shown that the greedy heuristic can be expressed in closed form for most probabilistic models.

In the proposed model, the fusion center based on some a priori knowledge of the sensors in its neighborhood, selects one sensor at a time while maximizing a greedy heuristic. Specifically, instead of collecting a fixed number of measurements, the fusion center collects measurements from one sensor at each time step, until by some sequential decision rule the collection stops and a decision is made. We compare the performance of the sequential detector with that of a non-sequential or fixed sample size (FSS) detector. The comparison is based on the number of measurements needed

to achieve a certain detection accuracy. The sequential procedure significantly outperforms the FSS detector in that it always needs fewer measurements on average to achieve the same detection performance. A simplified heuristic is also derived for the Gaussian probabilistic model. The simplified heuristic performs as good as or slightly better than the greedy heuristic. The greedy heuristic based sensor selection provides a general framework applicable for arbitrary probabilistic models and for models where a simplified heuristic is difficult to obtain. We also present an extension of the sensor selection and sequential detection procedure for M-ary hypothesis testing problems.

6.1 Sensor Selection and Event Detection: System Model

In this section we describe the underlying system model. A wireless sensor network typically consists of a *fusion center* – a central node with high computational power for data processing and decision making, few *local coordinator nodes* – nodes with relatively high computational power which manage sensor nodes in a certain geographical region, and a large number of *sensor nodes*. Fig. 6.1 illustrates the system architecture. The fusion center communicates with the local coordinators directly or through multiple hops involving other nodes. As a result of this communication the fusion center is aware of the overall network topology. We consider sensor network deployments aimed at event detection where sensor nodes are used to automatically detect critical events or conditions. It has been observed that typically the occurrence and non-occurrence of events can be modeled as two hypotheses which differ in the statistics (or distributions) of the sensor measurements. The local coordinators are tasked with querying the sensor nodes in their region regularly to obtain estimates of the measurement statistics. This can be carried out using a mechanism similar to continuous probabilistic query (CPQ) proposed in [54]. The local coordinators pass this information to the fusion center which uses it to query specific sensor nodes, following some sensor selection criterion, allowing it to detect events when they occur. This outlines the centralized framework where the decision making and event detection occurs at the fusion center. If the local coordinators manage relatively large geographical regions a distributed implementation might be more efficient where each local coordinator uses the above framework to detect events in its region and notifies the fusion center when an event is detected.

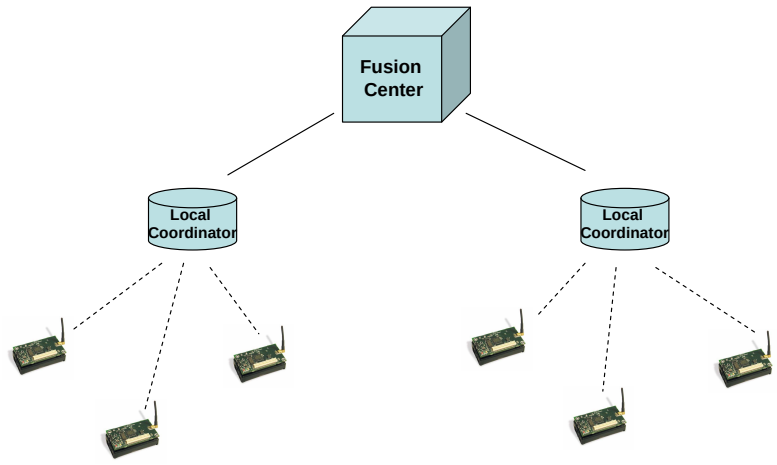


Figure 6.1: Typical Wireless Sensor Network System Model

6.2 Binary Sequential Hypothesis Testing

Consider a scenario where the goal is to detect the occurrence of an event using a network of N sensor nodes. We can formulate this as a binary hypothesis testing problem, where H_0 represents the null hypothesis or non-occurrence of the event and H_1 the alternative hypothesis or the case where the event occurred. Assuming we have some knowledge about the probability density functions (PDF) or distributions of the sensor measurements under each hypothesis, we can model the two hypotheses as:

$$\begin{aligned} H_0 &: y^k \sim p\left(y^k/H_0\right) \quad \text{and} \\ H_1 &: y^k \sim p\left(y^k/H_1\right), \quad \text{for} \quad k = 1, \ldots, N \end{aligned} \tag{6.1}$$

where y^k represents the k^{th} sensors' measurement and $p\left(y^k/H_m\right), m = 0, 1$ its conditional PDF. Let y_n^k denote the k^{th} sensors' measurement at the n^{th} sampling instant. Given a set of measurements $\{y_1^{k_1}, \cdots, y_n^{k_n}\}$, where the sequence $k_i (i = 1, \ldots, n)$ represents the indices of the sensors which were selected for measurement at each sampling instant, we can compute the log-likelihood ratio $\Lambda(n)$ as:

$$\Lambda(n) = \log \left\{ \frac{p(y_1^{k_1}, \cdots, y_n^{k_n}/H_1)}{p(y_1^{k_1}, \cdots, y_n^{k_n}/H_0)} \right\} \tag{6.2}$$

A sequential decision rule is formulated as follows:

1. Decide H_1 if $\Lambda(n) > b$
2. decide H_0 if $\Lambda(n) < a$, else

3. perform another measurement if $\Lambda(n) > a$ and $\Lambda(n) < b$.

where a and b are thresholds such that $-\infty < a \leq 0 \leq b < \infty$. This is an example of a sequential probability ratio test (SPRT). It has been shown that SPRT is the detector with the smallest average sample size [76]. Using Wald's approximation, the thresholds a and b are given as:

$$\begin{aligned} b &\simeq \log\left(\frac{P_D}{P_{FA}}\right) \\ a &\simeq \log\left(\frac{1-P_D}{1-P_{FA}}\right) \end{aligned} \tag{6.3}$$

where

$$\begin{aligned} P_D &= \text{Probability of detection} = \int_b^\infty p\left(\Lambda(n) > b/H_1\right) d\Lambda \\ P_{FA} &= \text{Probability of false alarm} = \int_b^\infty p\left(\Lambda(n) > b/H_0\right) d\Lambda. \end{aligned} \tag{6.4}$$

For a desired detection performance, in terms of P_D and P_{FA}, we can use (6.3) to determine the thresholds. Assuming that all the sensor measurements are independent the log-likelihood ratio can be written in a recursive form as [77]:

$$\begin{aligned} \Lambda(n) &= \log\left\{\left(\prod_{i=1}^{n-1}\frac{p(y_i^{k_i}/H_1)}{p(y_i^{k_i}/H_0)}\right)\cdot\frac{p(y_n^{k_n}/H_1)}{p(y_n^{k_n}/H_0)}\right\} \\ &= \Lambda(n-1) + \log\left\{\frac{p(y_n^{k_n}/H_1)}{p(y_n^{k_n}/H_0)}\right\} \end{aligned} \tag{6.5}$$

6.3 Greedy Heuristic based Sensor Selection

In event detection problems, in addition to meeting the performance criterion based on the detection and false alarm probabilities, it is equally important to detect the occurrence of the event as soon as possible. Assuming that each sensor observation takes the same amount of time, we would like to select the sensors to be sampled for measurement in a way that minimizes the total observation time. In other words, we would like to select as few sensors as possible while meeting the detection performance. Based on this we define the following *greedy* heuristic:

At each time instant, select a sensor for measurement which maximizes the probability of making a correct decision on the next time instant (P_C).

The probability of making a correct decision can be expressed as the sum of: (a) probability that $\Lambda(n) > b$ (i.e., decide on H_1) when H_1 is the true hypothesis, and (b) probability that $\Lambda(n) < a$

when H_0 is the true hypothesis. Thus we can express the greedy heuristic or the probability measure P_C as:

$$P_C(k,n) = P(\Lambda(n) > b/H_1) \cdot P^p(H_1) + P(\Lambda(n) < a/H_0) \cdot P^p(H_0) \tag{6.6}$$

where

$$\Lambda(n) = \Lambda(n-1) + \log\left\{\frac{p(y_n^k/H_1)}{p(y_n^k/H_0)}\right\}$$

and $P^p(H_m), m = 0, 1$ represents the a posteriori probability of each hypothesis which can be written as:

$$\begin{aligned} P^p(H_1) &= P\left(H_1/y_1^{k_1}, \cdots, y_{n-1}^{k_{n-1}}\right) \\ &= \frac{P\left(y_1^{k_1}, \cdots, y_{n-1}^{k_{n-1}}/H_1\right) \cdot P(H_1)}{P\left(y_1^{k_1}, \cdots, y_{n-1}^{k_{n-1}}\right)} \\ &= \frac{\left\{\prod_{i=1}^{n-1} p\left(y_i^{k_i}/H_1\right)\right\} \cdot P\left(H_1\right)}{P\left(y_1^{k_1}, \cdots, y_{n-1}^{k_{n-1}}/H_0\right) \cdot P(H_0) + P\left(y_1^{k_1}, \cdots, y_{n-1}^{k_{n-1}}/H_1\right) \cdot P(H_1)}. \end{aligned}$$

Thus,

$$P^p(H_m) = \frac{P_m(n-1) \cdot P(H_m)}{\sum_{m=0,1} P_m(n-1) \cdot P(H_m)}, \quad \text{for} \quad m = 0, 1 \tag{6.7}$$

where

$$P_m(n-1) = \prod_{i=1}^{n-1} p\left(y_i^{k_i}/H_m\right)$$

and $P(H_m)$ is the a priori probability of each hypothesis before any sensor measurements were obtained. At any time instant n we perform the following maximization to decide which sensor to select for the next measurement:

$$k_n = \underset{k}{\text{argmax}} \quad P_C(k,n) \quad \text{where} \quad k \in \{1, \cdots, N\}. \tag{6.8}$$

6.3.1 Greedy sensor selection and sequential detection

The overall combined strategy for greedy heuristic based sensor selection and sequential detection can be outlined as follows:

Step 1. Compute the thresholds a and b based on the desired detection performance (P_D and P_{FA}).

Step 2. Initialize $\Lambda(0) = 0$ and $i = 1$.

Step 3. Greedy Selection: Solve for k_i, the index of the sensor to be used for the next measurement, using (6.8) and (6.6).

Step 4. Obtain measurement from the k_i^{th} sensor and update the log-likelihood ratio to $\Lambda(i)$ using (6.5).

Step 5. Compare the log-likelihood ratio to the thresholds based on the sequential decision rule.

Step 6. Repeat steps $3-5$ above for successive measurements ($i = i+1$), until a decision is made in step 5.

6.4 Optimal Sensor Selection

In this section we consider the *optimal* sensor selection strategy. The optimal sensor selection strategy can be stated as follows:

Given the sample observation values $y_i^{k_i}$, where $i = 1, \ldots, n$ and $k_i \in \{1, \ldots, N\}$, from all sensors for all time steps, choose a sequence of sensors (or sensor observations) which will allow us to make a decision at the earliest (i.e., shortest sequence) while maximizing the probability of a correct decision.

The optimal sensor selection and optimal sequence length are determined as follows (refer to Fig. 6.2 for an illustration of the optimal selection strategy with two sensors):

Step 1. We assume that the observation values $y_i^{k_i}$, where $i = 1, \ldots, n$ and $k_i \in \{1, \ldots, N\}$, from all sensors for all time steps are known *a priori*.

Step 2. Start with all sequences of length $i = 1$.

Step 3. For each sequence of length i (there are $O(2^i)$ such sequences), compute the log-likelihood ratio and compare it with the thresholds a and b based on the sequential decision rule.

Step 4. If any sequence in step 3 results in a decision it represents the *optimal* sequence. If more than one sequence of length i results in a decision, the optimal sequence is chosen as the one which maximizes the posteriori probability.

Step 5. Repeat steps $3-4$ for sequences of length $i = i+1$, until a decision results in step 4.

In Section 6.9 we provide a representative comparison, based on Monte Carlo simulations, between the greedy heuristic based sensor selection and the optimal sensor selection strategies.

6.5 Analytical Evaluation under Gaussian Modeling

The general framework for sequential detection and greedy heuristic based sensor selection was presented in Sections 6.2 and 6.3. We now consider a specific model and provide detailed analysis for the sensor selection and sequential detection problems. We assume the sensor measurements to be modeled as being Gaussian distributed. This model is justified since measurement noise in sensors is frequently modeled by a zero-mean Gaussian distribution with finite variance. In addition, in parametric estimation theory estimation errors are commonly approximated using Gaussian distributions.

Consider the binary hypothesis testing problem with N sensors which are the available sources of measurements. The hypotheses can be expressed as:

$$\begin{aligned} H_0 &: y^k \sim \mathcal{N}(\mu_{0,k}, \sigma^2_{0,k}) \\ H_1 &: y^k \sim \mathcal{N}(\mu_{1,k}, \sigma^2_{1,k}) \quad \text{for} \quad k = 1, \cdots, N. \end{aligned} \tag{6.9}$$

Using this we obtain an expression for $P_C(k,n)$.

$$\begin{aligned} P(\Lambda(n) > b/H_1) &= P\left(\Lambda(n-1) + \log\left(\frac{\sigma_{0,k}}{\sigma_{1,k}}\right) + \frac{(y_n^k - \mu_{0,k})^2}{2\sigma^2_{0,k}} - \frac{(y_n^k - \mu_{1,k})^2}{2\sigma^2_{1,k}} > b/H_1\right) \\ &= P\left(\frac{(y_n^k - \mu_{0,k})^2}{2\sigma^2_{0,k}} - \frac{(y_n^k - \mu_{1,k})^2}{2\sigma^2_{1,k}} > b - C_{n-1}/H_1\right) \end{aligned} \tag{6.10}$$

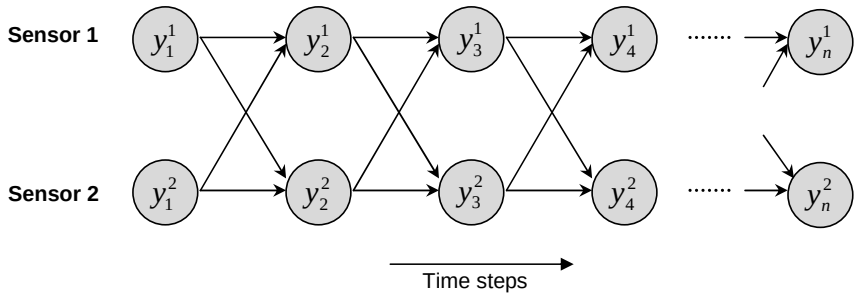


Figure 6.2: Optimal sensor selection with two sensors (or observation/data sources) for n time steps. $y_i^{k_i}$ represents the sample observation from the k_i^{th} sensor at the i^{th} time step.

where $C_{n-1} = \Lambda(n-1) + \log\left(\frac{\sigma_{0,k}}{\sigma_{1,k}}\right)$. The above expression can be re-written as:

$$P\left(\Lambda(n) > b/H_1\right) = 1 + \frac{1}{2}\cdot \text{erfc}(\alpha_2') - \frac{1}{2}\cdot \text{erfc}(\alpha_1'), \quad \text{for} \quad \sigma_{0,k} \neq \sigma_{1,k} \tag{6.11}$$

and

$$P(\Lambda(n) > b/H_1) = \frac{1}{2}\text{erfc}\left(\frac{-\frac{C}{B} - \mu_{1,k}}{\sqrt{2}\sigma_{1,k}}\right), \quad \text{for} \quad \sigma_{0,k} = \sigma_{1,k}. \tag{6.12}$$

Similarly, we can write:

$$\begin{aligned} P(\Lambda(n) < a/H_0) &= P\left(\Lambda(n-1) + \log\left(\frac{\sigma_{0,k}}{\sigma_{1,k}}\right) + \frac{(y_n^k - \mu_{0,k})^2}{2\sigma_{0,k}^2} - \frac{(y_n^k - \mu_{1,k})^2}{2\sigma_{1,k}^2} > a/H_0\right) \\ &= P\left(\frac{(y_n^k - \mu_{0,k})^2}{2\sigma_{0,k}^2} - \frac{(y_n^k - \mu_{1,k})^2}{2\sigma_{1,k}^2} < a - C_{n-1}/H_0\right) \end{aligned} \tag{6.13}$$

Thus:

$$P\left(\Lambda(n) < a/H_0\right) = \frac{1}{2}\text{erfc}(\beta_1') - \frac{1}{2}\text{erfc}(\beta_2'), \quad \text{for} \quad \sigma_{0,k} \neq \sigma_{1,k} \tag{6.14}$$

and

$$P\left(\Lambda(n) < a/H_0\right) = 1 - \frac{1}{2}\text{erfc}\left(\frac{-\frac{C'}{B} - \mu_{0,k}}{\sqrt{2}\sigma_{0,k}}\right), \quad \text{for} \quad \sigma_{0,k} = \sigma_{1,k} \tag{6.15}$$

The details are given in Appendix C. We now consider two scenarios under the Gaussian assumption.

6.5.1 Homogeneous Model: Each sensor has the same variance across hypotheses

In the homogeneous model, we assume that a sensor has the same variance across hypotheses, i.e. $\sigma_{0,k} = \sigma_{1,k} = \sigma_k$ or $A = 0$. Under this assumption we can express $P_C(k,n)$ as:

$$\begin{aligned} P_C(k,n) &= \frac{1}{2}\text{erfc}\left(\frac{-\frac{C}{B} - \mu_{1,k}}{\sqrt{2}\sigma_k}\right)\cdot P^p(H_1) + \left[1 - \frac{1}{2}\text{erfc}\left(\frac{-\frac{C'}{B} - \mu_{0,k}}{\sqrt{2}\sigma_k}\right)\right]\cdot P^p(H_0) \\ &= \frac{1}{2}\text{erfc}\left(\frac{-\frac{C}{B} - \mu_{1,k}}{\sqrt{2}\sigma_k}\right) + \left[1 - \frac{1}{2}\text{erfc}\left(\frac{-\frac{C'}{B} - \mu_{0,k}}{\sqrt{2}\sigma_k}\right) - \frac{1}{2}\text{erfc}\left(\frac{-\frac{C}{B} - \mu_{1,k}}{\sqrt{2}\sigma_k}\right)\right]\cdot P^p(H_0) \end{aligned}$$

Thus the sensor selection rule is:

$$\begin{aligned} k_n &= \underset{k}{\text{argmax}} \quad P_C(k,n) \\ &\iff \underset{k}{\text{argmax}} \left\{\frac{1}{2}\text{erfc}\left(\frac{-\frac{C}{B} - \mu_{1,k}}{\sqrt{2}\sigma_k}\right) + \left[\frac{1}{2}\text{erf}\left(\frac{-\frac{C'}{B} - \mu_{0,k}}{\sqrt{2}\sigma_k}\right) + \frac{1}{2}\text{erf}\left(\frac{-\frac{C}{B} - \mu_{1,k}}{\sqrt{2}\sigma_k}\right)\right]\cdot P^p(H_0)\right\} \end{aligned} \tag{6.16}$$

6.5.2 Heterogeneous Model: Each sensor has different variance across hypotheses

In the heterogeneous model, we assume that a sensor has different variance across hypotheses, i.e. $\sigma_{0,k} \neq \sigma_{1,k}$ or $A \neq 0$. Thus,

$$P_C(k,n) = \left(1 + \frac{1}{2}\text{erfc}(\alpha_2') - \frac{1}{2}\text{erfc}(\alpha_1')\right) \cdot P^p(H_1) + \left(\frac{1}{2}\text{erfc}(\beta_1') - \frac{1}{2}\text{erfc}(\beta_2')\right) \cdot P^p(H_0).$$

The sensor selection rule is:

$$\begin{aligned} k_n = \arg\max_k \quad & \left[\left(1 + \frac{1}{2}\text{erfc}(\alpha_2') - \frac{1}{2}\text{erfc}(\alpha_1')\right)\right. \\ & \left. + \frac{1}{2}\left(\text{erfc}(\alpha_1') - \text{erfc}(\alpha_2') + \text{erfc}(\beta_1') - \text{erfc}(\beta_2') - 2\right) \cdot P^p(H_0)\right] \end{aligned} \tag{6.17}$$

It should be noted that the greedy heuristic based selection and sequential detection strategy can be extended to other probabilistic models using the general framework presented in Sections 6.2 and 6.3.

6.6 Sensor Selection based on Distance between Hypotheses: Simplified Heuristic

In this section we attempt to obtain a simplified heuristic under the Gaussian probabilistic model based on the expressions presented in Section 6.5. It will be shown that for the Gaussian case the simplified heuristic provides a measure of the distance between the hypotheses.

6.6.1 Homogeneous Model

The following upper bounds for $\text{erfc}(x)$ are well known [78]:

$$\text{erfc}(x) < \text{e}^{-x^2} \leq \frac{1}{(1+x^2)} \tag{6.18}$$

Using (6.16), (6.18) and the notation from Appendix C we can approximate the greedy heuristic based sensor selection rule as:

$$k_n = \underset{k}{\text{argmax}} \quad \frac{1}{2}\text{erfc}\underbrace{\left(\frac{-\frac{C}{B}-\mu_{1,k}}{\sqrt{2}\sigma_k}\right)}_{f_{1k}} + \left[1-\frac{1}{2}\text{erfc}\underbrace{\left(\frac{-\frac{C'}{B}-\mu_{0,k}}{\sqrt{2}\sigma_k}\right)}_{f_{2k}} - \frac{1}{2}\text{erfc}\left(\frac{-\frac{C}{B}-\mu_{1,k}}{\sqrt{2}\sigma_k}\right)\right]\cdot P^p(H_0)$$

$$\iff \underset{k}{\text{argmax}} \quad \frac{1}{2}\left(\frac{1}{1+f_{1k}^2}\right) + \left[1-\frac{1}{2}\left(\frac{1}{1+f_{2k}^2}\right) - \frac{1}{2}\left(\frac{1}{1+f_{1k}^2}\right)\right]\cdot P^p(H_0) \tag{6.19}$$

$$\iff \underset{k}{\text{argmax}} \quad \left(\frac{\mu_{1,k}-\mu_{0,k}}{\sigma_k}\right) \tag{6.20}$$

where $f_{1k} = \frac{(b-C_{n-1})\sigma_k}{\sqrt{2}(\mu_{1,k}-\mu_{0,k})} - \frac{(\mu_{1,k}-\mu_{0,k})}{2\sqrt{2}\sigma_k}$ and $f_{2k} = \frac{(\mu_{1,k}-\mu_{0,k})}{2\sqrt{2}\sigma_k} - \frac{(C_{n-1}-a)\sigma_k}{\sqrt{2}(\mu_{1,k}-\mu_{0,k})}$ after simple algebraic manipulations. It can be seen that f_{1k} is monotonically decreasing and f_{2k} monotonically increasing with $\left(\frac{\mu_{1,k}-\mu_{0,k}}{\sigma_k}\right)$. (6.19) is obtained after using the upper bounds from (6.18). (6.20) follows from the monotonicity properties of f_{1k} and f_{2k}. We assume that the second term in (6.19) can be safely left out since f_{1k} and f_{2k} are monotonic in opposite directions with similar rates of change.

6.6.2 Heterogeneous Model

Starting with (6.17) and following an approach similar to that used for the homogeneous model above we obtain:

$$k_n = \underset{k}{\text{argmax}} \quad [\text{erfc}(\alpha_2') - \text{erfc}(\alpha_1')]\cdot(1-P^p(H_0)) - [\text{erfc}(\beta_1') - \text{erfc}(\beta_2')]\cdot P^p(H_0)$$

$$\iff \underset{k}{\text{argmax}} \quad \left[\frac{1}{1+{\alpha_2'}^2} - \frac{1}{1+{\alpha_1'}^2}\right]\cdot(1-P^p(H_0)) + \left[\frac{1}{1+{\beta_2'}^2} - \frac{1}{1+{\beta_1'}^2}\right]\cdot P^p(H_0) \tag{6.21}$$

$$\iff \underset{k}{\text{argmax}} \quad \left(\frac{(\mu_{1,k}-\mu_{0,k})^2}{\sigma_{1,k}\cdot\sigma_{0,k}}\right) \tag{6.22}$$

where $\alpha_1' = -\frac{\sigma_{1,k}}{\sqrt{2}}\left(\frac{\mu_{1,k}-\mu_{0,k}}{\sigma_{1,k}^2-\sigma_{0,k}^2}\right) - \sigma_{0,k}\sqrt{\frac{1}{2}\left(\frac{\mu_{1,k}-\mu_{0,k}}{\sigma_{1,k}^2-\sigma_{0,k}^2}\right)^2 + \left(\frac{b-C_{n-1}}{\sigma_{1,k}^2-\sigma_{0,k}^2}\right)}$. To better understand the monotonic property the following approximation can be used: $\alpha_1' \approx -\frac{1}{\sqrt{2}}\left(\frac{\mu_{1,k}-\mu_{0,k}}{\sigma_{1,k}-\sigma_{0,k}}\right)$. Similar expressions can be obtained for α_2', β_1' and β_2'. Using these approximate expressions in (6.21) followed by algebraic manipulations results in (6.22). It can be seen that the simplified heuristic in (6.22) is a measure of the distance between the two hypotheses.

It must be noted that the simplified heuristic in this section are derived for the Gaussian model. For other probabilistic models such simplified expressions may be difficult to obtain in which case the greedy heuristic (6.6) will be used for sensor selection.

6.7 Fixed Sample Size Detector

In this section we derive a fixed sample size (FSS) detector. An FSS detector is designed to operate on a block of data. A test statistic is computed based on a fixed number of observations and compared against a threshold [79]. For the binary hypothesis problem with two sources in (6.9), consider an FSS detector with the decision rule $\Lambda(n_{FSS}) \underset{H_0}{\overset{H_1}{\gtrless}} \tilde{\gamma}$ with n_{FSS} being the fixed number of observations. The log-likelihood ratio can be written as:

$$\begin{aligned}\Lambda(n_{FSS}) &= \log\left\{\prod_{i=1}^{n_{FSS}} \frac{p(y_i^{k_i}/H_1)}{p(y_i^{k_i}/H_0)}\right\} \\ &= \log\left\{\prod_{i_1=1}^{n_1} \frac{p(y_{i_1}^1/H_1)}{p(y_{i_1}^1/H_0)} \cdot \prod_{i_2=1}^{n_2} \frac{p(y_{i_2}^2/H_1)}{p(y_{i_2}^2/H_0)}\right\}\end{aligned}$$

where the first product term corresponds to observations from sensor source $k_i = 1$ and the second term is due to observations from the source $k_i = 2$. Also, $n_1 + n_2 = n_{FSS}$. Assuming Gaussian distributions (6.9):

$$\begin{aligned}\Lambda(n_{FSS}) = \log\left\{\left(\frac{\sigma_{0,1}}{\sigma_{1,1}}\right)^{2n_1} \cdot \left(\frac{\sigma_{0,2}}{\sigma_{1,2}}\right)^{2n_2}\right\} &+ \sum_{i_1=1}^{n_1}\left[\frac{\left(y_{i_1}^1 - \mu_{0,1}\right)^2}{2\sigma_{0,1}^2} - \frac{\left(y_{i_1}^1 - \mu_{1,1}\right)^2}{2\sigma_{1,1}^2}\right] \\ &+ \sum_{i_2=1}^{n_2}\left[\frac{\left(y_{i_2}^2 - \mu_{0,2}\right)^2}{2\sigma_{0,2}^2} - \frac{\left(y_{i_2}^2 - \mu_{1,2}\right)^2}{2\sigma_{1,2}^2}\right] \qquad (6.23)\end{aligned}$$

6.7.1 Homogeneous Model

Under the homogeneous model ($\sigma_{0,k} = \sigma_{1,k} = \sigma_k, k = 1, 2$), (6.23) reduces to:

$$\begin{aligned}\Lambda(n_{FSS}) &= \frac{(\mu_{1,1} - \mu_{0,1})}{\sigma_1^2}\sum_{i_1=1}^{n_1} y_{i_1}^1 + \frac{(\mu_{1,2} - \mu_{0,2})}{\sigma_2^2}\sum_{i_2=1}^{n_2} y_{i_2}^2 + \text{constant terms} \underset{H_0}{\overset{H_1}{\gtrless}} \tilde{\gamma} \\ &= c_1\sum_{i_1=1}^{n_1} y_{i_1}^1 + c_2\sum_{i_2=1}^{n_2} y_{i_2}^2 + c_3 \underset{H_0}{\overset{H_1}{\gtrless}} \tilde{\gamma}\end{aligned}$$

Thus the decision rule for the FSS detector can be written as:

$$L = c_1\sum_{i_1=1}^{n_1} y_{i_1}^1 + c_2\sum_{i_2=1}^{n_2} y_{i_2}^2 \underset{H_0}{\overset{H_1}{\gtrless}} \gamma \qquad (6.24)$$

The conditional distributions of L under the two hypotheses are:

$$p(L/H_0) = \mathcal{N}(M_0, S)$$

$$p(L/H_1) = \mathcal{N}(M_1, S)$$

where $M_0 = c_1 n_1 \mu_{0,1} + c_2 n_2 \mu_{0,2}$, $M_1 = c_1 n_1 \mu_{1,1} + c_2 n_2 \mu_{1,2}$ and $S = c_1^2 n_1 \sigma_1^2 + c_2^2 n_2 \sigma_2^2$. Thus we obtain the following constraints based on the desired P_D and P_{FA}:

$$\begin{aligned} \frac{1}{2}\text{erfc}\left(\frac{\gamma - M_1}{\sqrt{2S}}\right) &\geq P_D \\ \frac{1}{2}\text{erfc}\left(\frac{\gamma - M_0}{\sqrt{2S}}\right) &\leq P_{FA} \end{aligned} \tag{6.25}$$

Using (6.25) we obtain:

$$\frac{M_1 - M_0}{\sqrt{2S}} \geq \text{erfc}^{-1}(2P_{FA}) - \text{erfc}^{-1}(2P_D) = \kappa \tag{6.26}$$

We are trying to solve for n_1 and n_2, while trying to minimize the total number of samples (n_{FSS}). This can be formulated as the following optimization problem:

$$\begin{aligned} &\min_{n_1, n_2} \quad n_1 + n_2 \\ \text{subject to} \quad &\tfrac{(\mu_{1,1} - \mu_{0,1})^2}{\sigma_1^2} n_1 + \tfrac{(\mu_{1,2} - \mu_{0,2})^2}{\sigma_2^2} n_2 \geq 2\kappa^2 \\ &n_1 + n_2 \geq 1 \\ &n_1 \geq 0, n_2 \geq 0 \end{aligned} \tag{6.27}$$

6.7.2 Heterogeneous Model

The log-likelihood ratio (6.23) indicates that the FSS detector under the heterogeneous model is somewhat intractable. In addition the FSS detector is unable to adapt to non-stationarity of the measurements such as time-varying means.

6.8 Sensor Selection for M-ary Hypothesis Testing

In this section we present the framework for extending sensor selection and event detection to scenarios involving a decision between M ($M > 2$) simple hypotheses, $H_1, H_2, \ldots, H_M$. Let us define the following pairwise log-likelihood ratio [80]:

$$\Lambda_{p,q}(n) = \log\left\{\frac{p(y_1^{k_1}, \cdots, y_n^{k_n}/H_p)}{p(y_1^{k_1}, \cdots, y_n^{k_n}/H_q)}\right\}$$

for $p, q = 1, \ldots, M$. The binary sequential hypothesis testing procedure, outlined in Section 6.2, can be modified for the M-hypotheses case as follows:

1. Compute $\Lambda_{p,q}(n), p, q = 1, \ldots, M, p \neq q$.

2. Decide H_p if $\Lambda_{p,q}(n) > A_{p,q}\ \forall q = 1, \ldots, M, q \neq p$, else

3. perform another measurement and repeat the test.

The thresholds $A_{p,q}$ are given as [81]:

$$A_{p,q} = \frac{1}{P_e(p,q)} \cdot \left(1 - \sum_{q \neq p} P_e(q,p)\right) \tag{6.28}$$

where $P_e(p,q)\ p \neq q$ is the probability of incorrectly deciding in favor of H_p when H_q is the true hypothesis. Thus, $P_e(p,p)$ is the probability of correctly deciding in favor of H_p. Also, note that $\Sigma_{q=1}^{M} P_e(p,q) = 1$.

In addition to meeting the performance constraints imposed by these probabilities, we would like to make a decision as soon as possible. This can be achieved by collecting as few sensor observations as possible in order to make a decision while meeting the detection performance. We consider the following two sensor selection strategies:

1. Select a sensor, for the next observation instant, which maximally separates the most likely hypothesis from all the other hypotheses, i.e., mathematically:

$$k_n = \underset{k}{\operatorname{argmax}} \left\{ \sum_{\substack{q=1 \\ q \neq m_l}}^{M} d_{m_l,q}^{k} \right\} \tag{6.29}$$

   where $d_{p,q}^{k}$ is a measure of the separation between hypotheses H_p and H_q based on the observations from sensor k, and H_{m_l} is the most likely hypothesis based on the posteriori probabilities (6.7).

2. Select a sensor, for the next observation instant, which maximally separates the two most likely hypotheses, i.e., mathematically:

$$k_n = \underset{k}{\operatorname{argmax}} \left\{ d_{m_{l1},m_{l2}}^{k} \right\} \tag{6.30}$$

   where $H_{m_{l1}}$ and $H_{m_{l2}}$ are the two most likely hypotheses based on the posteriori probabilities.

Assuming Gaussian models for the different hypotheses, we can express the separation between two hypotheses as $d_{p,q}^{k} = \left(\frac{(\mu_{p,k} - \mu_{q,k})^2}{\sigma_{p,k} \cdot \sigma_{q,k}} \right)$, based on the simplified heuristic derived in Section 6.6.

6.9 Sensor Selection and Detection Performance under Gaussian Modeling

For the experiments in this section, we consider the following binary hypothesis testing case with two sensors (i.e., $N = 2$ for simplicity of illustration):

$$\begin{aligned} H_0: & \quad y^1 \sim \mathcal{N}(\mu_{0,1}, \sigma^2_{0,1}) \\ & \quad y^2 \sim \mathcal{N}(\mu_{0,2}, \sigma^2_{0,2}) \\ H_1: & \quad y^1 \sim \mathcal{N}(\mu_{1,1}, \sigma^2_{1,1}) \\ & \quad y^2 \sim \mathcal{N}(\mu_{1,2}, \sigma^2_{1,2}) \end{aligned}$$

To investigate the sensor selection performance, we compare the greedy heuristic based sensor selection with the optimal sensor selection strategy. The average sequence length needed for detection using the greedy heuristic and the optimal selection strategies are computed using 1000 runs of the Monte Carlo simulation. Fig. 6.3 shows the results for the Gaussian homogeneous model with time-invariant mean. It can be seen that on average the number of samples needed for the greedy heuristic based sensor selection strategy is within 10% of that needed using the optimal selection strategy.

Next, we discuss the performance of the following strategies for the sensor selection and event detection problems:

1. Greedy heuristic based sensor selection and sequential detection.

2. Distance between hypotheses (simplified heuristic) based sensor selection and sequential detection.

3. Fixed sample size detector.

We compare the strategies in terms of the average number of samples required for detection. The average number of samples is a measure of the *speed* of the detector, since smaller this number is, the faster the detection will be. We investigate the performance under both homogeneous and heterogeneous Gaussian models. In addition, for each model we investigate the effect of time varying statistics on the detection performance. Specifically, we consider the effect of time varying means. The results of 5000 runs of Monte Carlo simulations are averaged to obtain the performance curves. The performance criterion for detection are: $P_{FA} = 1 \times 10^{-5}$ and P_D is varied from 0.85 to 0.99.

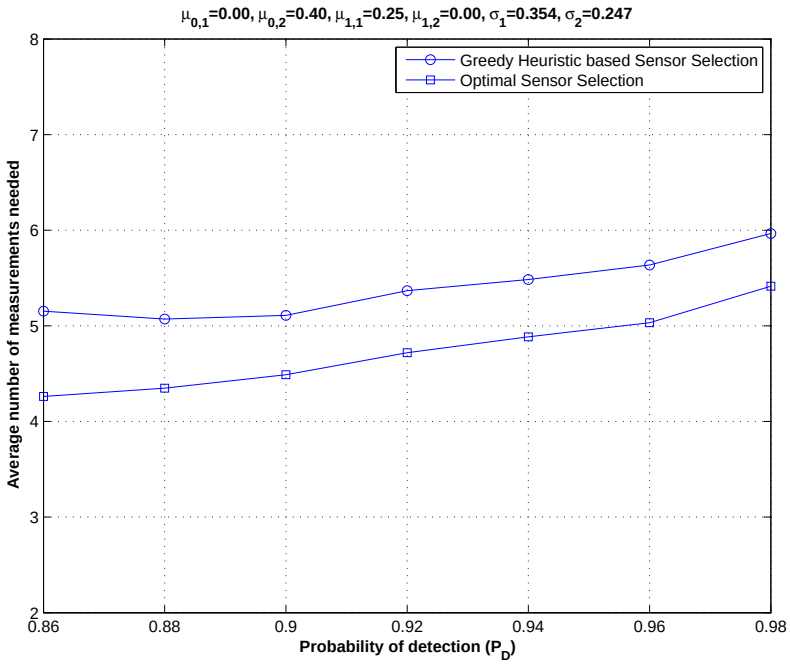


Figure 6.3: Comparison of the greedy heuristic based sensor selection with the *optimal* sensor selection.

6.9.1 Performance under Homogeneous Model

Distributions with Time Invariant Mean

We assume that the means of the Gaussian distributions are time-invariant. Fig. 6.4 shows that the strategies based on greedy heuristic and the simplified heuristic have very similar performance. In addition these methods need on average less than half the number of samples for detection as the FSS detector (i.e., the FSS detector needs at least twice the amount of time for detection).

Distributions with Time Varying Mean

Next we assume that the means of the Gaussian distributions are time varying. Specifically,

$$\mu_{m,k} := \mu_{m,k}(t) \sim \mathcal{N}(\mu_{m,k}, \sigma_\mu^2) \quad \text{for} \quad m = 0, 1 \quad \text{and} \quad k = 1, 2.$$

Fig. 6.5 shows that the greedy heuristic based method takes on average about 20% more samples than the method based on the simplified heuristic. However, both these methods significantly outperform an FSS detector. This indicates that under the homogeneous model the sensor selection

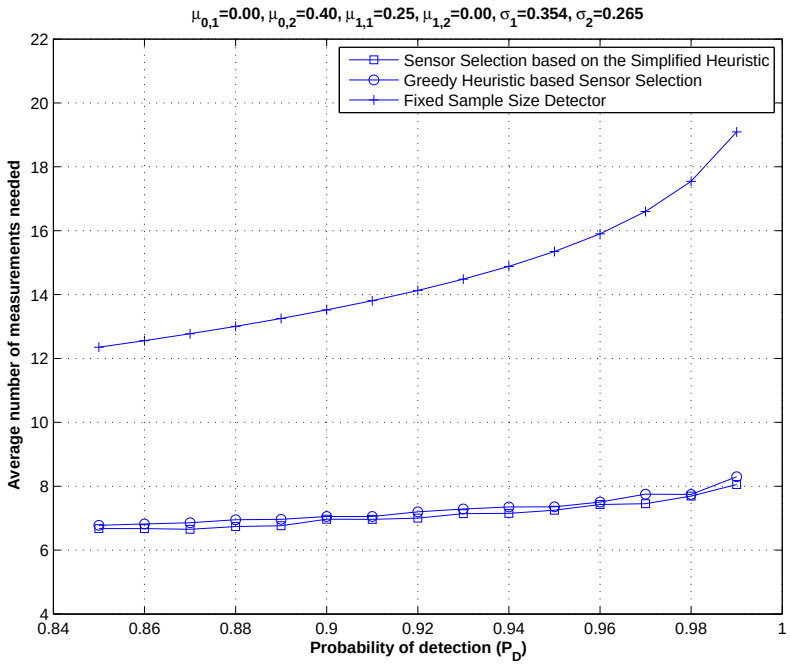


Figure 6.4: Gaussian Homogeneous Model with time invariant mean.

based on the simplified heuristic which is analogous to maximizing the distance between hypotheses takes better advantage of the time varying statistics.

The proposed strategies extend in a straightforward way for a large number of sensors ($N > 2$). Fig. 6.6 illustrates the detection performance when four sensors are available for measurement under the Gaussian homogeneous model with time varying means.

6.9.2 Performance under Heterogeneous Model

Distributions with Time Invariant Mean

For the Gaussian heterogeneous model with time-invariant means, the sensor selection based on simplified heuristic performs slightly better than the greedy heuristic, taking on average 15% less samples for detection (refer Fig. 6.7).

Distributions with Time Varying Mean

With time varying means, the greedy heuristic and the simplified heuristic based sensor selection methods result in very similar performance. Fig. 6.8 shows that the greedy and the simplified

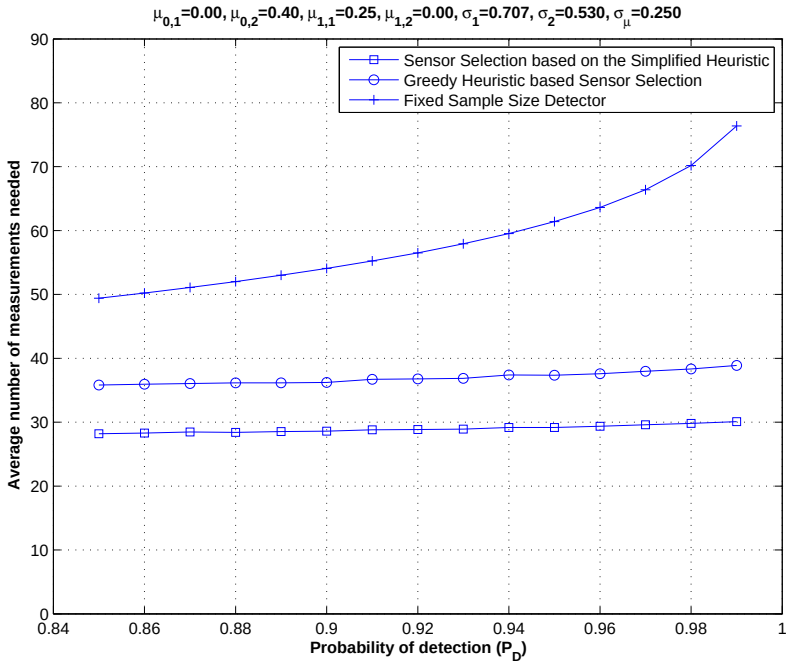


Figure 6.5: Gaussian Homogeneous Model with time varying mean.

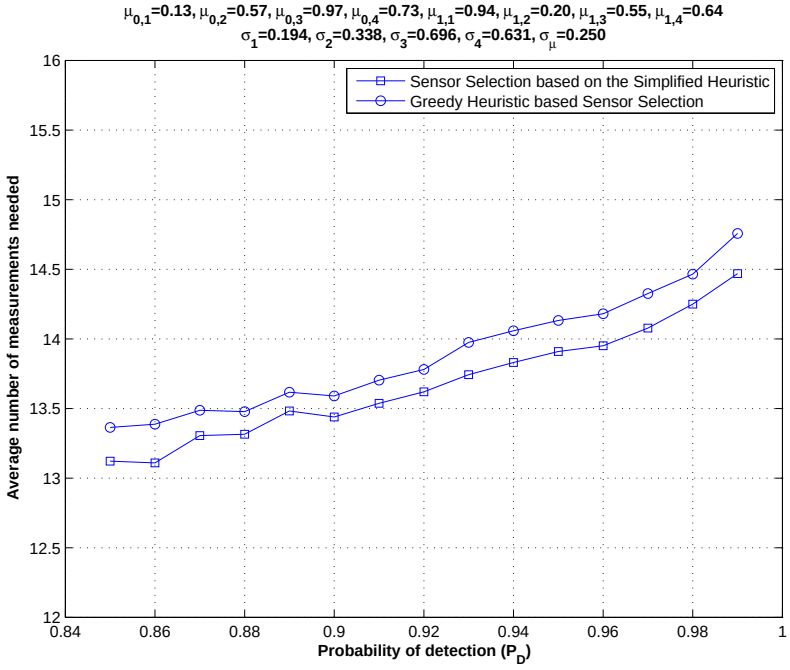


Figure 6.6: Gaussian Homogeneous Model with time varying mean for the case with four sensors ($N = 4$).

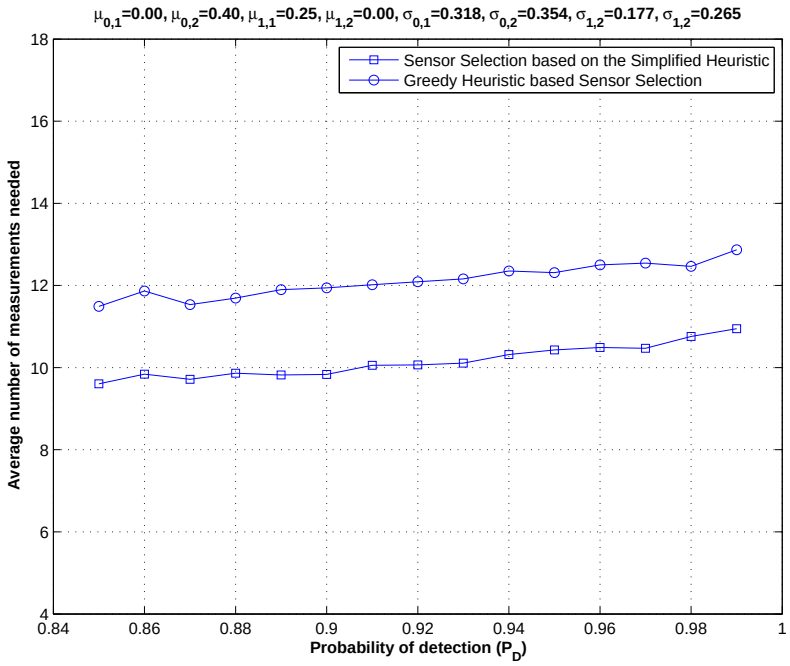


Figure 6.7: Gaussian Heterogenous Model with time invariant mean.

heuristic need about the same number of samples on average for detection.

6.9.3 M-ary Hypothesis Testing

Next we consider the M-ary hypothesis testing scenario with $M > 2$. Assume that we have two sensors or sources for the observations (i.e., $N = 2$) and there exist three mutually exclusive and exhaustive hypotheses (i.e., $M = 3$). The desired probability of making a correct decision is the same for all hypotheses i.e., $P_e(p, p)\ \forall p = 1, \ldots, M$. We also assume that the probability of making an incorrect decision is same for all incorrect decisions i.e., $P_e(p, q)\ p \neq q$. The thresholds $A_{p,q}$ are determined based on these probabilities. Fig. 6.9 shows that the average number of measurements needed for detection steadily increases with the desired probability of correct decision. It is also seen that the sensor selection strategy of maximally separating the most likely hypothesis from all other hypotheses performs slightly better than the strategy of maximally separating only the two most likely hypotheses, at the cost of some additional computations.

Thus overall we observe that the sensor selection based on the greedy heuristic and the simplified heuristic provide comparable performance. However, the greedy heuristic proves valuable

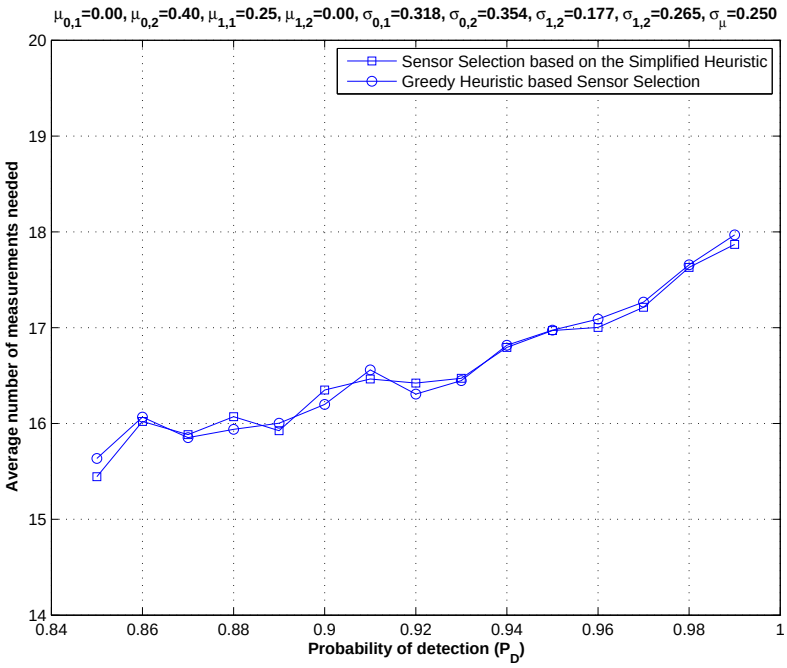


Figure 6.8: Gaussian Heterogenous Model with time varying mean.

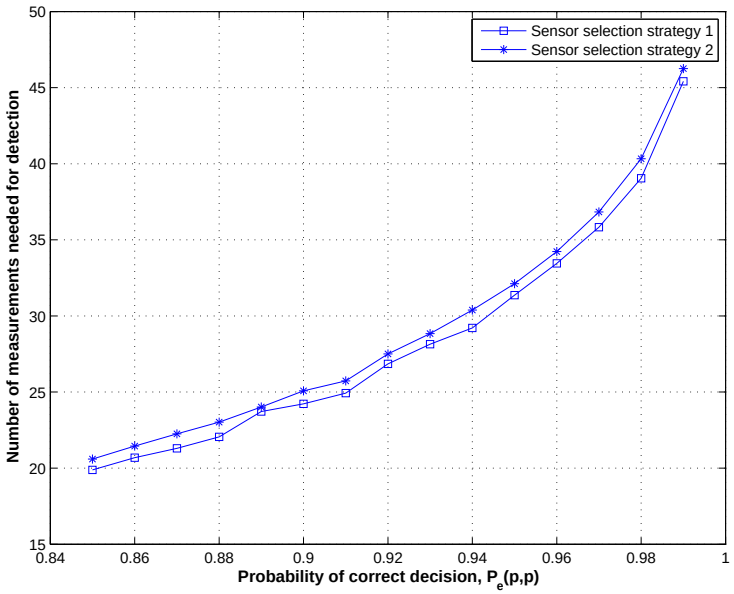


Figure 6.9: M-ary sequential hypothesis testing. Three hypotheses and two sensors ($M = 3, N = 2$). Sensor-selection strategy 1: maximally separate most likely hypothesis from all other hypotheses. Sensor-selection strategy 2: maximally separate the two most likely hypotheses.

for probabilistic models where a simplified heuristic is difficult to derive. In addition, these sensor selection methods combined with sequential detection perform significantly better than the FSS detector.

6.10 Conclusion

A greedy heuristic based sensor selection and a sequential detection procedure has been presented that significantly improves the detection speed. The greedy heuristic was expressed in closed form for the Gaussian model and can be extended for most probabilistic models. The numerical experiments have shown that the sequential procedure significantly outperforms the FSS detector in that it always needs fewer measurements on average to achieve the same detection performance. A simplified heuristic, which provided a measure of the distance between the hypotheses, was derived for the Gaussian probabilistic model. The simplified heuristic was seen to perform as good as or slightly better than the greedy heuristic. The greedy heuristic provides a general framework applicable to any probabilistic model and for models which do not yield a simplified heuristic. Extension of the sensor selection and sequential hypothesis testing procedure for M-ary hypotheses ($M > 2$) was also presented.

Chapter 7

Conclusions and Future Directions

7.1 Conclusions

The research presented in this thesis addresses the localization problem in wireless sensor networks. We presented two localization algorithms, based on the ultra-wideband channel model, suitable for multipath propagation environments. Simulation results using these algorithms have shown the ability to achieve close to sub-meter ranging accuracy. In addition, it was observed that the positioning accuracy is limited only by the bandwidth of the transmit (or ranging) signal and the signal-to-noise ratio.

Next we presented a distributed algorithm based on the second-order cone programming relaxation which solves the localization problem in the presence of inaccuracies in the reference node positions and distance measurements, with significant computational savings and without sacrificing positioning accuracy. This algorithm is well-suited for large networks with thousands of nodes. An extensive numerical study of the algorithm under different scenarios was presented. This method is also able to improve positioning of the reference nodes which are in the convex hull of their neighbors.

We also presented the prototype implementation for a directional beacon based positioning algorithm using radio frequency signals. Novel techniques for improving the position estimation accuracy using maximum likelihood amplitude estimation, least squares based time-delay estimation and combining data from multiple antenna rotations, were presented. The positioning accuracy of the directional beacon based algorithm depends on the width of the antenna beampat-

tern and the rotational speed of the directional antenna. The experiments demonstrated the ability to obtain position estimation results with sub-meter accuracy. The robustness of the algorithm (and the implementation) to timing and synchronization errors was also demonstrated.

Finally, a heuristic based sensor selection and sequential detection procedure was presented that significantly improves the detection speed. The sequential procedure significantly outperforms the fixed sample size detector in that it always needs fewer measurements on average to achieve the same detection performance. A greedy heuristic was presented as a general framework applicable to any probabilistic model. A simplified heuristic, which provided a measure of the distance between the hypotheses, was derived for the Gaussian probabilistic model.

7.2 Future Directions

Below we describe some promising areas and/or problems for future investigation.

7.2.1 Implementation of Localization Algorithms for UWB Multipath Environments

It will be interesting to implement the localization algorithms presented in Chapter 3 on hardware testbeds. The availability of UWB radio modules which offer enough flexibility to implement user-defined algorithms is crucial in this regard. The currently available commercial modules do not seem to offer such flexibility.

7.2.2 Distributed SOCP Algorithm for RSSI based Localization

One of the advantages of the proposed distributed SOCP algorithm is the ability to handle large distance estimation errors which are common in RSSI based schemes. The effectiveness of this algorithm with experimental RSSI data is something that needs to be explored.

We have shown the convergence of the distributed SOCP algorithm numerically. It is desirable to demonstrate the convergence theoretically.

7.2.3 Sensor Selection with Unknown Parameters and other Probabilistic Models

It was seen that the greedy heuristic for sensor selection, under the Gaussian model, reduces to a simplified heuristic that maximizes the distance between the two hypotheses. It is desirable to understand if this is true for any generic probabilistic model. It is also not clear at this point how unknown parameters in the probability distributions would be handled for the sensor selection problem.

Appendix A

SOCP Problem Formulation in SeDuMi form

SeDuMi solves problems of the form:

$$\max\ b^T y \quad \text{s.t.} \quad c - A^T y \in K^* \tag{A.1}$$

where K^* is the dual cone. We now express (4.5) in this form. Assuming 2-D ($d = 2$), we define:

$$L_i := \begin{bmatrix} 1 & 0^{1\times(4n_i+2)} \end{bmatrix}$$

$$r_j := \begin{bmatrix} 0 & d_{ij}^2 \end{bmatrix}^T, t_j := \begin{bmatrix} 0 & 0 & x_{j1} & x_{j2} \end{bmatrix}^T$$

$$S_j := \begin{bmatrix} 0^{1\times(2n_i+3)} & 1_{ij} & 0^{1\times n_i} \\ 0^{1\times(2n_i+3)} & 0^{1\times n_i} & 1_{ij} \end{bmatrix}$$

$$U_j := \begin{bmatrix} 0.5 & 0^{1\times(2n_i+2)} & 0^{1\times n_i} & 0.5(1_{ij}) \\ -0.5 & 0^{1\times(2n_i+2)} & 0^{1\times n_i} & 0.5(1_{ij}) \\ 0 & 1_{i1} & 0^{1\times n_i} & 0^{1\times n_i} \\ 0 & 1_{i2} & 0^{1\times n_i} & 0^{1\times n_i} \end{bmatrix}$$

where $j \in N_A(i)$, $\mathbf{1_{ij}}$ is a row vector of length n_i and has a 1 corresponding to the variable t_{ij} and zero elsewhere. Similarly, $\mathbf{1_{i1}}$, $\mathbf{1_{i2}}$ are row vectors each of length $(2n_i + 2)$ with a 1 corresponding to the coordinates of $x_i = (x_{i1}, x_{i2})$ and zero elsewhere.

Define:

$$S := \begin{bmatrix} S_1 \\ \vdots \\ S_{n_i} \end{bmatrix}, r := \begin{bmatrix} r_1 \\ \vdots \\ r_{n_i} \end{bmatrix}, U := \begin{bmatrix} U_1 \\ \vdots \\ U_{n_i} \end{bmatrix}, t := \begin{bmatrix} t_1 \\ \vdots \\ t_{n_i} \end{bmatrix}, y := \begin{bmatrix} t_i' \\ \mathbf{x} \\ \mathbf{t_{ij}} \\ \mathbf{y_{ij}} \end{bmatrix}$$

where $\mathbf{x} = \begin{bmatrix} x_{i1} & x_{i2} \end{bmatrix}^{\mathbf{T}}_{\mathbf{(2n_i+2)\times 1}}$, $\mathbf{t_{ij}} = \begin{bmatrix} t_{ij} \end{bmatrix}_{\mathbf{n_i \times 1}}$ and $\mathbf{y_{ij}} = \begin{bmatrix} y_{ij} \end{bmatrix}_{\mathbf{n_i \times 1}}$. Then the conic constraints can be expressed as $r - S^T y \in \text{Qcone}^2 \times \ldots \times \text{Qcone}^2$, and $t - U^T y \in \text{Qcone}^4 \times \ldots \times \text{Qcone}^4$, where the Cartesian product is taken over n_i cones. ($\text{Qcone}^k = \{(x, y) \in R \times R^{k-1} : x \geq \|y\|\}$).

Defining:

$$\tilde{b} := -\begin{bmatrix} 0 \\ 0^{2(n_i+1)\times 1} \\ 1^{n_i \times 1} \\ 0^{n_i \times 1} \end{bmatrix}, \tilde{A} := \begin{bmatrix} L_i \\ S \\ U \end{bmatrix}^T, \tilde{c} := -\begin{bmatrix} 1 \\ r \\ t \end{bmatrix}$$

the problem (4.5) can be written in the form (A.1) with $b = \tilde{b}$, $A = \tilde{A}$, $c = \tilde{c}$ and K being the Cartesian product of all the cones.

Appendix B

Lower Bound on Position Estimation Error Variance

Consider the signal model described by (5.25). For our experimental setup, it is sufficient to consider $M = 2$. Then the vector of parameters to be estimated is $\boldsymbol{\lambda} := [\tau_1 \quad \tau_2 \quad a_2]$. We ignore a_1 as it can always be normalized to unity. Using the Cramer-Rao lower bound (CRLB) we can bound the variance as

$$\text{var}(\lambda_i) \geq [\mathbf{I}^{-1}(\boldsymbol{\lambda})]_{ii} \quad \text{for} \quad i = 1, 2, 3 \tag{B.1}$$

where $\mathbf{I}(\boldsymbol{\lambda})$ is a 3×3 Fisher Information matrix (FIM) with $[\mathbf{I}(\boldsymbol{\lambda})]_{ij} = -E\left[\frac{\partial^2 p(\tilde{\mathbf{y}}^{(L)}, \boldsymbol{\lambda})}{\partial \lambda_i \partial \lambda_j}\right]$. To avoid the matrix inversion in (B.1), we invoke the property that a FIM satisfies [72, p. 65]:

$$[\mathbf{I}^{-1}(\boldsymbol{\lambda})]_{ii} \geq \frac{1}{[\mathbf{I}(\boldsymbol{\lambda})]_{ii}}. \tag{B.2}$$

This leads to a loose bound: $\text{var}(\lambda_i) \geq \frac{1}{[\mathbf{I}(\boldsymbol{\lambda})]_{ii}}$, that is easier to compute though difficult to attain, in general. It can be shown that for time-delay estimation, the diagonal elements of the FIM are given by [72]:

$$[\mathbf{I}(\boldsymbol{\lambda})]_{ii} = \frac{\xi}{N_o/2} \bar{F}^2 \quad \text{for} \quad i = 1, 2 \tag{B.3}$$

where ξ is the signal energy, $\bar{F}^2$ is the mean square bandwidth of the signal and $N_o/2$ is the noise variance. Thus,

$$\text{var}(\tau_i) \geq \frac{1}{\frac{\xi}{N_o/2} \bar{F}^2} \quad \text{for} \quad i = 1, 2. \tag{B.4}$$

Without loss of generality, we can assume that $\hat{\tau}_1 < \hat{\tau}_2$. Thus, for the signal received from RN-1 we have $t_1 = \hat{\tau}_1$ and the same principle applies to the estimation of t_2 and t_3.

For the sake of clarity, we assume that the error associated with the estimated coordinates $(\hat{x}, \hat{y})$ of node Q is due to the error in the estimation of t_1 or $\hat{\tau}_1$. In order to obtain a bound on the variance of position estimation error $(\mathcal{E})$ we proceed as follows. The position estimation error $\mathcal{E}$ is defined as

$$\mathcal{E}^2 := (\hat{x} - x)^2 + (\hat{y} - y)^2 \tag{B.5}$$

where $\hat{x}$ and $\hat{y}$ are the estimated values of x and y co-ordinates, respectively. Taking expectation of (B.5) we can write

$$E(\mathcal{E}^2) = \text{var}(\hat{x}) + \text{var}(\hat{y}) \tag{B.6}$$

To compute $\text{var}(\hat{x})$, we assume that the error in $\hat{x}$ is due to the error in the estimation of t_1. Using (5.1) and (5.2) we can write

$$\begin{aligned} \hat{x} &= L_2 \cos\gamma \sin\gamma + L_2 \cot\beta \cos^2\gamma \\ &:= g(t_1) \end{aligned}$$

where the dependence on t_1 stems from the fact that γ depends on t_1. Now, the CRLB for $\hat{x}$ can be expressed as

$$\text{var}(\hat{x}) \geq K_x^2 \cdot \text{var}(\tau_1). \tag{B.7}$$

where $K_x := \partial g(t_1)/\partial t_1$. Similarly, if we define $\hat{y} := h(t_1)$ and $K_y := \partial h(t_1)/\partial t_1$ then using (B.4) we can bound the mean square value of the estimation error as

$$\boxed{E(\mathcal{E}^2) \geq \left(K_x^2 + K_y^2\right) \cdot \frac{1}{\frac{\xi}{N_o/2}\bar{F}^2}}$$

where $\frac{\xi}{N_o/2}$ is the signal to noise ratio. For the computation of K_x and K_y, we can use chain rule so that

$$K_x = \frac{\partial g(t_1)}{\partial \gamma} \cdot \frac{\partial \gamma}{\partial \alpha} \cdot \frac{\partial \alpha}{\partial t_1} \quad \text{and} \quad K_y = \frac{\partial g(t_1)}{\partial \gamma} \cdot \frac{\partial \gamma}{\partial \alpha} \cdot \frac{\partial \alpha}{\partial t_1}. \tag{B.8}$$

The partial derivatives in (B.8) can be evaluated as

$$\frac{\partial \alpha}{\partial t_1} = \omega,$$
$$\frac{\partial \gamma}{\partial \alpha} = \frac{L_1 \sin 2\gamma}{L_1 \sin 2\alpha - 2L_2 \sin^2 \alpha},$$
$$\frac{\partial g(t_1)}{\partial \gamma} = L_2 \cos 2\gamma - L_2 \cot \beta \sin 2\gamma \quad \text{and}$$
$$\frac{\partial h(t_1)}{\partial \gamma} = -L_2 \sin 2\gamma - L_2 \cot \beta \cos 2\gamma.$$

Appendix C

Gaussian Modeling for Binary Hypothesis Testing

Using (6.10) we can write:

$$\begin{aligned} P(\Lambda(n) > b/H_1) &= P\left(\frac{(y_n^k - \mu_{0,k})^2}{2\sigma_{0,k}^2} - \frac{(y_n^k - \mu_{1,k})^2}{2\sigma_{1,k}^2} > b - C_{n-1}/H_1\right) \\ &= P\left(A(y_n^k)^2 + By_n^k + C > 0/H_1\right) \end{aligned}$$

where

$$\begin{aligned} A &= \frac{1}{2} \cdot \left(\frac{1}{\sigma_{0,k}^2} - \frac{1}{\sigma_{1,k}^2}\right) \\ B &= \frac{\mu_{1,k}}{\sigma_{1,k}^2} - \frac{\mu_{0,k}}{\sigma_{0,k}^2} \\ C &= \frac{\mu_{0,k}^2}{2\sigma_{0,k}^2} - \frac{\mu_{1,k}^2}{2\sigma_{1,k}^2} - b + C_{n-1}. \end{aligned} \tag{C.1}$$

Thus, if $A \neq 0$ (i.e., $\sigma_{0,k} \neq \sigma_{1,k}$):

$$\begin{aligned} P(\Lambda(n) > b/H_1) &= P\left(y_n^k > \frac{-B + \sqrt{B^2 - 4AC}}{2A}/H_1\right) + P\left(y_n^k < \frac{-B - \sqrt{B^2 - 4AC}}{2A}/H_1\right) \\ &= \frac{1}{2}\mathrm{erfc}(\alpha_2') + 1 - \frac{1}{2}\mathrm{erfc}(\alpha_1') \end{aligned} \tag{C.2}$$

where

$$\alpha_1' = \frac{\alpha_1 - \mu_{1,k}}{\sqrt{2}\sigma_{1,k}} \quad \text{and}$$
$$\alpha_1 = \frac{-B - \sqrt{B^2 - 4AC}}{2A},$$
$$\alpha_2' = \frac{\alpha_2 - \mu_{1,k}}{\sqrt{2}\sigma_{1,k}} \quad \text{and}$$
$$\alpha_2 = \frac{-B + \sqrt{B^2 - 4AC}}{2A}.$$

However, if $A = 0$ (i.e., $\sigma_{0,k} = \sigma_{1,k}$) then:

$$P(\Lambda(n) > b/H_1) = \frac{1}{2}\text{erfc}\left(\frac{-\frac{C}{B} - \mu_{1,k}}{\sqrt{2}\sigma_{1,k}}\right) \tag{C.3}$$

Starting with (6.13) and following the steps outlined above we can write:

$$P(\Lambda(n) < a/H_0) = P\left(\frac{-B - \sqrt{B^2 - 4AC'}}{2A} < y_n^k < \frac{-B + \sqrt{B^2 - 4AC'}}{2A}/H_0\right)$$

where A and B are as defined in (C.1) and

$$C' = \frac{\mu_{0,k}^2}{2\sigma_{0,k}^2} - \frac{\mu_{1,k}^2}{2\sigma_{1,k}^2} - a + C_{n-1}.$$

If $A \neq 0$ then:

$$P(\Lambda(n) < a/H_0) = \frac{1}{2}\text{erfc}(\beta_1') - \frac{1}{2}\text{erfc}(\beta_2') \tag{C.4}$$

and if $A = 0$:

$$P(\Lambda(n) < a/H_0) = 1 - \frac{1}{2}\text{erfc}\left(\frac{-\frac{C'}{B} - \mu_{0,k}}{\sqrt{2}\sigma_{0,k}}\right) \tag{C.5}$$

where

$$\beta_1' = \frac{\beta_1 - \mu_{0,k}}{\sqrt{2}\sigma_{0,k}} \quad \text{and}$$
$$\beta_1 = \frac{-B - \sqrt{B^2 - 4AC'}}{2A},$$
$$\beta_2' = \frac{\beta_2 - \mu_{0,k}}{\sqrt{2}\sigma_{0,k}} \quad \text{and}$$
$$\beta_2 = \frac{-B + \sqrt{B^2 - 4AC'}}{2A}.$$

Bibliography

[1] J. A. Costa, N. Patwari, and A. O. Hero, "Distributed Weighted-Multidimensional Scaling for Node Localization in Sensor Networks," *ACM Trans. Sensor Networks*, vol. 2, p. 3964, Feb. 2006.

[2] Y. Shang, W. Ruml, Y. Zhang, and M. Fromherz, "Localization from Connectivity in Sensor Networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 15, pp. 961–974, Nov. 2004.

[3] K. Holger and A. Willig, "A Short Survey of Wireless Sensor Networks." TKN Technical Reports Series, Technical University Berlin, Oct. 2003.

[4] R. Szewczyk, E. Osterweil, J. Polastre, M. Hamilton, A. Mainwaring, and D. Estrin, "Habitat Monitoring with Sensor Networks," *Commun. ACM*, vol. 47, pp. 34–40, Jun. 2004.

[5] "IEEE Standard for Information Technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - specific requirement Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)," *IEEE Std 802.15.4a-2007 (Amendment to IEEE Std 802.15.4-2006)*, pp. 1–203, 2007.

[6] K. Pahlavan, Xinrong Li, and J. P. Makela, "Indoor Geolocation Science and Technology," *IEEE Commun. Mag.*, vol. 4, pp. 112–118, Feb. 2002.

[7] A. H. Tewfik and S. Srirangarajan, "Localization in Non Line-of-sight Environments for Wireless Sensor Networks," in *Proc. 2nd European Workshop Wireless Sensor Networks (EWSN)*, pp. 410–414, Jan.-Feb. 2005.

[8] S. Srirangarajan and A. H. Tewfik, "Localization in Wireless Sensor Networks under Non Line-of-Sight Propagation," in *Proc. IEEE GLOBECOM*, vol. 6, pp. 3477–3481, Nov. 2005.

[9] S. Srirangarajan and A. H. Tewfik, "Sensor Node Localization via Spatial Domain Quasi-Maximum Likelihood Estimation," in *Proc. 14th EUSIPCO*, Sep. 2006.

[10] J.-Y. Lee and R. A. Scholtz, "Ranging in a Dense Multipath Environment using an UWB Radio Link," *IEEE J. Select. Areas Commun.*, vol. 20, pp. 1677–1683, Dec. 2002.

[11] W. C. Chung and D. S. Ha, "An Accurate Ultra Wideband Ranging for Precision Asset Location," in *Proc. IEEE Conf. Ultra Wideband Syst. and Technologies*, pp. 389–393, Nov. 2003.

[12] K. C. Ho and W. Xu, "An Accurate Algebraic Solution for Moving Source Location using TDOA and FDOA measurements," *IEEE Trans. Signal Process.*, vol. 52, pp. 2453–2463, Sep. 2004.

[13] A. Pages-Zamora and J. Vidal, "Closed-form Solution for Positioning based on Angle of Arrival Measurements," in *Proc. IEEE Int. Symp. Pers., Indoor and Mobile Radio Commun.*, vol. 4, pp. 1522–1526, Sep. 2002.

[14] J. G. Castano, M. Svensson, and M. Ekstrom, "Local Positioning for Wireless Sensor Networks based on Bluetooth," in *IEEE Radio and Wireless Conf.*, pp. 195–198, Sep. 2004.

[15] J. Hightower and G. Borriello, "Location Systems for Ubiquitous Computing," in *Computer*, vol. 34, pp. 57–66, IEEE Computer Society Press, Aug. 2001.

[16] J. Caffery and G. L. Stuber, "Overview of Radiolocation in CDMA Cellular Systems," *IEEE Commun. Mag.*, vol. 36, pp. 38–45, Apr. 1998.

[17] S. Srirangarajan, A. H. Tewfik, and Z.-Q. Luo, "Distributed Sensor Network Localization with Inaccurate Anchor Positions and Noisy Distance Information," in *Proc. ICASSP 2007*, pp. III521–III524, Apr. 2007.

[18] S. Srirangarajan, A. H. Tewfik, and Z.-Q. Luo, "Distributed Sensor Network Localization using SOCP Relaxation," *to appear IEEE Trans. on Wireless Commun.*, 2008.

[19] S. F. A. Shah, "Design of Low Complexity OFDM Schemes and Position Location Algorithm for Wireless Systems." PhD Thesis, University of Minnesota, March 2008.

[20] S. Srirangarajan, S. F. A. Shah, and A. H. Tewfik, "Directional Beacon based Positioning System using RF Signals," *to appear in Proc. 16th EUSIPCO*, Aug. 2008.

[21] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less Low-Cost Outdoor Localization for Very Small Devices," *IEEE Pers. Commun.*, vol. 7, pp. 28–34, Oct. 2000.

[22] P. Bahl and V. N. Padmanabhan, "RADAR: An In-building RF-based User Location and Tracking System," in *IEEE Infocom*, vol. 2, pp. 775–784, Mar. 2000.

[23] K. Yu and I. Oppermann, "UWB Positioning for Wireless Embedded Networks," in *Proc. of Radio and Wireless Conference*, pp. 459–462, Sep. 2004.

[24] W. C. Chung, D. S. Ha, and D. Ni, "Performance evaluation of an ultra wideband radiolocation system with directional beacon," in *IEEE Radio and Wireless Conference*, pp. 59–62, Sep. 2004.

[25] A. H. Quazi, "An Overview on the Time Delay Estimate in Active and Passive Systems for Target Localization," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-29, pp. 527–533, Jun. 1981.

[26] M. P. Wylie and J. Holtzman, "The Non-Line of Sight Problem in Mobile Location Estimation," in *Proc. 5th IEEE Int. Conf. Universal Pers. Commun.*, vol. 2, pp. 827–831, Sep. 1996.

[27] J. Borras, P. Hatrack, and N. B. Mandayam, "Decision Theoretic Framework for NLOS Identification," in *Proc. 48th IEEE Veh. Technol. Conf.*, vol. 2, pp. 1583–1587, May 1998.

[28] M. P. Wylie-Green and S. S. Wang, "Robust Range Estimation in the Presence of Non-Line-of-Sight Error," in *Proc. 54th IEEE Veh. Technol. Conf.*, vol. 1, pp. 101–105, Oct. 2001.

[29] X. Wang, Z. Wang, and B. O'Dea, "A TOA-based Location Algorithm Reducing the Errors due to NLOS Propagation," *IEEE Trans. Veh. Technol.*, vol. 52, pp. 112–116, Jan. 2003.

[30] L. Doherty, K. S. J. Pister, and L. E. Ghaoui, "Convex Position Estimation in Wireless Sensor Networks," in *IEEE Infocom*, vol. 3, pp. 1655–1663, Apr. 2001.

[31] P. Biswas and Y. Ye, "Semidefinite Programming for Ad Hoc Sensor Network Localization," in *Proc. Int. Symp. on Information Processing in Sensor Networks (IPSN)*, pp. 46–54, Springer Verlag, Apr. 2004.

[32] P. Biswas, T.-C. Liang, K.-C. Toh, T.-C. Wang, and Y. Ye, "Semidefinite Programming Approaches for Sensor Network Localization with Noisy Distance Measurements," *IEEE Trans. Autom. Sci. Eng.*, vol. 3, pp. 360–371, Oct. 2006.

[33] P. Biswas and Y. Ye, "A Distributed Method for Solving Semidefinite Programs Arising from Ad Hoc Wireless Sensor Network Localization," in *Multiscale Optimization Methods and Applications*, pp. 69–84, Jan. 2006.

[34] M. W. Carter, H. H. Jin, M. A. Saunders, and Y. Ye, "SPASELOC: An Adaptive Subproblem Algorithm for Scalable Wireless Sensor Network Localization," *SIAM J. Optimization*, vol. 17, pp. 1102–1128, Dec. 2006.

[35] P. Biswas, T.-C. Lian, T.-C. Wang, and Y. Ye, "Semidefinite Programming Based Algorithms for Sensor Network Localization," *ACM Trans. Sensor Networks*, vol. 2, pp. 188–220, May 2006.

[36] P. Tseng, "Second-Order Cone Programming Relaxation of Sensor Network Localization," *SIAM J. on Optimization*, vol. 18, pp. 156–185, Feb. 2007.

[37] J. Liu, Y. Zhang, and F. Zhao, "Robust Distributed Node Localization with Error Management," in *ACM Int. Symp. Mobile Ad Hoc Networking and Computing (MobiHoc)*, pp. 250–261, May 2006.

[38] H. Lim and J. C. Hou, "Localization for Anisotropic Sensor Networks," in *IEEE Infocom*, vol. 1, pp. 138–149, Mar. 2005.

[39] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, "Range-free Localization Schemes for Large Scale Sensor Networks," in *Proc. 9th Int. Conf. Mobile Computing and Networking (MobiCom)*, pp. 81–95, 2003.

[40] R. Stoleru, T. He, J. A. Stankovic, and D. Luebke, "A High-accuracy, Low-cost Localization System for Wireless Sensor Networks," in *Proc. 3rd Int. Conf. Embedded Networked Sensor Systems (SenSys)*, pp. 13–26, 2005.

[41] R. Stoleru, P. Vicaire, T. He, and J. A. Stankovic, "StarDust: A Flexible Architecture for Passive Localization in Wireless Sensor Networks," in *Proc. 4th Int. Conf. Embedded Networked Sensor Systems (SenSys)*, pp. 57–70, 2006.

[42] K. Römer, "The Lighthouse Location System for Smart Dust," in *Proc. 1st Int. Conf. Mobile Systems, Applications and Services (MobiSys)*, pp. 15–30, 2003.

[43] S. F. A. Shah and A. H. Tewfik, "Performance analysis of directional beacon based postion location algorithm for UWB systems," in *Proc. of IEEE GLOBECOM*, vol. 4, pp. 2409–2413, Nov. 2005.

[44] C. D. McGillem and T. S. Rappaport, "A beacon navigation method for autonomous vehicles," *IEEE Trans. Veh. Technol.*, pp. 132–139, Aug. 1989.

[45] A. Nasipuri and K. Li, "A Directionality based Location Discovery Scheme for Wireless Sensor Networks," in *ACM Symp. Wireless Sensor Network Applications*, pp. 105– 111, Sep. 2002.

[46] A. Nasipuri and R. el Najjar, "Experimental Evaluation of an Angle Based Indoor Localization System," in *Int. Symp. Modeling and Optimization Mobile, Ad Hoc and Wireless Networks*, Apr. 2006.

[47] H. Rowaihy, S. Eswaran, P. Johnson, T. Brown, A. Barnoy, D. Verma, and T. L. Porta, "A Survey of Sensor Selection Schemes in Wireless Sensor Networks," in *SPIE Unattended Ground, Sea, and Air Sensor Technologies and Applications IX*, May 2007.

[48] A. O. Hero, C. M. Kreucher, and D. Blatt, "Information Theoretic Approaches to Sensor Management," in *Foundations and Applications of Sensor Management*, pp. 33–57, Oct. 2007.

[49] J. Liu, J. Reich, and F. Zhao, "Collaborative In-network Processing for Target Tracking," *EURASIP J. Appl. Signal Process.*, no. 1, pp. 378–391, 2003.

[50] E. Ertin, J. W. Fisher, and L. C. Potter, "Maximum Mutual Information Principle for Dynamic Sensor Query Problems," in *Proc. IPSN 03*, pp. 405–416, 2003.

[51] H. Wang, K. Yao, G. Pottie, and D. Estrin, "Entropy-based sensor selection heuristic for target localization," in *Proc. IPSN 04*, pp. 36–45, 2004.

[52] J. L. Williams, "Information Theoretic Sensor Management." PhD Thesis, Massachusetts Institute of Technology, Feb. 2007.

[53] J. L. Williams, J. W. Fisher, and A. S. Willsky, "Performance Guarantees for Information Theoretic Sensor Resource Management," in *Proc. ICASSP 07*, pp. III933–III936, Apr. 2007.

[54] S. Han, E. Chan, R. Cheng, and K.-Y. Lam, "A statistics-based sensor selection scheme for continuous probabilistic queries in sensor networks," in *Proc. IEEE RTCSA*, pp. 331–336, Aug. 2005.

[55] A. Urrela, E. Ström, and M. Pydström, "Novel Wireless Location Approach for W-CDMA Systems based on Multiple Sliding Correlators," in *Proc. ICASSP 2005*, pp. 945–948, Mar. 2005.

[56] K. Römer, "Time synchronization in Ad-hoc Networks," in *Proc. 2nd ACM Int. Symp. Mobile Ad Hoc Networking & Computing*, pp. 173–182, 2001.

[57] E. Saberinia and A. H. Tewfik, "Enhanced Localization in Wireless Personal Area Networks," in *Proc. IEEE GLOBECOM*, vol. 4, pp. 2429–2434, Nov. 2004.

[58] M. Pendergrass, "Empirically based Statistical Ultra Wideband Channel Model." IEEE P802.15-02/240-SG3a, 2002.

[59] IEEE 802.15 WG, "Channel modeling sub-committee final report." IEEE P802.15 Working Group for Wireless Personal Area Networks (WPANs), Feb. 2003.

[60] A. M. Saleh and R. A. Valenzuela, "A Statistical Model for Indoor Multipath Propagation," *IEEE J. Select. Areas Commun.*, vol. 5, pp. 128–137, Feb. 1987.

[61] M. Z. Win and R. A. Scholtz, "Characterization of Ultra-Wide Bandwidth Wireless Indoor Channels: A Communication-Theoretic View," *IEEE J. Select. Areas Commun.*, vol. 20, pp. 1613–1627, Dec. 2002.

[62] S. S. Ghassemzadeh, L. J. Greenstein, T. Sveinsson, A. Kavcic, and V. Tarokh, "UWB Delay Profile Models for Residential and Commercial Indoor Environments," *IEEE Trans. Veh. Technol.*, vol. 54, pp. 1235–1244, Jul. 2005.

[63] P. C. Chen, "A Non-Line-of-Sight Error Mitigation Algorithm in Location Estimation," in *IEEE Wireless Commun. and Netw. Conf.*, vol. 1, pp. 316–320, Sep. 1999.

[64] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[65] Y. Ye, "Semidefinite Programming for Sensor Network and Graph Localization," in *Nonconvex Optimization and Its Applications: Robust Optimization-Directed Design*, pp. 247–275, Jun. 2006.

[66] D. Bertsekas and J. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[67] J. Strum, "Using SeDuMi 1.02, A Matlab Toolbox for Optimization Over Symmetric Cones (Updated for Version 1.05)," Oct. 2001.

[68] Ettus Research LLC, "http://www.ettus.com."

[69] Telex Inc., "http://www.telex.com."

[70] Lin Engineering, "http://www.linengineering.com/."

[71] GNU Radio - The GNU Software Radio, "http://www.gnu.org/software/gnuradio/."

[72] S. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Englewood Cliffs, NJ: Prentice Hall, 1993.

[73] D. W. Tufts and R. L. Field, "Least-squares Time-delay Estimation for Transient Signals in a Multipath Environment," *J. Acoust. Soc. Amer*, vol. 92, pp. 210–218, Jul. 1992.

[74] T. G. Manickam, R. J. Vaccaro, and D. W. Tufts, "A Least-squares Algorithm for Multipath Time-delay Estimation," *IEEE Trans. Signal Process.*, vol. 42, pp. 3229–3233, Nov. 1994.

[75] L. Yu and A. Ephremides, "Detection Performance and Energy Efficiency of Sequential Detection in a Sensor Network," in *Proc. HICSS 06*, p. 236a, 2006.

[76] A. Wald, *Sequential Analysis*. New York: Wiley, 1947.

[77] H. V. Poor, *An Introduction to Signal Detection and Estimation*. Springer-Verlag, 1988.

[78] M. Chiani, D. Dardari, and M. K. Simon, "New Exponential Bounds and Approximations for the Computation of Error Probability in Fading Channels," *IEEE Trans. on Wireless Commun.*, vol. 2, pp. 840–845, Jul. 2003.

[79] R. Chandramouli and N. D. Memon, "On Sequential Watermark Detection," *IEEE Trans. Signal Process.*, vol. 51, pp. 1034–1044, Apr. 2003.

[80] P. Armitage, "Sequential analysis with more than two alternative hypotheses, and its relation to discriminant function analysis," *J. Roy. Stat. Soc.*, vol. 12, no. Series C, pp. 137–144, 1950.

[81] I. Jouny and F. D. Garber, "M-ary sequential hypothesis tests for automatic target recognition," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 28, pp. 473–483, Apr. 1992.