Cooperative Positioning in Wireless Sensor Networks Using Semidefinite Programming

Sayed Reza Monir Vaghefi

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Electrical Engineering

R. Michael Buehrer, Chair
Jeffrey H. Reed
Allen B. MacKenzie
J. Michael Ruohoniemi
Anne G. Ryan

December 08, 2014
Blacksburg, Virginia

Keywords: Cooperative, localization, semidefinite programming, convex optimization

Cooperative Positioning in Wireless Sensor Networks Using Semidefinite Programming

Sayed Reza Monir Vaghefi

Abstract

With the rapid development of wireless technologies, the demand for positioning services has grown dramatically over the past three decades. The Global Positioning System (GPS) is widely used in wireless devices for positioning purposes. However, in addition to having bulky and expensive equipment, GPS receivers do not operate properly in dense and indoor environments. Difficulties in using GPS lead us to use sensor localization in which the position information is obtained from the measurements collected within the network without the aid of external resources. Sensor localization has been a great topic of interest during past decades. Although many positioning algorithms have been developed previously in the literature, positioning is still a challenging task. There are many factors that can affect the positioning performance if they are neglected or not treated properly. These factors introduce many nuisance parameters which need to be either estimated or considered when the location is estimated.

In this work, we exploit cooperative localization as a recent and trending technology and semidefinite programming (SDP) as a powerful tool in our research. Cooperative localization has several advantages over the traditional noncooperative localization in terms of positioning accuracy and localizability. Cooperation is also highly beneficial for networks with few anchor nodes and low communication range. On the other hand, SDP provides an alternative solution to the optimal maximum-likelihood (ML) estimation. Unlike in the ML estimator, convergence to the global minimum is guaranteed in SDP. It also has significantly lower complexity especially for cooperative networks in exchange for small performance degradation. Using these two concepts, four open problems within the area of cooperative localization and tracking in the presence of nuisance parameters are addressed. In particular, we focus on cooperative received signal strength-based localization when the propagation parameters including path-loss exponent and transmit powers are unknown. Cooperative time-of-arrival-based localization in harsh environments in the presence of severe non-line-of-sight (NLOS) propagation is also investigated. Cooperative localization in asynchronous networks is studied where the clock parameters are considered as nuisance parameters and the focus is on a joint synchronization and localization approach. Lastly, source tracking in NLOS environments is studied where source nodes are mobile and their status changes rapidly from LOS to NLOS and vice versa.

Dedication

To my wife and to my parents.

Acknowledgments

This work would not have been possible without the support of many individuals. Therefore, I would like to take this opportunity to thank them all.

First and foremost, I would like to thank my advisor, Dr. R. Michael Buehrer, for his guidance, support, and commitment. I am also grateful to him for giving me freedom to explore my ideas and curiosities and always believing in me. I have learned a lot from his unique blend of knowledge and personality. I also would like to express my appreciations to my committee members for their valuable comments during the course of this research.

I extend great gratitude towards my colleagues and friends in the Mobile and Portable Radio Research Group (MPRG) for providing an excellent and friendly environment.

I would like to particularly thank my wife and my parents for believing in me, and for their encouragement and support.

Table of Contents

List of Figures

List of Tables

Abbreviations

AOA	Angle-of-arrival
CDF	Cumulative density function
CRLB	Cramér-Rao lower bound
EKF	Extended Kalman filter
GPS	Global Positioning System
LLS	Linear least squares
LOS	Line-of-sight
LP	Linear programming
MAP	Maximum a posteriori
ML	Maximum liklihood
MSE	Mean square error
NLOS	Non-line-of-sight
PDF	Probability density function
RMSE	Root mean square error
RSS	Received-signal-strength
SDP	Semidefinite programming
SNR	Signal-to-noise-ratio
TDOA	Time-difference-of-arrival
TOA	Time-of-arrival
WSN	Wireless sensor network

Chapter 1

Introduction

Wireless communications has grown rapidly over the past two decades. Nowadays, wireless devices such as cell phones, tablets, and laptops can be found everywhere. Along with the rapid development of wireless technologies, the demand for positioning services has increased as well. Positioning has many civilian, commercial, and military applications such as finding the location of an E-911 caller, locating an aircraft, locating sensors in a wireless sensor network (WSN), or tracking firefighters inside a burning building. The Global Positioning System (GPS) is the primary and the most accessible navigation technique for most applications. However, there are several drawbacks associated with satellite-based positioning techniques such as GPS. First, in many commercial and military applications, low-cost and low-power wireless devices are required. Hence, equipping each sensor in the network with a GPS receiver would be expensive. Moreover, sensors may be too small to carry a GPS receiver. Second and more importantly, GPS receivers do not work properly in indoor and dense environments (such as inside buildings, tunnels, or a forest) where the satellite signals are severely attenuated. Therefore, having a simple and efficient sensor localization technique is required. The main aim of sensor localization is to determine the locations of unknown nodes (called source nodes) by using measurements obtained within the network without the aid of external resources such as GPS. Sensor localization includes a series of anchor nodes with known locations which try to estimate the locations of source nodes whose locations are unknown. The anchor nodes might be equipped with a GPS receiver or have been placed at known positions. Although using this approach will address the cost and space limitation of GPS, there are still many problems which must be addressed to have a reliable positioning performance.

Generally in wireless networks, there are many nodes with unknown locations to be localized, perhaps substantially more than the number of anchor nodes. However, most studies on sensor localization focus on single-node positioning in which they are dealing only with a single unknown source node at the time. In traditional approaches, the network is broken down into several subnetworks with a single unknown source node and source nodes are treated individually. The examples of such networks are cellular networks and GPS networks. However, traditional approaches are not useful in low-power networks with limited connectivity. The performance of such networks rely on relaying due to cost and energy limitations. Therefore, these networks cannot be easily broken

down into subnetworks, because all source node may not see a sufficient number or any anchor nodes. In this case, source nodes cannot be treated individually and the network as a whole should be examined where the relationship between all nodes is taken into account.

At first sight, positioning seems to be trivial where a simple trilateration process is used. However, positioning is a challenging task in practice. There are many factors which can impact positioning performance such as multi-path and non-line-of-sight (NLOS) propagation, interference, limited connectivity, poor geometry of anchor nodes, synchronization, and uncertainties in the assumed model. These factors result in many nuisance parameters that must be estimated or otherwise considered when estimating the location. In estimation theory, a nuisance parameter is referred to as any parameter that is not of interest, but must be estimated (or considered) for analyzing the parameters of interest. For instance, Although localization in general has a rich history in the literature, *cooperative* localization is relatively recent and there are still several open problems in this area. Cooperative localization is an emerging technology for dealing with multiple-node networks which has received substantial interest in the area of wireless communications, optimization, and robotics. Unlike traditional noncooperative localization which is used for single-node networks, cooperative localization utilizes measurements collected between unlocalized nodes in addition to the measurements collected by the anchor nodes (i.e., nodes with known locations). Cooperative localization in general can significantly improve localization accuracy and localizability. Localizability refers to a condition where a source node can be localized uniquely in a network. In many networks, source nodes cannot be localized uniquely using traditional noncooperative localization, while their locations can be determined uniquely if cooperation is taken into account. Cooperative localization is also highly beneficial in indoor and dense environments where additional measurements can be used to mitigate NLOS propagation. Although cooperative localization is typically applied to wireless sensor networks, it can be applied to nearly any wireless network, e.g., a cellular network.

Nuisance parameters have been typically neglected in the previous studies of cooperative localization. However, these parameters significantly affect the positioning performance, if they are not treated properly. In this work, we examine the impact of nuisance parameters within the area of network localization and tracking. We have used cooperative localization as a trending technology in wireless communications to improve localization performance and also semidefinite programming (SDP) as a promising and recent optimization tool to deal with the problem of nuisance parameters.

Localization using received signal strength (RSS) measurements is highly popular for wireless sensors, mainly due to simple and inexpensive implementation. However, RSS-based localization requires strong knowledge of the propagation parameters such as the transmit powers of the source nodes and the path loss exponent of the environment. In most of the previous studies on RSS-based localization, perfect knowledge about propagation parameters is assumed which is impractical. Moreover, these studies mainly limit themselves to noncooperative networks. In this work, we assume that the propagation parameters are unknown. Considering the propagation parameters as nuisance parameters in the estimation problem, we develop a cooperative RSS-based localization

technique.

One of the most significant challenges in ranged-based localization techniques such as time-of-arrival (TOA) and time-difference-of-arrival (TDOA) is NLOS propagation. Sensor localization is more beneficial in indoor and dense environments where GPS receivers cannot operate properly. However, in such environments most connections are NLOS due to the presence of many objects and walls. NLOS propagation degrades localization performance, if not treated properly. Although many studies have been done on mitigation of NLOS propagation for noncooperative networks, relatively few studies have focused on cooperative networks and we believe that there is still room for improvement. Therefore, this work concentrates on cooperative range-based localization in NLOS environments where not only do the anchor nodes collect measurements but also the source nodes are involved in collecting measurements from each other. Cooperative localization is highly beneficial in severe NLOS environments, as the effect of NLOS propagation can be mitigated by providing additional measurements to the estimator.

The performance of range-based localization techniques is also highly dependent on accurate time synchronization between source nodes and anchor nodes. A small error in time synchronization causes significant localization errors. Although it is reasonable to assume that the anchor nodes are synchronized, the source nodes are not typically synchronized in practice. Generally in asynchronous networks, the source nodes are first synchronized with the anchor nodes and then localization is performed. However, synchronization and localization can be performed jointly, as there is a close relationship between them. Previous studies that have considered the impact of synchronization mainly concentrate on noncooperative networks where dense anchor node placement is required to ensure that source nodes have a sufficient number of connections. We address the problem of joint synchronization and localization for the first time for cooperative networks. We will show that the localization performance and localizability can be improved significantly using cooperation.

Another problem which has not gained enough attention in the literature is source tracking in cooperative networks. The difference between static localization and tracking is that in the latter source nodes are mobile and an underlying dynamic model is used to localize source nodes over time. While there have been many studies on source tracking in noncooperative networks, no study focuses on the tracking performance of mobile nodes in cooperative networks, especially when we have NLOS propagation. Therefore in this work, for the first time, the tracking performance of cooperative networks in severe NLOS environments is examined.

1.1 Problem Statement

In this research, we wish to address four open problems in network localization and tracking, in all of which the impact of nuisance parameters is considered. More specifically, the following problems are addressed in this work:

1. Network localization based on RSS measurements when transmit powers of the source node

and the path loss exponent are unknown and when shadowing is potentially correlated.

2. Network localization and synchronization based on TOA measurements when node timing is unknown (i.e., asynchronous networks).

3. Network localization based on TOA measurements in presence of severe NLOS propagation.

4. Network position tracking of mobile nodes based on TOA measurements in presence of NLOS propagation.

1.2 Contributions

This work is primarily based on two important concepts: cooperative localization and semidefinite programming.

Cooperative Localization: Localization in general can be divided into two groups: noncooperative and cooperative. In conventional noncooperative localization, the unlocalized wireless devices only communicate with infrastructure (i.e., anchor nodes). Examples of such localization techniques are GPS and cellular-based localization. In satellite-based navigation systems such as GPS, the receivers only collect measurements from the visible satellites. Also in cellular systems, cell phones are only able to communicate with the base stations. Generally speaking, in noncooperative networks, each source node with unknown location should be connected to at least 3 anchor nodes to be localized uniquely. This requirement is usually referred to as node localizability. In this case, a large number of anchor nodes is required to ensure that the source nodes are connected to a sufficient number of anchor nodes which significantly increases the implementation cost. On the other hand, energy limitations and interference consideration do not allow for longer communication range. In indoor and dense environments, even the presence of many anchor nodes does not improve connectivity in the network as most connections are unreliable due to NLOS propagation. Such limitations associated with noncooperative localization leads to the emergence of cooperative localization in which the source nodes communicate not only with the anchor nodes but also with each other. Cooperative localization provides several advantages over noncooperative localization. First and foremost, cooperation can significantly improve the localization accuracy, as more measurements are available to the estimator. The need for dense anchor node deployment or longer communication range is also eliminated, since source nodes will have additional connections through other source nodes. Lastly, localizability is also increased through collaboration. Because of these reasons, cooperative localization comprises a significant part of this work.

Semidefinite Programming: The main and most important tool used in this research is SDP. SDP is a form of convex optimization which includes minimization (or maximization) of a linear cost function constrained by the intersection of the cones defined by positive semidefinite matrices. SDP is relatively recent in the area of optimization and has been a subject of interest during the past decade mainly because of its several interesting features. Now the most important question that comes to mind is why have we used SDP in this work and why not other algorithms or approaches.

In localization, a nonlinear and nonconvex optimization problem needs to be solved to obtain the optimal maximum likelihood (ML) estimate. The major drawback of nonlinear and noncovex problems is that they do not have a closed-from solution. Such problems can be approximately solved by iterative algorithms such as the Gauss-Newton algorithm which needs an initial solution. Since the cost function of the ML estimator is noncovex, it will have several local minima and saddle points. Therefore, a good initial point is required to ensure that the algorithm converges to the global minimum. Moreover, the complexity of such algorithms grows considerably in cooperative networks with many unlocalized nodes. One way to deal with a nonlinear and convex problem is to linearize the model and then formulate it as a linear problem such as linear least squares or linear programming. Although their complexity is substantially lower than their nonlinear counterparts, their performance is not as good as the general nonlinear problems. Moreover, linear estimators cannot be easily developed for cooperative networks, especially with limited connectivity. Relaxing the nonlinear and noncovex problem into a nonlinear but convex problem provides another solution for the ML estimator. Convex optimization problems such as SDP have a convex cost function in which convergence to the global minimum is guaranteed. Although there are several types of convex optimization problems such quadratic programming and second order cone programming, SDP is the most general form of them which can be solved effectively in polynomial time.

Using the potential of cooperative localization and also SDP as a powerful tool, we address several open problems in the area of localization. The contributions of this research are summarized as follows.

1.2.1 Cooperative RSS-Based Localization with Unknown Transmit Powers

Cooperative localization (also known as sensor network localization) using RSS measurements when the transmit powers of the source nodes are different and unknown is investigated in this contribution. Previous studies are based on the assumption that the transmit powers of source nodes are the same and perfectly known which is not practical. In this work, the transmit powers of the source nodes are considered as nuisance parameters and estimated along with the locations of the source nodes. The corresponding Cramér-Rao Lower Bound (CRLB) and the maximum likelihood (ML) estimator are derived. To determine the ML estimate, it is necessary to solve a nonlinear and nonconvex problem, which is shown to be computationally complex. To avoid the difficulty in solving the ML estimator, we derive a novel SDP relaxation technique by converting the ML minimization problem into a convex problem which can be solved efficiently. The algorithm requires only an estimate of the path-loss exponent (PLE). We initially assume that perfect knowledge of the PLE is available, but we then examine the effect of imperfect knowledge of the PLE on the proposed SDP algorithm and also develop an algorithm to estimate it. The complexity analyses of the proposed algorithms are also provided in detail. Computer simulations shows that the proposed SDP estimator performs as well as the optimal ML estimator with significantly lower complexity in cooperative networks.

1.2.2 Cooperative TOA-Based Localization in NLOS Environments

TOA-based sensor localization in NLOS environments is investigated in this contribution. In the first part of this work, we assume that the estimator is able to discriminate NLOS connections from LOS connections. We introduce a novel SDP algorithm for cooperative localization which exploits NLOS connections to enhance the accuracy of localization. The performance of the proposed algorithm is compared with that of the ML estimator and previously considered algorithms. In the second part, we assume that the estimator knows neither which connections are NLOS nor the distribution of the NLOS errors. It is shown that the ML estimator using only LOS connections provides a lower bound on the estimation accuracy. Furthermore, a novel NLOS mitigation technique based on SDP is proposed. The proposed SDP technique estimates the locations of source nodes jointly with the NLOS biases as nuisance parameters. The performance of the proposed SDP estimator is compared with the aforementioned lower bound and with the previous algorithms through computer simulations. Simulation results show that the proposed SDP estimator outperforms the other algorithms substantially, especially in severe NLOS environments.

1.2.3 Cooperative Joint Synchronization and Localization

Cooperative sensor localization using asynchronous TOA measurements is investigated in this contribution. It is well known that localization performance in wireless networks using time-based ranging or pseudo-ranging methods is severely affected by the precision of the timing synchronization between the nodes involved in the estimation. Commonly, the original estimation problem is broken down into two subproblems, the synchronization problem and the localization problem, in what is known as a two-step approach. However in this work, clock parameters are considered as nuisance variables and a joint synchronization and localization technique is proposed for use in cooperative networks. This is the first time that this problem has been addressed for cooperative networks. It will be shown that the cooperation between the source nodes eliminates the need for high anchor node densities and improves localization performance significantly. Moreover, the CRLB and the ML estimator are derived for this problem. It will be shown that the ML estimator is highly nonlinear and nonconvex and must, therefore, be solved by using computationally complex algorithms. To reduce the estimation complexity, a novel SDP estimator is formulated by relaxing the original nonconvex ML problem into a convex problem. The performance of the proposed SDP estimator is shown through computer simulations to be nearly equal to that of the ML estimator. The approach was also applied to noncooperative networks where it is found to be superior in performance than the previously proposed estimators. Finally, complexity analyses are included for the estimators under consideration.

1.2.4 Cooperative Source Node Tracking

Cooperative source node tracking in NLOS environments is examined in this contribution. This is the first study which addresses tracking for cooperative localization. Although many studies

have examined NLOS-degraded localization of a static node in noncooperative networks, and many others have examined the impact of cooperation for static localization, there is no work which considers cooperative tracking of mobile nodes, especially in NLOS environments. To address this open problem, in this work we examine cooperative tracking, particularly in NLOS environments. More specifically, we develop an extended Kalman filter (ELF) for use in cooperative networks. We also develop a novel sensor tracking algorithm based on SDP which has the ability to mitigate NLOS propagation. Our simulations show that the proposed SDP estimator outperforms the classic EKF as well as the other recently proposed estimators for noncooperative tracking in NLOS environments. We also show that the algorithm can be extended to cooperative networks, and that a substantial performance benefit is realized by cooperation.

1.2.5 List of Publications

The work in this dissertation is based on the following papers:

1. R. M. Vaghefi and R. M. Buehrer, "Cooperative sensor localization with NLOS mitigation using semidefinite programming," *in Proc. IEEE WPNC*, pp. 13-18, Mar. 2012.

2. R. M. Vaghefi, J. Schloemann, and R. M. Buehrer, "NLOS mitigation in TOA-based localization using semidefinite programming," *in Proc. IEEE WPNC*, pp. 1-6, Mar. 2013.

3. R. M. Vaghefi, M. R. Gholami, R. M. Buehrer, and E. G. Strom, "Cooperative received signal strength-based sensor localization with unknown transmit powers," IEEE Transactions on Signal Processing, vol. 61, no. 6, pp. 1389-1403, 2013.

4. R. M. Vaghefi and R. M. Buehrer, "Received signal strength-based sensor localization in spatially correlated shadowing," *in Proc. IEEE ICASSP*, pp. 4076-4080, May 2013.

5. R. M. Vaghefi and R. M. Buehrer, "Asynchronous time-of-arrival-based source localization," *in Proc. IEEE ICASSP*, pp. 4086-4090, May 2013.

6. R. M. Vaghefi and R. M. Buehrer, "Target tracking in NLOS environments using semidefinite programming," *in Proc. IEEE MILCOM*, pp. 169-174, Nov. 2013.

7. R. M. Vaghefi and R. M. Buehrer, "Improving positioning in LTE through collaboration," *in Proc. IEEE WPNC*, pp. 1-6, Mar. 2014.

8. R. M. Vaghefi and R. M. Buehrer, "Joint TOA-based sensor synchronization and localization using semidefinite programming," *in Proc. IEEE ICC 2014*, pp. 520-525, Jun. 2014.

9. R. M. Vaghefi and R. M. Buehrer, "Cooperative RF pattern matching positioning for LTE cellular systems," *in Proc. IEEE PIMRC*, Sept. 2014.

10. R. M. Vaghefi and R. M. Buehrer, "A linear estimator for joint synchronization and localization in wireless sensor networks," *in Proc. IEEE GLOBECOM*, Dec. 2014.

11. R. M. Vaghefi, S. Amuru, and R. M. Buehrer, "Improving mobile node tracking performance in NLOS environments using cooperation," *in Proc. IEEE ICC*, 2015.

12. R. M. Vaghefi and R. M. Buehrer, "Cooperative joint synchronization and localization in wireless sensor networks," Submitted to IEEE Transactions on Signal Processing.

13. R. M. Vaghefi and R. M. Buehrer, "Cooperative source node tracking in non-line-of-sight environments," Submitted to IEEE Transactions on Mobile Computing.

14. R. M. Vaghefi, J. Overfield, and R. M. Buehrer, "Impact of MIMO transmission on cross ambiguity function-based positioning," Submitted to IEEE Transactions on Signal Processing.

Chapter 2

Background

In this chapter, a brief background about the main topic of this work is provided. This chapter helps the reader to better understand the topic of this research and the content of the following chapters. First, different measurement methods which are typically used for sensor localization are described. Then, the CRLB of the TOA-based localization and its corresponding ML estimation are introduced. Afterwards, an introduction about SDP and its features are provided. Lastly, we talk about cooperative localization and its advantages over traditional noncooperative localization.

2.1 Measurement Models

Generally in a WSN, the positions of a number of sensors are known (anchor nodes), while there are some sensors (source nodes) whose positions are unknown and thus must be estimated using sensor localization. The main purpose of sensor localization is to determine the location of sensors in a WSN via noisy measurements [1]. These measurements may include received signal strength (RSS) [2, 3, 4], time-of-arrival [5], time-difference-of-arrival [6, 7, 8], and angle-of-arrival [9]. Suppose there is a network with M anchor nodes and one source node. Let $\mathbf{x} = [x, y]^{\mathrm{T}} \in \mathbb{R}^2$ be the unknown coordinates to be estimated and $\mathbf{y}_i = [x_i, y_i]^{\mathrm{T}} \in \mathbb{R}^2$, $i = 1, \ldots, M$ be the known coordinates of the ith anchor node.

The TOA measurement is considered as the travel time of signal between the source node and anchor node. The TOA measured at the ith anchor is modeled as [1],

$$t_i = \frac{d_i}{c} + w_i \tag{2.1}$$

where c is the propagation velocity (e.g., in vacuum electro-magnetic propagation speed is $c \approx 3x10^8$ m/s), w_i is the measurement noise modeled as zero-mean Gaussian random variable with variance σ_i^2.

In RSS method, the amount of signal attenuation from the source node to the anchor node is measured. The average measured power received at the node is defined as RSS. Under log-normal

shadowing, the received power (in dBm) at the ith anchor node is modeled as [2]

$$P_i = P_0 - 10\beta \log_{10} \frac{d_i}{d_0} + n_i \tag{2.2}$$

where P_0 (in dBm) is the reference power at distance d_0 from the source node. β is the path loss exponent which varies typically between 2 and 4 depending on the propagation environment [1]. d_i is the true distance between the source node and the ith anchor node

$$d_i = \|\mathbf{x} - \mathbf{y}_i\|_2 \tag{2.3}$$

where $\|\cdot\|_2$ denotes the ℓ_2 norm. n_i are the log-normal shadowing terms. The standard deviation of shadowing is reported in dB unit and is topically between 4 and 12 dB [1].

The angle between the source node and anchor node is measured in AOA technique. The hardware for measuring AOA is relatively more complex than other methods which makes it less popular for sensor localization. The AOA measurement is modeled as [1]

$$\alpha_i = \tan^{-1} \frac{y - y_i}{x - x_i} + n_i \tag{2.4}$$

where $\tan^{-1}\{\cdot\}$ is four-quadrant inverse tangent, and n_i is the measurement noise modeled as zero-mean Gaussian random variable with variance σ_α^2. The experimental results show that the standard deviation of error varies between 2 and 6 degrees in an acoustic-based AOA method, depending on the inter-node distance.

2.2 Cramér-Rao Lower Bound

The CRLB provides a lower bound on the performance (variance) of any unbiased estimator [10, Ch. 3]. The CRLB is used as a benchmark to compare the performance of different estimators. In other words, it would indicate how accurate an estimator is and how far its performance is from the lower bound. An estimator is asymptotically optimal if its variance reaches the CRLB as the number of measurements increases. The CRLB of the unknown parameter, $\mathbf{x}$, is obtained from the diagonal elements of the inverse of the Fisher information matrix [10, Ch. 3]. The elements of the Fisher information matrix are obtained as follows:

$$[\mathbf{I}(\boldsymbol{\theta})]_{mn} = -\mathrm{E}\left[\frac{\partial^2 \log p\left(\mathbf{r};\boldsymbol{\theta}\right)}{\partial\theta_m \partial\theta_n}\right]. \tag{2.5}$$

where $p\left(\mathbf{r}|\boldsymbol{\theta}\right)$ is the likelihood function of the measurements.

In a TOA-based method, the signal travel time from the source node to the anchor nodes is measured. For simplicity, all obtained measurements are multiplied by the propagation velocity to obtain distance information as

$$r_i = ct_i = d_i + n_i \tag{2.6}$$

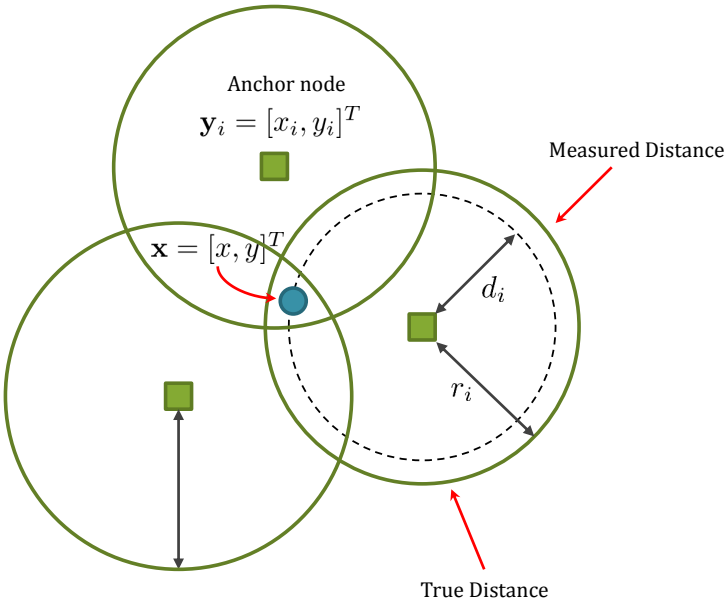


Figure 2.1: TOA-based Sensor localization. The squares represent the anchor nodes and the sold circle represents the source node. In TOA method, the distance between the source and anchor nodes are measured and used to estimate the location of source node.

where r_i is the measured distance obtained from TOA measurements, as shown in Fig. 2.1. $n_i = cw_i$ is the measurement error. Since the accuracy of TOA measurements is related to the received signal-to-noise ratio (SNR) [1, 11] which itself is related to the distance and the path-loss exponent [12, 2], the variance of n_i is typically modeled as [13]

$$\sigma_i^2 = \alpha d_i^{\beta_i} \tag{2.7}$$

where α is a constant which defines the relationship between the noise variance and the true distance and depends on the network environment and hardware implementation of nodes and β_i is the path loss exponent which typically varies between 2 and 4. The probability density function (pdf) of the model in (2.6) is written as

$$p(\mathbf{r}|\mathbf{x}) = \frac{1}{(2\pi)^{M/2}|\mathbf{Q}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{r} - \mathbf{d}(\mathbf{x}))^{\mathrm{T}}\mathbf{Q}^{-1}(\mathbf{r} - \mathbf{d}(\mathbf{x}))\right) \tag{2.8}$$

where $\mathbf{r} = [r_1, r_2, \ldots, r_M]^{\mathrm{T}}$, $\mathbf{d}(\mathbf{x}) = [d_1, d_2, \ldots, d_M]^{\mathrm{T}}$, $\mathbf{Q} = \mathrm{E}\{\mathbf{n}\mathbf{n}^{\mathrm{T}}\}$, and $\mathbf{n} = [n_1, n_2, \ldots, n_M]^{\mathrm{T}}$. Now, let us derive the CRLB of the known parameter $\mathbf{x}$ for the model in (2.6). To determine the CRLB, we need to calculate the Fisher information matrix. After some calculation and manipulations, the Fisher information matrix of the model in (2.6) is calculated as [10, Ch. 3]

$$\mathbf{I}(\mathbf{x}) = \mathbf{H}(\mathbf{x})^{\mathrm{T}}\mathbf{Q}^{-1}\mathbf{H}(\mathbf{x}) \tag{2.9}$$

where $\mathbf{H}(\mathbf{x}) \in \mathbb{R}^{M\times 2}$ is the Jacobian matrix obtained as

$$\mathbf{H}(\mathbf{x}) = \frac{\partial \mathbf{d}(\mathbf{x})}{\partial \mathbf{x}} = \begin{bmatrix} \dfrac{(\mathbf{x}-\mathbf{y}_1)^{\mathrm{T}}}{\|\mathbf{x}-\mathbf{y}_1\|_2} \\ \vdots \\ \dfrac{(\mathbf{x}-\mathbf{y}_M)^{\mathrm{T}}}{\|\mathbf{x}-\mathbf{y}_M\|_2} \end{bmatrix}. \tag{2.10}$$

Finally, the CRLB of the unknown parameter $\mathbf{x}$ is computed as

$$\mathrm{Var}([\mathbf{x}]_i) \geq \left[\mathbf{I}^{-1}(\mathbf{x})\right]_{i,i}, \quad i = 1, 2. \tag{2.11}$$

2.3 Maximum Likelihood

ML estimation is a popular method of estimating unknown parameters of a statistical model. The ML estimator has several attractive limiting properties (when the number of measurements tends to infinity) as follows:

- **Asymptotically optimal**: as the number of measurements increases, it can reach the CRLB accuracy.
- **Asymptotic normality**: as the measurement size increases, the distribution of the ML estimator reaches to a Gaussian distribution with mean of $\mathbf{x}$ (asymptotically unbiased) and the covariance matrix equal to the inverse of the Fisher information matrix (FIM).
- **Consistency**: as the number of measurements increases, the ML estimator converges to the true parameter.

The ML estimator is obtained by maximizing the likelihood function of the measurement model. Since logarithm is a monotonically increasing function, maximizing the log-likelihood function (the logarithm of the likelihood function) and the likelihood function yield the same estimate. The log-likelihood function is then given by

$$\log p(\mathbf{r}|\mathbf{x}) = K - \frac{1}{2}(\mathbf{r}-\mathbf{d}(\mathbf{x}))^{\mathrm{T}}\mathbf{Q}^{-1}(\mathbf{r}-\mathbf{d}(\mathbf{x})) \tag{2.12}$$

where K is a constant which is independent of the unknown parameter $\mathbf{x}$. Maximizing (2.12) is equal to minimizing the negative of (2.12). Therefore, the ML estimator is simply calculated by the following nonlinear optimization problem [10, Ch. 7]

$$\hat{\mathbf{x}} = \underset{\mathbf{x}\in\mathbb{R}^2}{\arg\min}\, (\mathbf{r}-\mathbf{d}(\mathbf{x}))^{\mathrm{T}}\mathbf{Q}^{-1}(\mathbf{r}-\mathbf{d}(\mathbf{x})). \tag{2.13}$$

The cost function of the ML estimator is highly nonlinear and nonconvex. Moreover, a closed-form solution for the ML estimator given in (2.13) cannot be found. Fig. 2.2 shows one realization

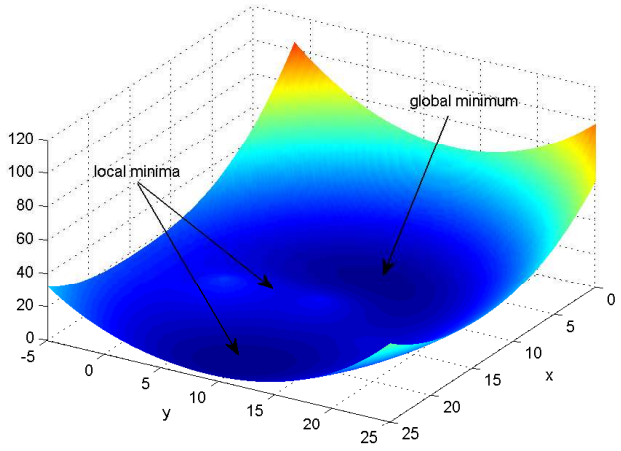


Figure 2.2: One realization of the ML cost function. The cost function of the ML estimator is highly nonlinear and noncovex. It has several local minima and saddle points.

of the ML cost function as a function of x and y coordinates in $\mathbf{x} = [x, y]^{\mathrm{T}}$. The cost function is nonconvex and there are several stationary points (local minima or saddle points). However, we are only interested in the global minimum which is the solution of the problem in (2.13). In this case, numerical methods such as grid-search and iterative optimization algorithms can be employed.

In grid search methods, the cost function is evaluated over a 2-D (or 3-D) dimensional variable and the grid point that has the least value of the cost function is selected as a solution. Accuracy of this approach depends on the granularity of the grid. For instance in Fig. 2.2, the step size 0.2 was selected for both coordinates. This approach is very time consuming, especially when we need to search over a wide range of unknown parameters.

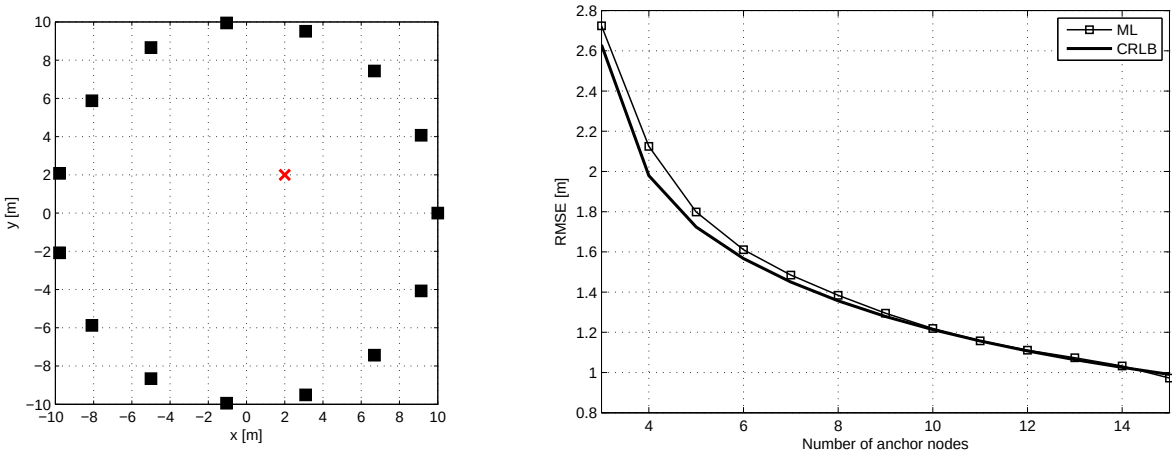
Alternatively, iterative optimization algorithms such as the Gauss-Newton algorithm can be used to solve nonlinear problems. In the Gauss-Newton method, the nonlinear cost function of the ML estimator is approximately linearized by using a first-order Taylor series and solved iteratively [10, Ch. 8]. Applying the Gauss-Newton method to our problem in (2.13) yields [14, 10]

$$\hat{\mathbf{x}}^{k+1} = \hat{\mathbf{x}}^{k} + \left(\mathbf{H}_k^{\mathrm{T}}\mathbf{Q}^{-1}\mathbf{H}_k\right)^{-1}\mathbf{H}_k^{\mathrm{T}}\mathbf{Q}^{-1}(\mathbf{r} - \mathbf{h}(\hat{\mathbf{x}}^{k})), \tag{2.14}$$

where $\mathbf{h}(\mathbf{x}) = [h_1(\mathbf{x}), h_2(\mathbf{x}), \ldots, h_M(\mathbf{x})]^{\mathrm{T}}$, $h_i(\mathbf{x}) = \|\mathbf{x} - \mathbf{y}_i\|_2$, $\mathbf{H}_k = \mathbf{H}(\mathbf{x})|_{\mathbf{x}=\mathbf{x}^k}$ and

$$\mathbf{H}(\mathbf{x}) = \frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}} = \begin{bmatrix} \dfrac{(\mathbf{x} - \mathbf{y}_1)^{\mathrm{T}}}{\|\mathbf{x} - \mathbf{y}_1\|_2} \\ \vdots \\ \dfrac{(\mathbf{x} - \mathbf{y}_M)^{\mathrm{T}}}{\|\mathbf{x} - \mathbf{y}_M\|_2} \end{bmatrix}. \tag{2.15}$$

The ML estimator is asymptotically optimal (efficient) which means that its performance reaches the CRLB accuracy and no unbiased estimator has lower asymptotic mean squared error (MSE) than the ML estimator. To show the optimality of the ML estimator, a simulation is setup in Fig.



(a) Configuration **(b)** RMSE

Figure 2.3: The RMSE of the ML estimator versus the number of anchor nodes. As the number of anchor nodes increases, the performance of the ML estimator reaches to the CRLB. The solid squares represent the anchor nodes and the cross represents the source node.

2.3. The location of the source node is fixed at $[2, 2]^{\mathrm{T}}$ and the anchor nodes are equally spaced on a circle with radius of 10 m and center of $[0, 0]^{\mathrm{T}}$. Fig. 2.3b shows that as the number of anchor nodes increases, the performance of the ML estimator achieve the CRLB in terms of root mean square error (RMSE).

2.4 Semidefinite Programming

We would like to describe what an SDP optimization problem is. The general form of an SDP optimization problem is given by

$$\begin{aligned} \text{minimize} \quad & \mathbf{c}^{\mathrm{T}}\mathbf{x} \\ \text{subject to} \quad & \mathbf{F}(\mathbf{x}) \succeq \mathbf{0} \\ & \mathbf{A}\mathbf{x} = \mathbf{b}. \end{aligned}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the optimization variable. An SDP problem includes a linear cost function subject to several constraints. The equality constraints should also be linear and the inequality constraints are in the form of linear matrix inequalities (LMI) which are convex. An LMI can be expressed as

$$\mathbf{F}(\mathbf{x}) = \mathbf{F}_0 + x_1\mathbf{F}_1 + x_2\mathbf{F}_2 + \cdots + x_n\mathbf{F}_m \succeq \mathbf{0} \tag{2.16}$$

where for a given matrix $\mathbf{A}$, the inequality $\mathbf{A} \succeq \mathbf{0}$ means that $\mathbf{A}$ is a positive semidefinite matrix. A symmetric m by m matrix is positive semidefinite if and only if

$$\mathbf{A} \succeq \mathbf{0} \quad \Leftrightarrow \quad \mathbf{z}^{\mathrm{T}}\mathbf{A}\mathbf{z} \geq 0 \ \text{ for all } \mathbf{z}. \tag{2.17}$$

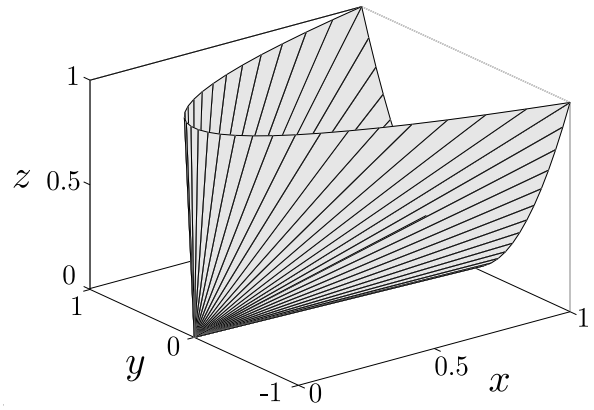


Figure 2.4: The boundary of a positive semidefinite cone.

For instance, it can be easily shown that a symmetric 2 by 2 matrix $\mathbf{A}$ is a positive semidefinite matrix if

$$\mathbf{A} = \begin{bmatrix} x & y \\ y & z \end{bmatrix} \succeq \mathbf{0} \quad \Leftrightarrow \quad xz \geq y^2. \tag{2.18}$$

The boundary of the above constraint is illustrated in Fig. 2.4 which is obviously convex.

2.4.1 Why SDP?

In this section, we describe why we have used SDP as the main tool in our research. In model-based estimation problems, the parameters of interest are estimated by solving an optimization problem. Optimization problems in general are divided into two categories: linear and nonlinear problems. Linear optimization problems such as linear programming and linear least squares either have a closed form solution or the global minimum can be found efficiently in polynomial time. Nonlinear optimization problems include minimizing (or maximizing) a nonlinear cost function subject to several constraints, which can be linear constraints or nonlinear constraints. Such problem have no closed-from solution and iterative numerical methods such as interior-point or trust-region algorithms should be used to find the global minimum. Nonlinear problems can be further divided into two groups of nonconvex and covex problems. Nonconvex optimization problems may have multiple locally optimal point and it is difficult to determine whether the solution is the global minimum or the problem has any solution. On the other hand, convex problems might have several local minima but any local minimum is a global minimum.

As mentioned in the previous section, localization in general requires a nonlinear and noncovex optimization problem to be solved. Therefore, it is difficult to find the optimal solution of a localization problem. One can use an iterative method such as the Gauss-Newton algorithm to solve the original nonlinear and nonconvex problem. Iterative nonlinear solvers like the Gauss-Newton algorithm have several drawbacks. As can be seen in (2.14), the Gauss-Newton algorithm requires an initial point $\mathbf{x}^0$. Finding an initial point is difficult and requires some pre-processing

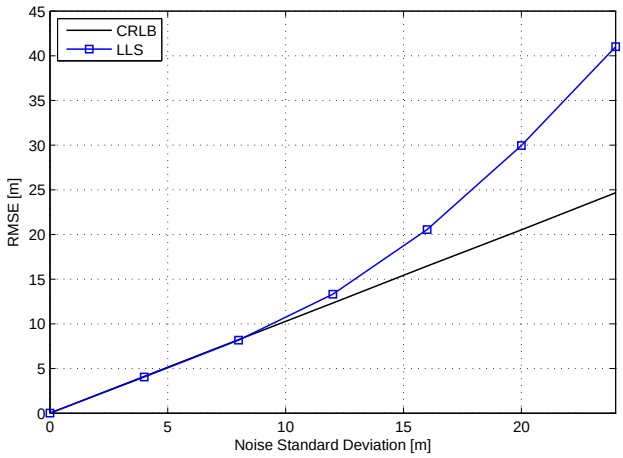
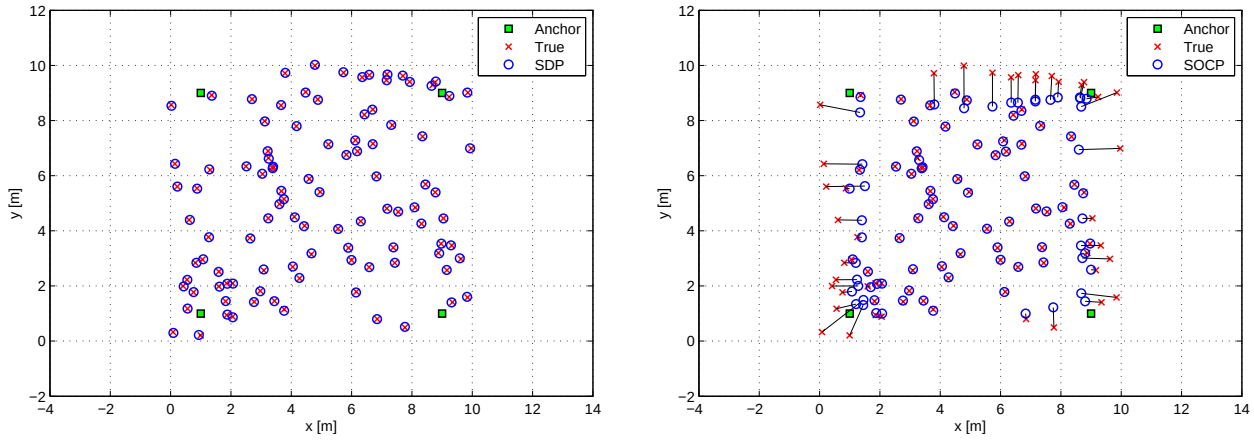


Figure 2.5: The performance of a linear estimator for localization.

time. Moreover, the algorithm may converge to a local minimum or saddle point which causes a large estimation error, as at these points the value of $\mathbf{H}_k$ would be zero and the solution is no longer updated. Therefore, if the initial point of the Gauss-Newton method is not sufficiently close to the global minimum, it is highly probable that the algorithm converges to a stationary point which is not the global minimum. Difficulties in solving the original and optimal localization problem leads to the emergence of suboptimal estimators.

Several studies have focused on linearizing the localization system model and formulating a linear optimization problem. Although in this way the solution is closed-form, the accuracy of linear estimators is far worse than the optimal accuracy, especially in low SNR (i.e., large measurement noises). In Fig. 2.5, we setup a scenario to show the performance a well-known linear estimator. As can be seen, the linear estimator performs very well in high SNR, while its performance degrades as the standard deviation of the measurement noise increases.

Convex relaxation can provide an alternative solution for the convergence problem of the ML estimator. In this technique, the nonlinear and nonconvex ML problem is relaxed into a nonlinear but convex optimization problem such as SDP. Although such problems are solved iteratively, the solver always converges to the global minimum, no matter how the initial point is selected. Although there are several types of convex optimization problems such quadratic programming (QP) and second order cone programming (SOCP), SDP is the most general form of them which can be solved effectively in polynomial time. For instance, a SOCP can be simply written as an SDP and solved by SDP solver. Since SDP is more general, it allows for wider applications and more freedom in relaxation. SDP is very recent in the area of optimization and has been a subject of interest during the past decade mainly because of its several interesting features. In Fig. 2.6, a comparison between the SOCP and SDP estimators for localization is shown. No measurement noise is considered in this simulation. The SDP estimator is able to estimate the locations of all source nodes accurately. On the other hand, SOCP fails to accurately estimate several source nodes even in the absence of noise. The main reason is that due to its limitations the SOCP pushes the



(a) SDP **(b)** SOCP

Figure 2.6: Comparison between the SDP and SOCP estimators. The red crosses show the true locations of the source nodes and the blue circles show their corresponding estimates. The distance between the true location and its estimate (error) is depicted by a solid line.

estimates of the source nodes inside the convex hull of the anchor nodes.

Here, we describe how to convert the original nonlinear and noncovex ML estimator of the localization problem into an SDP problem. The cost function of the ML estimator in (2.13) can be alternately written as

$$\sum_{i=1}^{M} w_i \left(r_i^2 - 2r_i d_i + d_i^2\right) \tag{2.19}$$

where $w_i = (\sigma_i^2)^{-1}$. Since the cost function of an SDP problem should be linear, we need to convert any nonlinear parameter into a linear one. Let us define an auxiliary variable as follows

$$h_i \triangleq d_i^2 = \begin{bmatrix} \mathbf{y}_i \\ -1 \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \mathbf{I}_2 & \mathbf{x} \\ \mathbf{x}^{\mathrm{T}} & z \end{bmatrix} \begin{bmatrix} \mathbf{y}_i \\ -1 \end{bmatrix} \tag{2.20}$$

where $z = \mathbf{x}^{\mathrm{T}}\mathbf{x}$. Therefore, we can alternatively write the minimization problem in (2.13) as

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & \sum_{i=1}^{M} w_i \left(r_i^2 - 2r_i d_i + h_i\right) \\ \text{subject to} \quad & h_i = \begin{bmatrix} \mathbf{y}_i \\ -1 \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \mathbf{I}_2 & \mathbf{x} \\ \mathbf{x}^{\mathrm{T}} & z \end{bmatrix} \begin{bmatrix} \mathbf{y}_i \\ -1 \end{bmatrix}, \\ & z = \mathbf{x}^{\mathrm{T}}\mathbf{x}, \quad h_i = d_i^2. \end{aligned} \tag{2.21}$$

In the minimization problem in (2.21), we have a linear cost function in terms of d_i and h_i. However, we still have two non-affine equality constraints. Now, the trick leading us to an SDP estimator is

to relax the equality constraints. The terms h_i and z are relaxed as follows

$$\begin{aligned} z = \mathbf{x}^{\mathrm{T}}\mathbf{x} \quad &\Rightarrow \quad z \geq \mathbf{x}^{\mathrm{T}}\mathbf{x} \quad \Leftrightarrow \quad \begin{bmatrix} \mathbf{I}_2 & \mathbf{x} \\ \mathbf{x}^{\mathrm{T}} & z \end{bmatrix} \succeq \mathbf{0}_3, \\ h_i = d_i^2 \quad &\Rightarrow \quad h_i \geq d_i^2 \quad \Leftrightarrow \quad \begin{bmatrix} 1 & d_i \\ d_i & h_i \end{bmatrix} \succeq \mathbf{0}_2. \end{aligned} \tag{2.22}$$

Finally, by using the above relaxations, the minimization problem in (2.13) can be converted into an SDP optimization problem as

$$\begin{aligned} \underset{\mathbf{x},z,h_i,d_i}{\text{minimize}} \quad & \sum_{i\in\mathcal{L}_a} w_i \left(r_i^2 - 2r_i d_i + h_i\right) \\ \text{subject to} \quad & h_i = \begin{bmatrix} \mathbf{y}_i \\ -1 \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \mathbf{I}_2 & \mathbf{x} \\ \mathbf{x}^{\mathrm{T}} & z \end{bmatrix} \begin{bmatrix} \mathbf{y}_i \\ -1 \end{bmatrix}, \\ & \begin{bmatrix} \mathbf{I}_2 & \mathbf{x} \\ \mathbf{x}^{\mathrm{T}} & z \end{bmatrix} \succeq \mathbf{0}_3, \ \begin{bmatrix} 1 & d_i \\ d_i & h_i \end{bmatrix} \succeq \mathbf{0}_2. \end{aligned} \tag{2.23}$$

An SDP optimization problem can be solved effectively with interior point methods [15], [16, Ch. 11]. Standard SDP solvers such as SeDuMi [17] or SDPT3 [18] can be employed to solve SDP optimization problems in MATLAB. An alternative SDP relaxation can be derived from the ML problem in (2.13) which has a simpler derivation and enables us to analyze the SDP easier. Squaring both sides of the model in (2.6) yields

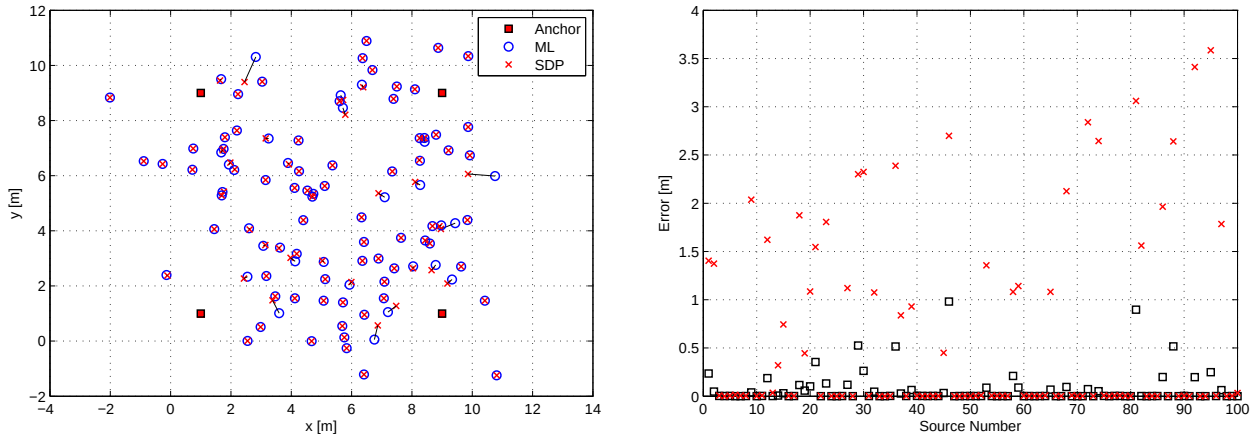
$$\begin{aligned} r_i^2 &= d_i^2 + 2d_i n_i + n_i^2 \\ &\approx d_i^2 + 2d_i n_i \end{aligned} \tag{2.24}$$

where the term n_i^2 is neglected for sufficiently small measurement noise. Using the approximate model in (2.24) instead of the original model in (2.6), we can develop an SDP relaxation as follows

$$\begin{aligned} \underset{\mathbf{x},h_i}{\text{minimize}} \quad & \sum_{i\in\mathcal{L}_a} v_i \left(r_i^2 - h_i\right)^2 \\ \text{subject to} \quad & h_i = \begin{bmatrix} \mathbf{y}_i \\ -1 \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \mathbf{I}_2 & \mathbf{x} \\ \mathbf{x}^{\mathrm{T}} & z \end{bmatrix} \begin{bmatrix} \mathbf{y}_i \\ -1 \end{bmatrix}, \\ & \begin{bmatrix} \mathbf{I}_2 & \mathbf{x} \\ \mathbf{x}^{\mathrm{T}} & z \end{bmatrix} \succeq \mathbf{0}_3. \end{aligned} \tag{2.25}$$

where $v_i = (r_i^2\sigma_i^2)^{-1}$. The SDP in (2.25) has less complexity than the SDP in (2.23), since it has fewer constraints. However, its accuracy can be slightly worse than the SDP in (2.23).

There are several points that can be observed from the SDP problems in (2.23) and (2.25). To relax the ML problem to a SDP problem, the relaxations in (2.22) are used in which the equality



(a) Network **(b)** Benchmark

Figure 2.7: Comparison between the ML and SDP estimators. In the right hand side figure, the red crosses represent the $\sqrt{\hat{z} - \hat{\mathbf{x}}^{\mathrm{T}}\hat{\mathbf{x}}}$ and the black squares represent the distance between the SDP and ML estimates.

is replaced by an inequality. Therefore, whenever the equality holds the solution of the SDP estimator and the ML estimator are the same. Moreover, the equality can be used as a benchmark to determine how accurate the solution of the SDP estimator is. Let $\hat{\mathbf{x}}$ and $\hat{z}$ be the SDP estimates of $\mathbf{x}$ and z, respectively. We can calculate

$$\hat{z} - \hat{\mathbf{x}}^{\mathrm{T}}\hat{\mathbf{x}} \tag{2.26}$$

to determine how reliable the estimate of the SDP is. In Fig. 2.7, a comparison between the SDP and ML estimators is made. There are 4 anchor nodes and 100 source nodes. Fig. 2.7a shows the estimates of the ML and SDP estimators. The SDP estimator provides accurately equal to the ML estimate for the majority of the source nodes. In Fig. 2.7b, the red crosses represent the $\sqrt{\hat{z} - \hat{\mathbf{x}}^{\mathrm{T}}\hat{\mathbf{x}}}$ and the black squares represent the distance between the SDP and ML estimates. Fig. 2.7b shows that whenever $\hat{z} - \hat{\mathbf{x}}^{\mathrm{T}}\hat{\mathbf{x}} \approx 0$, the estimates of the SDP and the ML estimators are the same.

The SDP estimator is looking for a source node location in a higher number of dimensions due to the relaxation $z \geq \mathbf{x}^{\mathrm{T}}\mathbf{x}$. Let us rewrite the relaxation as an equality constraint as

$$\begin{aligned} z &\geq \mathbf{x}^{\mathrm{T}}\mathbf{x} \\ &\geq x^2 + y^2 \\ &= x^2 + y^2 + \epsilon^2, \quad \epsilon^2 \geq 0. \end{aligned} \tag{2.27}$$

It can be seen from (2.27) by introducing the relaxation, we are basically searching for the point in a 3-dimensional space (rather than a 2-dimensional space) in which the cost function has the lowest value. The expression in (2.27) enables us to interpret the behavior of the SDP estimator geographically. The ML estimator is looking for a point on the x-y plane which has the smallest distance to the circles defined by the range measurements. On the other hand, the SDP estimator

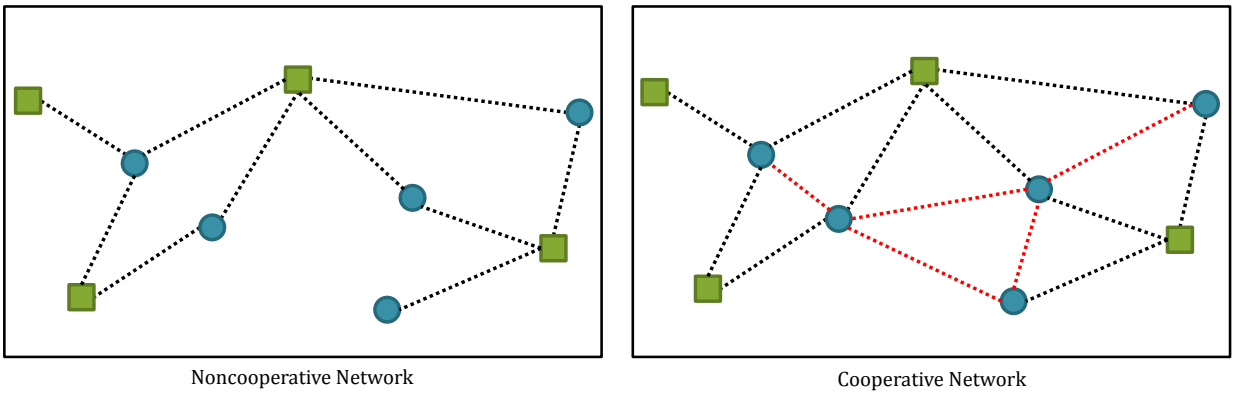


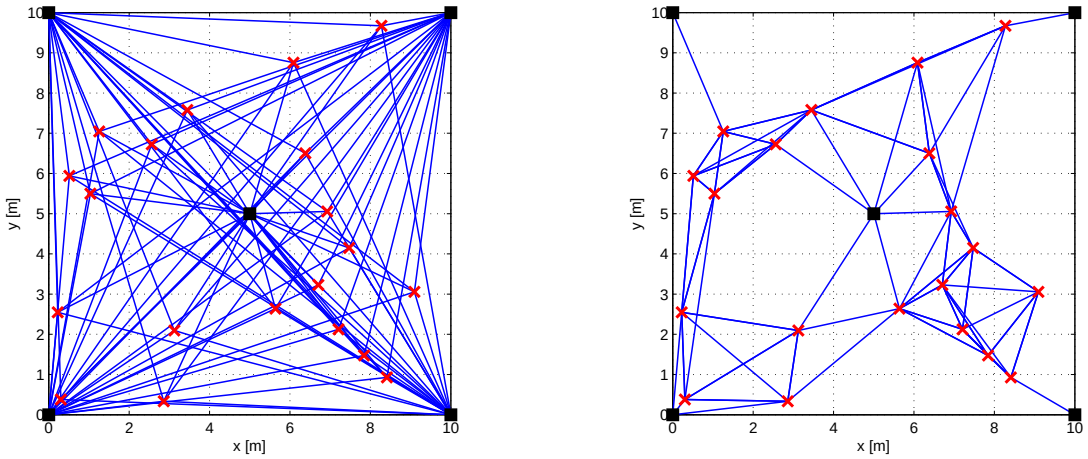
Figure 2.8: Comparing a cooperative network with a noncooperative network. Squares and circles represent the source and anchor nodes, respectively. In cooperative networks, the source nodes communicate with both anchor nodes and other source nodes.

is looking for a point in a 3-dimensional space which has the smallest distance to the spheres whose radii are closet to the range measurements. If $\hat{\epsilon}^2 \approx 0$, the solution of the SDP is located in the x-y plane and will coincide the ML estimate. However, if $\hat{\epsilon}^2 > 0$, the SDP solution is lifted to the third dimension and the estimate of the source location is the projection of the solution to the x-y plane.

2.5 Cooperative Localization

Localization can be divided into two categories: noncooperative and cooperative. In noncooperative localization, the source nodes only communicate with the anchor nodes. Current cellular communication systems are an example of noncooperative networks. As a result, a high density of anchor nodes and longer communication ranges are required to ensure that each source node is connected to a sufficient number of anchor nodes. Applications where there are a limited number of accessible anchor nodes and short communication ranges have led to the emergence of cooperative networks for localization. In cooperative localization, the source nodes communicate not only with the anchor nodes, but also with other source nodes. In this case, the need for dense anchor nodes and longer communication ranges is eliminated. Fig. 2.8 compares a cooperative network with a noncooperative network. Cooperative localization algorithms can be further divided into two groups: centralized and distributed. In centralized localization, all data and measurements are transferred to a central processor where the data is analyzed and the location of all source nodes in the network is estimated simultaneously. Then, the estimated locations are reported back to the source and anchor nodes. Centralized algorithms are less practical in large networks, since their complexities intensify as the network size increases. On the other hand, in distributed localization, the location of each source node is estimated within the device and then reported to the neighboring source and anchor nodes. Distributed algorithms are more popular for large networks. However, centralized algorithms typically provide more accurate estimation than distributed algorithms.

Now let us expand the TOA system model to cooperative networks. A network with N source



(a) Noncooperative network **(b)** Cooperative network

Figure 2.9: Comparison between a cooperative and a noncooperative network. The solid squares represent the anchor nodes and the crosses represent the source nodes.

nodes with unknown locations and M anchor nodes with known locations is assumed. Two sets of measurements are collected in the network: source-anchor and source-source measurements. Denote by $\mathcal{S} = \{1, \ldots, N\}$ the set of indices of the source nodes and by $\mathbf{x}_k = [x_k, y_k] \in \mathbb{R}^2$, $k \in \mathcal{S}$, the coordinates of the kth source location. In cooperative networks, connectivity is typically limited, meaning that the source nodes are not connected to all anchor nodes and other source nodes. In a cooperative network, all nodes are not connected to each other due to limited connectivity. Therefore, we define the following sets to consider the connectivity between nodes:

$$\begin{aligned} \mathcal{A}_k &= \{\ i \mid \text{anchor node } i \text{ is connected to source node } k\ \} \\ \mathcal{B}_k &= \{\ i \mid \text{source node } i \text{ is connected to source node } k\ \} \end{aligned} \tag{2.28}$$

where the first set includes the indices of the anchor nodes connected to the kth source node and the second set includes the indices of the source nodes connected to the kth source node. The cooperative TOA measurement model is expressed as

$$r_{ki} = d_{ki} + n_{ki}, \quad k \in \mathcal{S},\ i \in \mathcal{A}_k \cup \mathcal{B}_k \tag{2.29}$$

where d_{ki} is the true distance between the source node and the ith anchor node

$$\begin{aligned} d_{ki} &= \|\mathbf{x}_k - \mathbf{y}_i\|_2, \quad i \in \mathcal{A}_k \\ d_{ki} &= \|\mathbf{x}_k - \mathbf{x}_i\|_2, \quad i \in \mathcal{B}_k, \end{aligned} \tag{2.30}$$

n_{ki} are the measurement noises which are modeled as independent and identically distributed zero-mean Gaussian random variables with a variance of $\sigma_{ki}^2 = \alpha d_{ki}^{\beta_{ki}}$. Note that we use this model for illustration purposes only. We will modify this model at various points in this work.

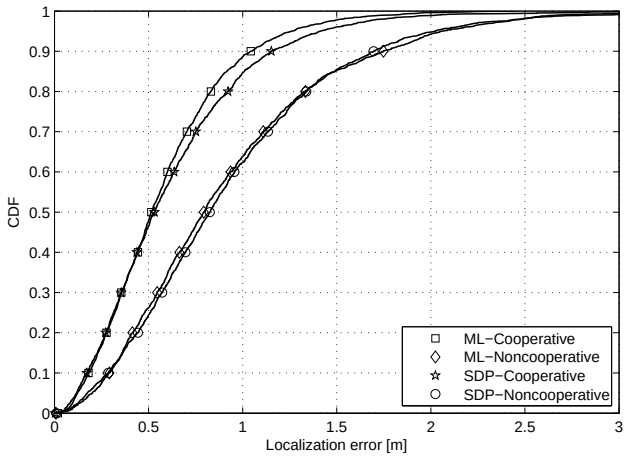


Figure 2.10: The CDF of localization error of cooperative and noncooperative localization.

In Fig. 2.9, we compare a noncooperative network with a cooperative one. For both networks, there are five anchor nodes and twenty source nodes. In the noncooperative network, each source node is only connected to five anchor nodes located at the vertices and the center of a square. In the cooperative network, we set the number of connections to five for each node. In this case, each source node is connected to five neighboring nodes, either anchor nodes or source nodes. Three important features of cooperative networks can be observed in Fig. 2.9. First, source nodes in a noncooperative network usually consume more power, since anchor nodes are not usually close to them and longer communication ranges are required. However in a cooperative network, source nodes can easily find several close neighboring nodes to connect. Therefore, source nodes can operate with considerably lower transmit power. Second, a noncooperative network is more crowded than a cooperative network, since source nodes are only communicating with anchor nodes which can cause severe interference in the network. However in a cooperative network, a group of nodes can communicate with each other in one side of the network without causing any interference with the nodes in another side of the network. Third, as depicted in 2.10, the accuracy of localization is significantly better in cooperative networks than in noncooperative networks. For instance, Fig. 2.10 shows that for the noncooperative network, the error is less than 1.8m at 90% of the time, while in the cooperative case, the error is less than 1.1m 90% of them time or less than 1.8m 98% of the time. The reason for improved accuracy is that in a cooperative network, measured ranges are usually smaller and since the measurement errors are dependent on distances, shorter measured ranges are more accurate than longer measured ranges.

Cooperative localization is a relatively new area of research. The concept of cooperative localization for WSNs was first introduced in [2] where the CRLB and the ML estimator of cooperative localization using RSS measurements are derived. Since its first introduction, many different algorithms have been proposed for cooperative localization. For instance, a linear approximation is developed in [19], although it is not applicable for networks with limited connectivity. A geometrical approximation of the ML estimator is developed in [13]. Convex approximations of the

ML estimator are also proposed in [20] and [21] for RSS and TOA measurements, respectively. However in the majority of the studies mentioned above, the presence of nuisance parameters has not been considered. For instance in [2], the locations of the source nodes are estimated based on the assumption that the transmit powers and the path loss exponent of the environment are known. In [21], it is assumed that the source nodes are time synchronized and pairwise distances between nodes are available. However in practical networks, these assumptions typically are not valid, affecting the localization performance dramatically. Therefore in this work, we examine the performance of cooperative localization in the presence of nuisance parameters. More specifically, we develop a cooperative RSS-based localization technique based on SDP when the transmit powers and path loss exponent are unknown. We also develop a TOA-based localization technique in severe NLOS environments and in asynchronous networks. Lastly, we introduce and address cooperative position tracking for mobile networks.

Chapter 3

Cooperative Received Signal Strength-Based Sensor Localization

3.1 Introduction

Among the different types of measurements, RSS is a popular method mainly because of its low complexity and cost in software and hardware implementations [1]. RSS-based localization has been widely studied in the literature. The ML estimator and the CRLB of cooperative RSS localization were studied in [1]. The cost function of the ML estimator is severely nonlinear and noncovex and, therefore, it can be optimized by iterative algorithms only with an appropriate initialization [22, 23]. The performance of iterative algorithms strongly depends on the initial solution. If the initialization is not sufficiently close to the global minimum, the iterative algorithm may converge to a local minimum or a saddle point causing a large estimation error. Therefore, determining an appropriate initialization point is a crucial problem in optimizing the ML cost function. As a result, some approaches such as grid search methods, linear estimators, and convex relaxation techniques have been introduced to address the ML problem [24, 3, 25]. The grid search methods are not generally popular because they are very time-consuming and require a huge amount of memory when the number of the unknown parameters is too large. Linear estimators having a closed-form solution are usually derived based on many approximations [25, 26, 19, 27] which affect its performance, especially when shadowing is very high [27]. A convex relaxation technique such as SDP is another solution for the ML convergence problem [24, 21, 28, 20, 29]. In the semidefinite relaxation technique, the nonlinear and nonconvex ML problem is transformed into a convex optimization problem. The advantage of an SDP is that its cost function does not have local minima and thus convergence to the global minimum is guaranteed [15, 16]. The downside is that the SDP technique is sub-optimal and cannot achieve the best possible performance in all conditions. Most studies mentioned above on RSS localization assume that the source transmit powers are the same and known [20, 4, 29].

The RSS measurement model depends on the transmit power of the source nodes. Therefore,

the anchor nodes are not able to find the location of a source node if its transmit power is not accounted for. Each source node has a specific transmit power depending on, e.g., its battery and antenna gain. In addition, the transmit power might change with time, e.g., when batteries begin to exhaust. Consequently, each source node has to report its transmit power to anchor nodes constantly during RSS measurements which requires additional hardware and software in both anchor nodes and source nodes making the network more convoluted [1]. When the transmit powers are not available, there are generally two common solutions suggested to address this problem. First, one can estimate the transmit power of the source along with its location [27, 30]. Second, one can eliminate the dependency of the transmit power from the RSS measurement model by using the differential RSS between a source node and two anchor nodes [27, 31, 32]. The number of unknown parameters in the latter method is fewer than in the former method. However, employing the latter method introduces noise correlation and noise enhancement which complicate the computations and degrade the accuracy [27]. It should also be noted that all previous studies on RSS localization with unknown transmit power are for the non-cooperative case. In this work, for the first time, we examine this problem for the cooperative localization case. Some range-free and non-model-based techniques were also suggested [33, 34] in which the localization algorithms do not require the model parameters, such as the transmit power and the path loss exponent, and the estimate of the source location was basically obtained based on the comparisons among RSS measurements. However, having a cooperative network and source nodes with different transmit powers make those algorithms complex and inapplicable.

Thus, cooperative localization using RSS in the practical case where transmit powers are different and unknown is currently an open problem. In this work, we consider this problem and provide a solution. Specifically, the source transmit powers are considered as nuisance parameters and estimated jointly with the source node locations. A novel SDP technique is introduced for this expanded estimation problem. To make the derivations easier to understand, we start by describing the proposed SDP algorithm for the non-cooperative case. Then, the measurement model is extended to the cooperative case and the corresponding SDP algorithm is derived. The original ML estimator is transformed into an approximate nonlinear least squares (NLS) problem. Then, an appropriate relaxation is applied to convert the NLS problem into an SDP optimization problem. It is worth mentioning that the SDP techniques introduced in [24, 21] are not applicable here, since they assumed that noisy pairwise distances between source nodes and anchor nodes are available. However, in our study, the distances cannot be computed because the transmit powers of the source nodes are not available to the estimator. Ouyang et al. [20] also derived an SDP approach which is directly applied to the RSS model. However, they assumed that all source nodes have the same known transmit power which is not practical. Conversely, in the current work, we assume that each source node has a unique transmit power which is not known. The CRLB of this problem is computed. We also propose a linear estimator for comparison with the proposed SDP algorithm. Moreover, we investigate the effect of imperfect knowledge of the path loss exponent (PLE) on the algorithms' accuracy and introduce a novel technique to improve performance when the PLE is

estimated. We also study the computational complexity of the considered algorithms in detail.

3.2 Non-cooperative Localization

This section describes the RSS localization model for the non-cooperative case. In non-cooperative localization problems, only the measurements between a source node and the anchor nodes are considered and the location of each source node is estimated independently. A network in a two-dimensional space is considered. The generalization to a three-dimensional space is straightforward, but is not explored in this work. Let $\mathbf{x} = [a, b]^T \in \mathbb{R}^2$ be the unknown coordinates of the source node to be determined. Denote by $\mathcal{C} = \{1, \ldots, M\}$ the set of indices of the anchor nodes, by $\mathbf{y}_i = [x_i, y_i]^T \in \mathbb{R}^2$, $i \in \mathcal{C}$, the known location of the anchor nodes. Let us define a set

$$\mathcal{A} = \{\ i \mid i \in \mathcal{C}, \text{anchor node } i \text{ is connected to source node}\ \} \tag{3.1}$$

which is the set of the indices of the anchor nodes connected to the source node. The received power (in dBm) at the ith anchor node, P_i, under log-normal shadowing is modeled as [2]

$$P_i = P_0 - 10\beta \log_{10} \frac{d_i}{d_0} + n_i, \quad i \in \mathcal{A} \tag{3.2}$$

where P_0 (in dBm) is the reference power at distance d_0 from the source (which depends on the transmit power), β is the path loss exponent, $d_i = \|\mathbf{y}_i - \mathbf{x}\|$ is the true distance between the source node and the ith anchor node, and n_i are the log-normal shadowing terms modeled as independent and identically distributed (i.i.d.) zero-mean Gaussian random variables with standard deviation σ_{dB} for $i \in \mathcal{A}$. The variance of the shadowing term is constant with distance and only depends on the environment where the network is set up [1]. Without loss of generality, we assume $d_0 = 1$ m. In this case, there are 3 unknown parameters that should be estimated including the source node coordinates and its transmit power.

3.2.1 Maximum Likelihood Estimator

The ML estimator is asymptotically efficient meaning that it can achieve the CRLB accuracy when the number of measurements tends to infinity [10, Ch. 7]. The ML estimator of the measurement model in (3.2) is obtained by the following nonconvex optimization problem [27, 2]

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta} \in \mathbb{R}^3}{\arg\min} \sum_{i \in \mathcal{A}} \left(P_i - P_0 + 10\beta \log_{10} d_i\right)^2 \tag{3.3}$$

where $\boldsymbol{\theta} = [\mathbf{x}; P_0]$. We can write (3.3) equivalently as

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta} \in \mathbb{R}^3}{\arg\min} \sum_{i \in \mathcal{A}} \left(\log_{10} \frac{h_i \lambda_i}{\alpha}\right)^2 \tag{3.4}$$

where $h_i \triangleq d_i^2$, $\lambda_i \triangleq 10^{P_i/5\beta}$, and $\alpha \triangleq 10^{P_0/5\beta}$. As mentioned before, a closed-form solution of (3.3) is not available and it should be approximately solved by numerical techniques [10, 14, 22]. The difficulty in finding the solution of the ML estimator leads us to employ suboptimal estimators, such as SDP and linear least squares (LLS) algorithms. The proposed SDP algorithm is derived in the next section. A linear estimator for the non-cooperative localization case was previously derived in [27].

3.2.2 Semidefinite Programming

This section describes the procedure of converting the ML problem of (3.3) into an SDP optimization problem. By rearranging the logarithmic term and dividing both sides by 5β, (3.2) can be reformulated as

$$\log_{10} d_i^2 \lambda_i = \frac{P_0}{5\beta} + \frac{n_i}{5\beta}. \tag{3.5}$$

Raising both sides to the power of 10 yields

$$d_i^2 \lambda_i = \alpha 10^{n_i/5\beta}. \tag{3.6}$$

For sufficiently small noise, the right-hand side (RHS) of (3.6) can be approximated using the first-order Taylor series expansion as

$$d_i^2 \lambda_i = \alpha \left(1 + \frac{\ln 10}{5\beta} n_i\right). \tag{3.7}$$

This can be rewritten as

$$d_i^2 \lambda_i = \alpha + \epsilon_i, \tag{3.8}$$

where ϵ_i is a zero-mean Gaussian random variable with variance $(\ln 10)^2 \alpha^2 \sigma_{dB}^2 / 25\beta^2$. The corresponding NLS estimator of the unknown parameters $[\hat{\mathbf{x}}; \hat{\alpha}]$ in (3.8) is [10, Ch. 8]

$$[\hat{\mathbf{x}}; \hat{\alpha}] = \underset{[\mathbf{x};\alpha]\in\mathbb{R}^3}{\arg\min} \sum_{i\in\mathcal{A}} \left(d_i^2 \lambda_i - \alpha\right)^2. \tag{3.9}$$

The cost function (3.9) is still nonlinear and noncovex. In the next step, an auxiliary variable z is defined as

$$z = \mathbf{x}^T \mathbf{x}. \tag{3.10}$$

The minimization problem of (3.9) can be relaxed to an SDP optimization problem as [15]

$$\underset{\mathbf{x},z,\alpha,h_i}{\text{minimize}} \quad \sum_{i\in\mathcal{A}} \left(h_i \lambda_i - \alpha\right)^2 \tag{3.11a}$$

$$\text{subject to } h_i = \begin{bmatrix} \mathbf{y}_i \\ -1 \end{bmatrix}^T \begin{bmatrix} \mathbf{I}_2 & \mathbf{x} \\ \mathbf{x}^T & z \end{bmatrix} \begin{bmatrix} \mathbf{y}_i \\ -1 \end{bmatrix}, \tag{3.11b}$$

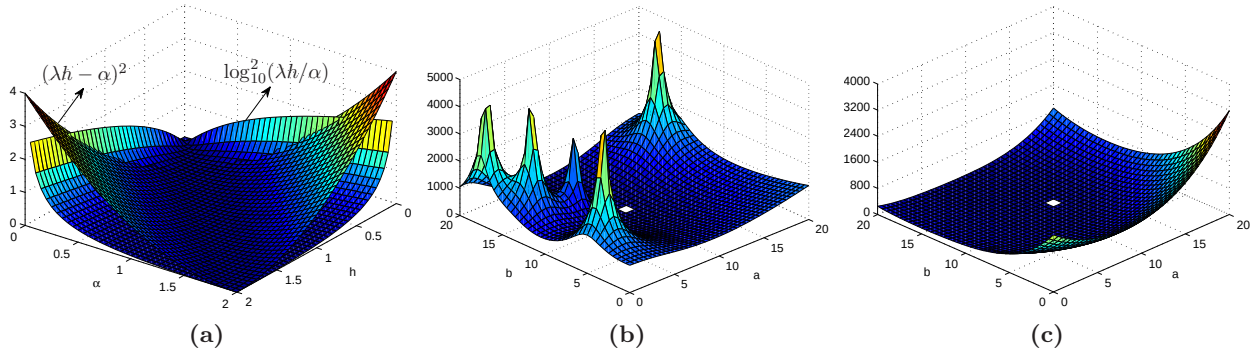


Figure 3.1: (a) illustration of functions $(\lambda h-\alpha)^2$ and $\log_{10}^2(\lambda h/\alpha)$ versus unknown variables h and α (for simplicity, $\lambda=1$), (b) cost function of (3.4), (c) cost function of (3.9) versus a and b coordinates (source location), the minimum of the cost functions is indicated by a white square.

$$\begin{bmatrix} \mathbf{I}_2 & \mathbf{x} \\ \mathbf{x}^T & z \end{bmatrix} \succeq \mathbf{0}_3. \tag{3.11c}$$

The solution of (3.11) can be effectively found with well known algorithms such as interior point methods [15, 16]. In MATLAB simulations, standard SDP solvers such as SeDuMi or SDPT3 [17, 18] are employed to solve SDP optimization problems. Note that we have employed the inequality constraint (3.11c) instead of the equality in (3.10) to relax the cost function in (3.9) to a convex problem in (3.11) [15, 24, 21].

In summary, an SDP solution for RSS localization with unknown transmit power is introduced by converting the nonconvex cost function of the ML estimator into a convex cost function using two steps of approximations and relaxations. In the first step, the ML cost function in (3.4) is approximated by another cost function of (3.9). More specifically, the function $\sum_i(\lambda_i h_i-\alpha)^2$ is substituted for the function $\sum_i \log_{10}^2(\lambda_i h_i/\alpha)$. Fig. 3.1 depicts the two mentioned functions versus unknown parameters h and α (λ is a known parameter). As can be seen, these functions have similar behaviors; the minimum of both functions appears at $\lambda h=\alpha$ and both monotonically increase and decrease in the same regions. Note that the value λ changes the minimum of the functions but does not affect their general behaviors. Hence, for simplicity $\lambda=1$ has been selected for the illustration. Fig. 3.1 also depicts one realization of the ML cost function of (3.4) and the cost function of (3.9). A source is located at $[10,10]^T$ and five anchor nodes are randomly placed inside a square area 20 m $\times$ 20 m. The standard deviation of log-normal shadowing is 3 dB and the path loss exponent is set to 4. Since it is not possible to show a plot in four-dimensional space, the value of P_0 is fixed at the true value (-10 dBm) and the functions are plotted versus a and b coordinates. Fig. 3.1b shows the cost function of the ML estimator given in (3.4) which has a global minimum at $[10.5,11.5]^T$ (the step of the mesh grid is 0.5) and some local minima and saddle points, e.g., a local minimum at $[2.5,17.5]^T$. The cost function of (3.9), shown in Fig. 3.1c, is much smoother than (3.4) and has a global minimum at $[10,11.5]^T$. However, it has some concave areas around its global minimum. It, therefore, still must be relaxed to a convex shape. In the next step, by using the relaxation of (3.11c), the function in (3.9) is relaxed to a convex function in (3.11). The solution of (3.9) and

(3.11) for the source location will coincide if the minimum of (3.11) occurs for $\mathbf{x}$ and z such that $z = \mathbf{x}^T\mathbf{x}$.

3.3 Cooperative Localization

This section describes the cooperative RSS localization model in which there are more than two source nodes with unknown locations, and moreover, source nodes can communicate not only with anchor nodes but also with each other [1]. Indeed, the power of the transmitted signal of each source can be measured at both anchor nodes and other source nodes. Therefore, two sets of RSS measurements are available to the estimator: source-anchor and source-source measurements. Denote by $\mathcal{S} = \{1, \ldots, N\}$ the set of indices of the source nodes and by $\mathbf{x}_j = [a_j, b_j] \in \mathbb{R}^2$, $j \in \mathcal{S}$, the coordinates of the jth source location. Let $\mathcal{A}_j = \{\, i \mid i \in \mathcal{C}, \text{anchor node } i \text{ is connected to source node } j \,\}$ be the set of the indices of the anchor nodes connected to the jth source node and $\mathcal{B}_j = \{\, i \mid i \in \mathcal{S},\ i > j, \text{source node } i \text{ is connected to source node } j \,\}$ be the set of indices of the source nodes connected to the jth source node. The cooperative RSS measurement model is expressed as [2]

$$P_{ij} = P_{0j} - 10\beta \log_{10} d_{ij} + n_{ij}, \quad j \in \mathcal{S},\ i \in \mathcal{A}_j \cup \mathcal{B}_j, \tag{3.12}$$

where P_{0j} is the reference power at a reference distance (1 m) from the jth source (which depends on the transmit power). $d_{ij} = \|\mathbf{y}_i - \mathbf{x}_j\|, i \in \mathcal{A}_j$, and $d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|, i \in \mathcal{B}_j$. In addition, n_{ij} are the log-normal shadowing terms which are modeled as i.i.d. zero-mean Gaussian random variables with standard deviation σ_{dB}. In this case, there are in total $2N + N$ unknown parameters that should be estimated including the source node coordinates and transmit powers.

3.3.1 Maximum Likelihood Estimator

The corresponding ML estimator of the measurement model in (3.12) is obtained by the following nonconvex optimization problem [10, Ch. 7]

$$\hat{\boldsymbol{\phi}} = \underset{\boldsymbol{\phi} \in \mathbb{R}^{3N}}{\arg\min} \sum_{j \in \mathcal{S}} \sum_{i \in \mathcal{A}_j \cup \mathcal{B}_j} \left(P_{ij} - P_{0j} + 10\beta \log_{10} d_{ij}\right)^2 \tag{3.13}$$

where $\boldsymbol{\phi} = [\mathbf{x}^T, \mathbf{p}_0^T]^T$ is the vector of unknown parameters to be estimated including $\mathbf{x} = [\mathbf{x}_1^T, \mathbf{x}_2^T, \ldots, \mathbf{x}_N^T]^T$ and $\mathbf{p}_0 = [P_{01}, P_{02}, \ldots, P_{0N}]^T$. Similarly to the non-cooperative case, (3.13) can be written alternatively as

$$\hat{\boldsymbol{\phi}} = \underset{\boldsymbol{\phi} \in \mathbb{R}^{3N}}{\arg\min} \sum_{j \in \mathcal{S}} \sum_{i \in \mathcal{A}_j \cup \mathcal{B}_j} \left(\log_{10} \frac{h_{ij}\lambda_{ij}}{\alpha_j}\right)^2 \tag{3.14}$$

where $h_{ij} \triangleq d_{ij}^2$, $\lambda_{ij} \triangleq 10^{P_{ij}/5\beta}$, and $\alpha_j \triangleq 10^{P_{0j}/5\beta}$. The proposed SDP algorithm is derived in the next section.

3.3.2 Semidefinite Programming

The SDP relaxation of cooperative RSS localization follows the same procedure as described previously for the non-cooperative case but with a slightly different relaxation. Consider similar approximations as applied to (3.5)-(3.9). Then, the ML cost function in (3.14) can be expressed approximately as [27]

$$[\hat{\mathbf{x}};\hat{\boldsymbol{\alpha}}] = \underset{[\mathbf{x};\boldsymbol{\alpha}]\in\mathbb{R}^{3N}}{\arg\min} \sum_{j\in\mathcal{S}} \sum_{i\in\mathcal{A}_j\cup\mathcal{B}_j} \left(\lambda_{ij}h_{ij} - \alpha_j\right)^2, \tag{3.15}$$

where $\boldsymbol{\alpha} = [\alpha_1,\ldots,\alpha_N]^T$. Similarly to the non-cooperative case, the cost function of (3.15) is relaxed to a convex optimization problem. Let $\mathbf{X} = [\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_N] \in \mathbb{R}^{2\times N}$ be the matrix of the source node coordinates. The auxiliary matrix $\mathbf{Z} \in \mathbb{R}^{N\times N}$ is introduced as

$$\mathbf{Z} = \mathbf{X}^T\mathbf{X}, \tag{3.16}$$

where $[\mathbf{Z}]_{ij} = \mathbf{x}_i^T\mathbf{x}_j$ is the (i,j)th element of the matrix $\mathbf{Z}$. The cost function (3.15) can be relaxed to an SDP optimization problem as [15]

$$\begin{aligned}
\underset{\mathbf{X},\mathbf{Z},\alpha_j,h_{ij}}{\text{minimize}} \quad & \sum_{j\in\mathcal{S}} \sum_{i\in\mathcal{A}_j\cup\mathcal{B}_j} \left(\lambda_{ij}h_{ij} - \alpha_j\right)^2 \\
\text{subject to} \quad & h_{ij} = \begin{bmatrix} \mathbf{y}_i \\ -\mathbf{e}_j \end{bmatrix}^T \begin{bmatrix} \mathbf{I}_2 & \mathbf{X} \\ \mathbf{X}^T & \mathbf{Z} \end{bmatrix} \begin{bmatrix} \mathbf{y}_i \\ -\mathbf{e}_j \end{bmatrix}, \ i\in\mathcal{A}_j, \\
& h_{ij} = \begin{bmatrix} \mathbf{0}_2 \\ \mathbf{e}_i - \mathbf{e}_j \end{bmatrix}^T \begin{bmatrix} \mathbf{I}_2 & \mathbf{X} \\ \mathbf{X}^T & \mathbf{Z} \end{bmatrix} \begin{bmatrix} \mathbf{0}_2 \\ \mathbf{e}_i - \mathbf{e}_j \end{bmatrix}, \ i\in\mathcal{B}_j, \\
& \begin{bmatrix} \mathbf{I}_2 & \mathbf{X} \\ \mathbf{X}^T & \mathbf{Z} \end{bmatrix} \succeq \mathbf{0}_{N+2},
\end{aligned} \tag{3.17}$$

where $\mathbf{e}_i$ is an M by 1 vector in which the ith element is one and other elements are zero. The solution of (3.17) and (3.15) for the source node locations will coincide if the minimum of (3.17) occurs for $\mathbf{X}$ and $\mathbf{Z}$ such that $\mathbf{Z} = \mathbf{X}^T\mathbf{X}$.

3.4 Linear Estimator

In the previous section, we derived an SDP estimator to deal with the convergence problem of the ML estimator and its complexity. Another solution for the ML convergence problem is to use a linear estimator which has an analytical closed-form solution. To obtain a linear estimator, some approximations should be applied to linearize the nonlinear measurement model of (3.2). Linear estimators for RSS localization have been previously considered in the literature [4]. However, to the best of our knowledge, all of them are based on availability of the transmitted power. Here, we derive a similar technique to the one used in [4] with considering unknown transmit powers. Since linearizing the source-source measurements is difficult, we propose a two-step algorithm. In the

first step (called coarse estimation), we estimate the location of each source node independently using only source-anchor measurements. Then, in the next step (called fine estimation), we improve the accuracy of the estimated source locations by using source-source measurements. The main drawback of this approach is that we require at least four source-anchor measurements for each source node, otherwise we cannot find a coarse estimate for the source location in the first step and therefore the second step will be not applicable. We now start by describing the first step. Let us rewrite (3.8) for cooperative localization as ($j \in \mathcal{S}$)

$$h_{ij}\lambda_{ij} = \alpha_j + \epsilon_{ij}, \quad i \in \mathcal{A}_j \tag{3.18}$$

where $\epsilon_{ij} = -n_{ij}\alpha_j(\ln 10)/5\beta$ is a zero-mean Gaussian random variable with variance of

$$\mathrm{E}\{\epsilon_{ij}^2\} = (\ln 10)^2\alpha_j^2\sigma_{dB}^2/25\beta^2. \tag{3.19}$$

Expanding the left-hand side (LHS) of (3.18) and rearranging gives

$$2\lambda_{ij}\mathbf{y}_i^T\mathbf{x}_j - \lambda_{ij}\mathbf{x}_j^T\mathbf{x}_j + \alpha_j = \lambda_{ij}\mathbf{y}_i^T\mathbf{y}_i + \epsilon_{ij}, \quad i \in \mathcal{A}_j. \tag{3.20}$$

Let $\boldsymbol{\theta}_{1,j} = [\mathbf{x}_j^T, \mathbf{x}_j^T\mathbf{x}_j, \alpha_j]^T$ be the unknown vector to be estimated. Then (3.20) can be expressed in matrix form as

$$\mathbf{G}_{1,j}\boldsymbol{\theta}_{1,j} = \mathbf{h}_{1,j} + \boldsymbol{\epsilon}_{1,j}, \tag{3.21}$$

where

$$\mathbf{G}_{1,j} = \begin{bmatrix} \vdots & \vdots & \vdots \\ 2\lambda_{ij}\mathbf{y}_i^T & -\lambda_{ij} & 1 \\ \vdots & \vdots & \vdots \end{bmatrix},$$

$$\mathbf{h}_{1,j} = \begin{bmatrix} \vdots \\ \lambda_{ij}\mathbf{y}_i^T\mathbf{y}_i \\ \vdots \end{bmatrix}, \quad \boldsymbol{\epsilon}_{1,j} = \begin{bmatrix} \vdots \\ \epsilon_{ij} \\ \vdots \end{bmatrix}, \ i \in \mathcal{A}_j \cup \mathcal{B}_j.$$

The LLS solution of (3.21) is [10, Ch. 4]

$$\hat{\boldsymbol{\theta}}_{1,j} = (\mathbf{G}_{1,j}^T\mathbf{W}_{1,j}\mathbf{G}_{1,j})^{-1}\mathbf{G}_{1,j}^T\mathbf{W}_{1,j}\mathbf{h}_{1,j}, \tag{3.22}$$

where $\mathbf{W}_{1,j}$ is the weighting matrix which is the inverse of the covariance matrix of the residual error vector $\boldsymbol{\epsilon}_{1,j}$

$$\mathbf{W}_{1,j} = \mathrm{E}[\boldsymbol{\epsilon}_{1,j}\boldsymbol{\epsilon}_{1,j}^T]^{-1} = ((\ln 10)\alpha_j\sigma_{dB}/5\beta)^{-2}\,\mathbf{I}_{m(j)}. \tag{3.23}$$

where $m(j) = |\mathcal{A}_j|$. The covariance matrix of the estimated vector $\hat{\boldsymbol{\theta}}_{1,j}$ is [10, Ch. 4]

$$\mathbf{C}_{\hat{\theta}_{1,j}} = (\mathbf{G}_{1,j}^T\mathbf{W}_{1,j}\mathbf{G}_{1,j})^{-1}. \tag{3.24}$$

The weighting matrix $\mathbf{W}_{1,j}$ depends on the unknown parameter α_j. Therefore, to compute the weighting matrix, we can eliminate the value of α_j in (3.23) and determine the solution of (3.22). Since α_j is a constant factor in $\mathbf{W}_{1,j}$, its value does not change the solution of $\hat{\boldsymbol{\theta}}_{1,j}$ in (3.22). However, for further computations, we can calculate the weighting matrix $\mathbf{W}_{1,j}$ approximately by $\hat{\alpha}_j$ obtained from (3.22).

Since the elements of estimated parameters in (3.22) are dependent, one method to improve the estimation accuracy, called correction technique [26, 35], is to take this relation between elements of $\hat{\boldsymbol{\theta}}_{1,j}$ into account. Here we extend the correction technique to our problem. The unknown vector $\boldsymbol{\theta}_{1,j}$ can be expressed as

$$\hat{\boldsymbol{\theta}}_{1,j} = \boldsymbol{\theta}_{1,j} + \Delta\boldsymbol{\theta}_{1,j}, \tag{3.25}$$

where $\Delta\boldsymbol{\theta}_{1,j}$ is the estimation error. Squaring the first two elements of (3.25) (element-wise) yields

$$\begin{aligned}
[\hat{\boldsymbol{\theta}}_{1,j}]_1^2 &= [\boldsymbol{\theta}_{1,j}]_1^2 + 2[\boldsymbol{\theta}_{1,j}]_1[\Delta\boldsymbol{\theta}_{1,j}]_1 + [\Delta\boldsymbol{\theta}_{1,j}]_1^2 \\
&\simeq [\boldsymbol{\theta}_{1,j}]_1^2 + 2[\boldsymbol{\theta}_{1,j}]_1[\Delta\boldsymbol{\theta}_{1,j}]_1, \\
[\hat{\boldsymbol{\theta}}_{1,j}]_2^2 &= [\boldsymbol{\theta}_{1,j}]_2^2 + 2[\boldsymbol{\theta}_{1,j}]_2[\Delta\boldsymbol{\theta}_{1,j}]_2 + [\Delta\boldsymbol{\theta}_{1,j}]_2^2 \\
&\simeq [\boldsymbol{\theta}_{1,j}]_2^2 + 2[\boldsymbol{\theta}_{1,j}]_2[\Delta\boldsymbol{\theta}_{1,j}]_2,
\end{aligned} \tag{3.26}$$

where we assume that the estimation error $\Delta\boldsymbol{\theta}_{1,j}$ is sufficiently small. Now, consider the following expressions [26]

$$\mathbf{G}_2\boldsymbol{\theta}_{2,j} = \mathbf{h}_{2,j} + \boldsymbol{\epsilon}_{2,j}, \tag{3.27}$$

where $\boldsymbol{\theta}_{2,j} = [a_j^2, b_j^2, \alpha_j]^T$ and

$$\mathbf{G}_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \ \mathbf{h}_{2,j} = \begin{bmatrix} [\hat{\boldsymbol{\theta}}_{1,j}]_1^2 \\ [\hat{\boldsymbol{\theta}}_{1,j}]_2^2 \\ [\hat{\boldsymbol{\theta}}_{1,j}]_3 \\ [\hat{\boldsymbol{\theta}}_{1,j}]_4 \end{bmatrix}, \ \boldsymbol{\epsilon}_{2,j} = -\mathbf{B}_j\Delta\boldsymbol{\theta}_{1,j},$$

and $\mathbf{B}_j = \mathrm{diag}\{2[\boldsymbol{\theta}_{1,j}]_1, 2[\boldsymbol{\theta}_{1,j}]_2, 1, 1\}$. The LLS solution of (3.27) is [10, Ch. 4]

$$\hat{\boldsymbol{\theta}}_{2,j} = (\mathbf{G}^T\mathbf{W}_{2,j}\mathbf{G})^{-1}\mathbf{G}^T\mathbf{W}_{2,j}\mathbf{h}_{2,j}, \tag{3.28}$$

where $\mathbf{W}_{2,j}$ is the weighting matrix equal to the inverse of the covariance matrix of the residual error vector $\boldsymbol{\epsilon}_{2,j}$

$$\mathbf{W}_{2,j} = \mathrm{E}[\boldsymbol{\epsilon}_{2,j}\boldsymbol{\epsilon}_{2,j}^T]^{-1} = \left(\mathbf{B}_j^T\mathbf{C}_{\hat{\theta}_{1,j}}\mathbf{B}_j\right)^{-1}, \tag{3.29}$$

where $\mathbf{C}_{\hat{\theta}_{1,j}}$ is computed in (3.24). It should be noted that the weighting matrix $\mathbf{W}_{2,j}$ depends on the unknown parameters $\boldsymbol{\theta}_1$. Therefore, the weighting matrix is approximately calculated by available $\hat{\boldsymbol{\theta}}_1$ rather than $\boldsymbol{\theta}_1$. Our simulations show that the accuracy degradation due to the

approximation is not significant. The covariance matrix of the estimated vector $\hat{\boldsymbol{\theta}}_{2,j}$ is [10, Ch. 4]

$$\mathbf{C}_{\hat{\theta}_{2,j}} = (\mathbf{G}_2^T \mathbf{W}_{2,j} \mathbf{G}_2)^{-1}. \tag{3.30}$$

Finally, the estimation of the source location $\hat{\mathbf{x}}_j = [\hat{a}_j, \hat{b}_j]^T$ is [26, 19, 36]

$$\begin{aligned} \hat{a}_j &= \mathrm{sgn}([\hat{\boldsymbol{\theta}}_{1,j}]_1)\sqrt{|[\hat{\boldsymbol{\theta}}_{2,j}]_1|} \\ \hat{b}_j &= \mathrm{sgn}(|[\hat{\boldsymbol{\theta}}_{1,j}]_2|)\sqrt{|[\hat{\boldsymbol{\theta}}_{2,j}]_2|}, \end{aligned} \tag{3.31}$$

where $\mathrm{sgn}(z) = z/|z|$ is the sign function. The covariance matrix of the estimated source location $\hat{\mathbf{x}}_j$ is computed as [19, 36]

$$\mathbf{C}_{\hat{\mathbf{x}}_j} = \mathbf{D}_j^T [\mathbf{C}_{\hat{\theta}_{2,j}}]_{1:2,1:2} \mathbf{D}_j, \tag{3.32}$$

where $\mathbf{D}_j = \frac{1}{2}\mathrm{diag}\{|a_j|^{-1}, |b_j|^{-1}\}$ and $[\mathbf{A}]_{1:n,1:m}$ denotes the upper-left $n \times m$ part of matrix $\mathbf{A}$.

Now we continue describing the second step using both source-source and source-anchor measurements. The relationships between the estimated and the true value of the source location and the transmit power of the source node are

$$\begin{aligned} \mathbf{x}_j &= \hat{\mathbf{x}}_j - \Delta\mathbf{x}_j, \\ P_{0j} &= \hat{P}_{0j} - \Delta P_{0j}, \end{aligned} \tag{3.33}$$

where $\hat{P}_{0j} = 5\beta \log_{10}(\hat{\alpha}_j)$ ($\hat{\alpha}_j = [\hat{\boldsymbol{\theta}}_{2,j}]_3$ is computed for each source node from (3.28)), $\Delta\mathbf{x}_j$ and ΔP_{0j} are the estimation error of the source location and the source transmit power, respectively. Equation (3.12) is a function of the unknown parameters $\mathbf{x}_j$ and P_{0j}. We substitute (3.33) for $\mathbf{x}_j$ and P_{0j} in (3.12). Therefore, ΔP_{0j} and $\Delta\mathbf{x}_j$ are the new unknown parameters to be estimated in (3.12).

$$\begin{aligned} P_{ij} &= (\hat{P}_{0j} - P_{0j}) - 10\beta \log_{10} \|\mathbf{y}_i - (\hat{\mathbf{x}}_j - \Delta\mathbf{x}_j)\| + n_{ij},\ i \in \mathcal{A}_j, \\ P_{ij} &= (\hat{P}_{0j} - \Delta P_{0j}) - 10\beta \log_{10} \|(\hat{\mathbf{x}}_i - \Delta\mathbf{x}_i) - (\hat{\mathbf{x}}_j - \Delta\mathbf{x}_j)\| + n_{ij},\ i \in \mathcal{B}_j. \end{aligned} \tag{3.34}$$

Taking the power of 10 on both sides of (3.34) and rearranging yields

$$\begin{aligned} \hat{\xi}_{ij}\Delta\alpha_j 10^{n_{ij}/10\beta} &= \|\mathbf{y}_i - (\hat{\mathbf{x}}_j - \Delta\mathbf{x}_j)\|, && i \in \mathcal{A}_j, \\ \hat{\xi}_{ij}\Delta\alpha_j 10^{n_{ij}/10\beta} &= \|(\hat{\mathbf{x}}_i - \Delta\mathbf{x}_i) - (\hat{\mathbf{x}}_j - \Delta\mathbf{x}_j)\|, && i \in \mathcal{B}_j, \end{aligned} \tag{3.35}$$

where $\hat{\xi}_{ij} \triangleq 10^{(\hat{P}_{0j} - P_{ij})/10\beta}$ and $\Delta\alpha_j \triangleq 10^{-\Delta P_{0j}/10\beta}$. By using the first-order Taylor series approximation for the third element of the LHS and the element of the RHS, (3.35) turns into

$$\hat{\xi}_{ij}\Delta\alpha_j(1 + \frac{\ln 10}{10\beta} n_{ij}) = \hat{d}_{ij} - \hat{\mathbf{u}}_{ij}^T \Delta\mathbf{x}_j, \qquad i \in \mathcal{A}_j,$$

$$\hat{\xi}_{ij}\Delta\alpha_j(1+\frac{\ln 10}{10\beta}n_{ij}) = \hat{d}_{ij} - \hat{\mathbf{u}}_{ij}^T\Delta\mathbf{x}_j + \hat{\mathbf{u}}_{ij}^T\Delta\mathbf{x}_i, \quad i\in\mathcal{B}_j \tag{3.36}$$

where

$$\hat{\mathbf{u}}_{ij} = \frac{(\hat{\mathbf{x}}_j - \mathbf{y}_i)}{\hat{d}_{ij}},\ \hat{d}_{ij} = \|\mathbf{y}_i - \hat{\mathbf{x}}_j\|,\ i\in\mathcal{A}_j,$$
$$\hat{\mathbf{u}}_{ij} = \frac{(\hat{\mathbf{x}}_j - \hat{\mathbf{x}}_i)}{\hat{d}_{ij}},\ \hat{d}_{ij} = \|\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j\|,\ i\in\mathcal{B}_j.$$

Rearranging (3.36) gives

$$\begin{aligned}\hat{\mathbf{u}}_{ij}^T\Delta\mathbf{x}_j + \hat{\xi}_{ij}\Delta\alpha_j &= \hat{d}_{ij} + \epsilon_{ij}, \quad i\in\mathcal{A}_j,\\ -\hat{\mathbf{u}}_{ij}^T\Delta\mathbf{x}_i + \hat{\mathbf{u}}_{ij}^T\Delta\mathbf{x}_j + \hat{\xi}_{ij}\Delta\alpha_j &= \hat{d}_{ij} + \epsilon_{ij}, \quad i\in\mathcal{B}_j\end{aligned} \tag{3.37}$$

where $\epsilon_{ij} \triangleq (\ln 10)\,\hat{\xi}_{ij}\Delta\alpha_j n_{ij}/10\beta$. Let $\Delta\boldsymbol{\theta}_3 = [\Delta\mathbf{x}_1^T,\ldots,\Delta\mathbf{x}_N^T,\Delta\alpha_1,\ldots,\Delta\alpha_N]^T$ be unknown parameters to be estimated. We can write (3.37) in matrix form as

$$\mathbf{G}_3\Delta\boldsymbol{\theta}_3 = \mathbf{h}_3 + \boldsymbol{\epsilon}_3 \tag{3.38}$$

where

$$\mathbf{G}_3 = \begin{bmatrix}\mathbf{G}_{3,1}\\ \vdots\\ \mathbf{G}_{3,N}\end{bmatrix},\ \mathbf{h}_3 = \begin{bmatrix}\mathbf{h}_{3,1}\\ \vdots\\ \mathbf{h}_{3,N}\end{bmatrix},\ \boldsymbol{\epsilon}_3 = \begin{bmatrix}\boldsymbol{\epsilon}_{3,1}\\ \vdots\\ \boldsymbol{\epsilon}_{3,N}\end{bmatrix}$$
$$\mathbf{G}_{3,j} = \begin{bmatrix}\vdots\\ \mathbf{g}_{ij}\\ \vdots\end{bmatrix},\quad \mathbf{h}_{3,j} = \begin{bmatrix}\vdots\\ \hat{d}_{ij}\\ \vdots\end{bmatrix},\quad \boldsymbol{\epsilon}_{3,j} = \begin{bmatrix}\vdots\\ \epsilon_{ij}\\ \vdots\end{bmatrix}_{i\in\mathcal{A}_j\cup\mathcal{B}_j}$$

and $\mathbf{g}_{ij} = [\mathbf{g}_{ij}^y,\ \mathbf{g}_{ij}^p]$,

$$\begin{aligned}\mathbf{g}_{ij}^y &= \Big[\mathbf{0}_{1\times 2(j-1)}, \hat{\mathbf{u}}_{ij}^T, \mathbf{0}_{1\times 2(N-j)}\Big], && i\in\mathcal{A}_j\\ \mathbf{g}_{ij}^y &= \Big[\mathbf{0}_{1\times 2(j-1)}, \hat{\mathbf{u}}_{ij}^T, \mathbf{0}_{1\times 2(i-j-1)}, -\hat{\mathbf{u}}_{ij}^T, \mathbf{0}_{1\times 2(N-i)}\Big], && i\in\mathcal{B}_j\\ \mathbf{g}_{ij}^p &= \Big[\mathbf{0}_{1\times(j-1)}, \hat{\xi}_{ij}, \mathbf{0}_{1\times(N-j)}\Big], && i\in\mathcal{A}_j\cup\mathcal{B}_j\end{aligned}$$

Since as prior knowledge, we know that the unknown vector $\Delta\boldsymbol{\theta}_3$ is not too large, (3.38) can be solved by the Tikhonov-regularized LS formulation in which the cost function includes the weighted squared norm of $\Delta\boldsymbol{\theta}_3$ to keep it as small as possible. The Tikhonov-regularized LS solution of (3.38) is obtained as [16, Ch. 6]

$$\Delta\hat{\boldsymbol{\theta}}_3 = \underset{\Delta\boldsymbol{\theta}_3\in\mathbb{R}^{3N}}{\arg\min}\ \|\mathbf{G}_3\Delta\boldsymbol{\theta}_3 - \mathbf{h}_3\|^2 + \delta\|\Delta\boldsymbol{\theta}_3\|^2 \tag{3.39}$$

where δ is a regularization parameter controlling the trade-off between $\|\mathbf{G}_3\Delta\boldsymbol{\theta}_3-\mathbf{h}_3\|^2$ and $\|\Delta\boldsymbol{\theta}_3\|^2$. The closed-form solution of (3.39) is given by [16, Ch. 6]

$$\Delta\hat{\boldsymbol{\theta}}_3 = \left(\mathbf{G}_3^T\mathbf{G}_3 + \delta\mathbf{I}_L\right)^{-1}\mathbf{G}_3^T\mathbf{h}_3. \tag{3.40}$$

Finally, the location of each source node is determined as

$$\tilde{\mathbf{x}}_j = \hat{\mathbf{x}}_j - \Delta\hat{\mathbf{x}}_j, \quad j \in \mathcal{S}. \tag{3.41}$$

3.5 Cramér-Rao Lower Bound

In this section, the CRLB of cooperative RSS localization with unknown transmit power is derived. The CRLB defines a lower bound on the variance of any unbiased estimator and is employed as a benchmark for evaluating the performance of estimators [10, Ch. 3], [1, 2, 20]. Since the transmit power of the source nodes is not available to the estimator, it should also be taken into account as an unknown parameter. Let us recall the vector of unknown parameters $\boldsymbol{\phi} = [\mathbf{x}^T, \mathbf{p}_0^T]^T$ including the source node locations $\mathbf{x} = [\mathbf{x}_1^T, \mathbf{x}_2^T, \ldots, \mathbf{x}_N^T]^T$ and the source node transmit powers $\mathbf{p}_0 = [P_{01}, P_{02}, \ldots, P_{0N}]^T$. From the measurement model in (3.12), the logarithm of the probability density function of the RSS measurements is written as

$$\ln p(\mathbf{p};\boldsymbol{\phi}) = k - \frac{1}{2\sigma_{dB}^2}(\boldsymbol{\mu}(\boldsymbol{\phi}) - \mathbf{p})^T(\boldsymbol{\mu}(\boldsymbol{\phi}) - \mathbf{p}) \tag{3.42}$$

where k is a constant which does not depend on the unknown parameters and $\mathbf{p}$ is the measurement vector defined in (3.47). $\boldsymbol{\mu}(\boldsymbol{\phi})$ is the mean of the measurement vector $\mathbf{p}$ defined as

$$\boldsymbol{\mu}(\boldsymbol{\phi}) = \begin{bmatrix} \boldsymbol{\mu}_1(\boldsymbol{\phi}) \\ \vdots \\ \boldsymbol{\mu}_N(\boldsymbol{\phi}) \end{bmatrix}, \ \boldsymbol{\mu}_j(\boldsymbol{\phi}) = \begin{bmatrix} \vdots \\ \mu_{ij} \\ \vdots \end{bmatrix}_{i\in\mathcal{A}_j\cup\mathcal{B}_j} \tag{3.43}$$

where

$$\mu_{ij} = P_{0j} - 10\beta\log_{10} d_{ij}.$$

The CRLB of the unknown parameters are the diagonal elements of the inverse of the Fisher information matrix [10, Ch. 3]. The Fisher information matrix of the measurement model in (3.12) is obtained as [10, Ch. 3]

$$\mathbf{J}(\boldsymbol{\phi}) = \sigma_{dB}^{-2}\mathbf{F}^T\mathbf{F} \tag{3.44}$$

where

$$\mathbf{F} = \frac{\partial\boldsymbol{\mu}(\boldsymbol{\phi})}{\partial\boldsymbol{\phi}} = \begin{bmatrix} \mathbf{F}_1 \\ \vdots \\ \mathbf{F}_N \end{bmatrix}, \ \mathbf{F}_j = \begin{bmatrix} \vdots \\ \mathbf{f}_{ij} \\ \vdots \end{bmatrix}_{i\in\mathcal{A}_j\cup\mathcal{B}_j}$$

and $\mathbf{f}_{ij} = [\mathbf{f}_{ij}^y,\ \mathbf{f}_{ij}^p]$,

$$\begin{aligned}
\mathbf{f}_{ij}^y &= \Big[\mathbf{0}_{1\times 2(j-1)}, \mathbf{u}_{ij}^T, \mathbf{0}_{1\times 2(N-j)}\Big],\ i \in \mathcal{A}_j \\
\mathbf{f}_{ij}^y &= \Big[\mathbf{0}_{1\times 2(j-1)}, \mathbf{u}_{ij}^T, \mathbf{0}_{1\times 2(i-j-1)}, -\mathbf{u}_{ij}^T, \mathbf{0}_{1\times 2(N-i)}\Big],\ i \in \mathcal{B}_j \\
\mathbf{f}_{ij}^p &= \Big[\mathbf{0}_{1\times (j-1)}, 1, \mathbf{0}_{1\times (N-j)}\Big],\ i \in \mathcal{A}_j \cup \mathcal{B}_j \\
\mathbf{u}_{ij} &= \frac{10\beta}{\ln 10}\frac{(\mathbf{y}_i - \mathbf{x}_j)}{d_{ij}^2},\ i \in \mathcal{A}_j \\
\mathbf{u}_{ij} &= \frac{10\beta}{\ln 10}\frac{(\mathbf{x}_i - \mathbf{x}_j)}{d_{ij}^2},\ i \in \mathcal{B}_j.
\end{aligned}$$

Therefore, the CRLB of the unknown parameters $\boldsymbol{\phi}$ is computed as

$$\text{CRLB}([\boldsymbol{\phi}]_r) = \left[\mathbf{J}^{-1}(\boldsymbol{\phi})\right]_{r,r}, \quad r = 1, 2, \ldots, 3N. \tag{3.45}$$

3.6 Path Loss Exponent Estimation

The PLE determines the rate of RSS attenuation with distance. The value of the PLE depends on the propagation environment and varies typically between 2 (free space) and 4 [1]. In RSS-based localization, the accuracy of the estimate of the source node location highly relies on the PLE value. Generally in wireless localization, first the PLE of the environment is obtained through experimental analysis [37]. However, the value of the PLE might change with time, for instance, due to changes in environment. Therefore, it may be required to calibrate the PLE and collect RSS measurements simultaneously. In [38], the PLE is considered as a nuisance parameter and estimated jointly with the source location. However, in our model, estimating the PLE together with the source location and transmit power appears to be very difficult. In the simulation section, we will examine a condition where the estimators have an imperfect estimate of the PLE and its impact on the performance of proposed algorithms will be presented. Here, we introduce a technique to deal with an inaccurate estimate of the PLE. First, we make an estimate for the PLE (based on the network environment) and compute the source location with the proposed algorithms. Then, we update the value of the PLE as

$$\hat{\boldsymbol{\psi}} = \underset{\boldsymbol{\psi}\in\mathbb{R}^{N+1}}{\arg\min} \sum_{j\in\mathcal{S}} \sum_{i\in\mathcal{A}_j\cup\mathcal{B}_j} \Big(P_{ij} - P_{0j} + 10\beta \log_{10} \hat{d}_{ij}\Big)^2, \tag{3.46}$$

where $\boldsymbol{\psi} = [\beta;\ \mathbf{p}_0]$ is the unknown vector to be estimated and $\mathbf{p}_0 = [P_{01}, P_{02}, \ldots, P_{0N}]^T$. $\hat{d}_{ij}$ are the estimates of the distances calculated from the estimates of the source node locations. The least squares solution of (3.46) is obtained as [10, Ch. 4]

$$\hat{\boldsymbol{\psi}} = (\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T\mathbf{p}, \tag{3.47}$$

where $\mathbf{B} = [\mathbf{b},\, \mathbf{B}_0]$ and

$$\mathbf{p} = \begin{bmatrix} \mathbf{p}_1 \\ \vdots \\ \mathbf{p}_N \end{bmatrix},\ \mathbf{p}_j = \begin{bmatrix} \vdots \\ P_{ij} \\ \vdots \end{bmatrix}_{i \in \mathcal{A}_j \cup \mathcal{B}_j}$$

$$\mathbf{b} = \begin{bmatrix} \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_N \end{bmatrix},\ \mathbf{b}_j = \begin{bmatrix} \vdots \\ -10 \log_{10} \hat{d}_{ij} \\ \vdots \end{bmatrix}_{i \in \mathcal{A}_j \cup \mathcal{B}_j}$$

$$\mathbf{B}_0 = \mathrm{diag}\{\mathbf{1}_{n(1)+m(1)}, \ldots, \mathbf{1}_{n(N)+m(N)}\},$$

where $m(j) = |\mathcal{A}_j|$ and $n(j) = |\mathcal{B}_j|$. We estimate $\hat{\beta} = [\hat{\boldsymbol{\psi}}]_1$. Now, the algorithms will be computed with the updated PLE and the iterative procedure continues until the change in $\hat{\beta}$ is sufficiently small. As can be seen in (3.46), we estimate the transmit powers jointly with the PLE. The main reason is that, in our simulations, we observed that the imperfect PLE causes large error in the estimates of the transmit powers, while the estimates of the source locations suffer less from the imperfect PLE estimate. Hence, when updating the PLE value in (3.47), we have used the source location estimates but not the transmit power estimates.

3.7 Complexity Analysis

In this section, we evaluate the computational complexity of the estimators considered in this study based on the total number of floating-point operations or *flops*. We assume that an addition, subtraction, multiplication, division, or square root operation in the real domain can be computed by one flop [39, 40]. For simplicity, we keep only the leading terms of the complexity expressions. We derive the complexity of cooperative approaches which includes non-cooperative case as a special case ($N = 1$). Note that the worst-case complexity is derived without any attempt to optimize computations to take advantage of, e.g., the structure of matrices. The complexity of the considered algorithms is computed as a function of N, the number of source nodes, M, the number of anchor nodes, and $L = \sum_{j \in \mathcal{S}} |\mathcal{A}_j| + |\mathcal{B}_j|$, the total number of connections. Note that for a network with full connectivity, we have $L = N(M + (N-1)/2)$.

3.7.1 Maximum Likelihood

As previously mentioned, the ML estimator is nonlinear and nonconvex. Therefore, ML complexity highly depends on the solution method. In addition, the complexity of every method also depends on many parameters, e.g., the number of iterations, the initial point, or the solution accuracy. Gauss-Newton (GN) is the one of the most popular methods used in solving nonlinear optimization problems [10, Ch. 8]. GN is an iterative technique and requires initialization. Assuming a nonlinear problem with a set of m equations and n unknown variables, the asymptotic computational

complexity ($m \gg n$) of the GN method in each iteration is $O(m^3)$ [41]. The number of iterations depends highly on the initial point and required accuracy. Assuming $O(k)$ iterations on average is required to solve a specific problem, the total complexity of the GN method is $O(km^3)$. For the proposed ML, we have $m = L$, and $n = 3N$. Consequently, the asymptotic complexity of the ML estimator in (3.13) is

$$\text{ML Complexity} \;\simeq\; O(kL^3). \tag{3.48}$$

3.7.2 Semidefinite Programming

Consider the general form of a semidefinite programming problem [15]

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & \mathbf{c}^T\mathbf{x} \\ \text{subject to} \quad & \mathbf{F}(\mathbf{x}) \succeq \mathbf{0}, \end{aligned} \tag{3.49}$$

where $\mathbf{x} \in \mathbb{R}^m$ and

$$\mathbf{F}(\mathbf{x}) = \mathbf{F}_0 + \sum_{i=1}^{m} x_i \mathbf{F}_i.$$

The available data includes the vector $\mathbf{c} \in \mathbb{R}^m$ and $m+1$ symmetric matrices $\mathbf{F}_0, \ldots, \mathbf{F}_m \in \mathbb{R}^{n\times n}$. An SDP problem can be solved by iterative optimization techniques, e.g., interior-point methods [15, 16]. The worst-case computational complexity of solving SDP in each iteration is $O(m^2n^2)$ [15]. The number of iterations is also bounded by $O(\sqrt{n}\log(1/\epsilon))$, where ϵ is the accuracy of SDP solution [15, 17]. For the proposed SDP, we have $m \simeq L + 3N$ and $n \simeq N$. Therefore, the complexity of the SDP in (3.17) is

$$\text{SDP Complexity} \;\simeq\; O(\sqrt{N}(L+3N)^2N^2\log(1/\epsilon)). \tag{3.50}$$

3.7.3 Linear Estimator

Consider a weighted linear least squares problem with a set of m equations and n unknown variables defined as [10, Ch. 5]

$$\hat{\boldsymbol{\theta}} = (\mathbf{G}^T\mathbf{W}\mathbf{G})^{-1}\mathbf{G}^T\mathbf{W}\mathbf{h}, \tag{3.51}$$

where $\boldsymbol{\theta} \in \mathbb{R}^n$, $\mathbf{G} \in \mathbb{R}^{m\times n}$, $\mathbf{W} \in \mathbb{R}^{m\times m}$, and $\mathbf{h} \in \mathbb{R}^m$. Computing $\hat{\boldsymbol{\theta}}$ includes five matrix multiplications and one matrix inversion. Therefore, the computational complexity of (3.51) is $O(2m^2n + mn^2 + mn + n^3 + n^2)$ [39, 42]. Assuming $m \gg 2n$, the complexity of (3.51) is upper bounded by $O(m^3)$. The complexity of the proposed LLS algorithms can be simply expressed as follows,

The cooperative LLS algorithm includes (3.22), (3.28) which are calculated for N source nodes and (3.40). Therefore, from Table 3.1, the total computational complexity of the cooperative LLS approach is

$$\text{LLS Complexity} \;\simeq\; O(6L^2N + 9LN^2 + \sum_{j\in\mathcal{S}} 8|\mathcal{A}_j|^2). \tag{3.52}$$

Table 3.1: Complexity of linear algorithms.

Algorithm	m	n	Complexity
LLS in (3.22)	$\|\mathcal{A}_j\|$	4	$O(8\|\mathcal{A}_j\|^2 + 20\|\mathcal{A}_j\| + 80)$
LLS in (3.28)	4	3	$O(180)$
LLS in (3.40)	L	$3N$	$O(6L^2N + 9LN^2)$
PLE in (3.47)	L	$N+1$	$O(2L^2(N+1) + L(N+1)^2)$

3.7.4 Comparisons

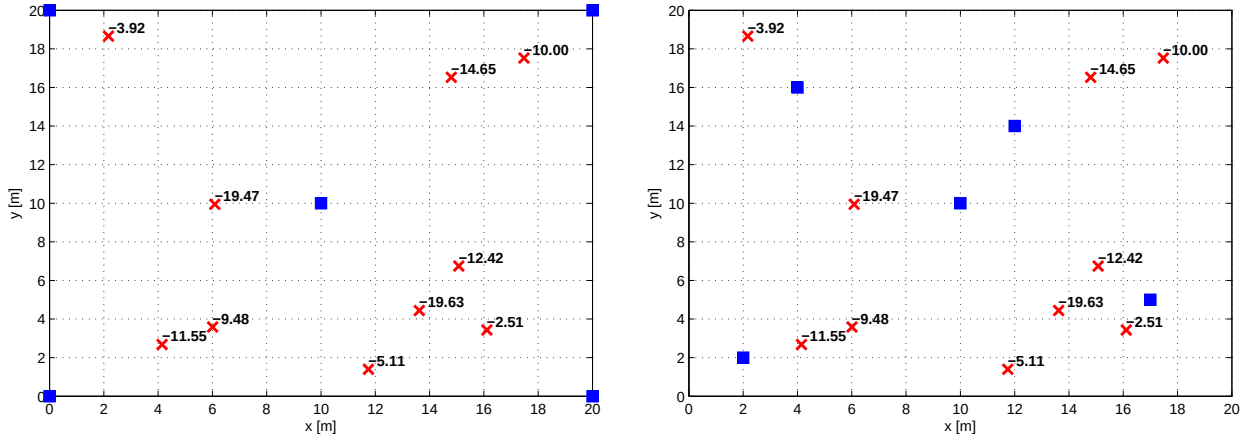
Table 3.2 shows the computational complexity of the algorithms for cooperative localization, assuming a network with full connectivity, where $|\mathcal{A}_j| = M$ and $L = N(M + (N-1)/2)$. Table 3.2 is obtained by substituting M and $N(M+(N-1)/2)$ for $|\mathcal{A}_j|$ and L respectively in (3.48), (3.50), and (3.52). It should be noted that Table 3.2 provides asymptotic complexities of the algorithms, meaning that only the dominating elements are presented. As can be seen, linear estimators (LLS and PLE) require only one iteration. The number of iterations for the SDP algorithm depends on the required accuracy ϵ [15, 43]. The number of iterations for the ML estimator using Gauss-Newton method depends on its initial point and required accuracy [10, Ch. 8]. We will later mention the experimental values for the number of iterations required for each algorithm in the simulation results section. From Table 3.2, we can observe that for a dense network with many source nodes, the complexities per iteration of SDP and ML algorithm are similar but much larger than that of LLS algorithm, however for a modest network size (e.g., 10 source nodes), the complexities per iteration of all algorithms are approximately the same.

3.8 Simulation Results

Computer simulations were conducted to evaluate the performance of the proposed algorithms. Two scenarios were examined; in the first scenario, five anchor nodes were placed regularly on the corners and in the center of a square 20 m $\times$ 20 m and ten source nodes were distributed in a square area 19 m $\times$ 19 m inside the convex hull of the anchor nodes. Fig. 3.2a shows the coordinates of the anchor nodes, source nodes, and reference powers for the first scenario. In the second scenario, the location of the source nodes is the same as in Fig. 3.2a, but the anchor nodes were placed irregularly. Fig. 3.2b shows the configuration of the network in the second scenario. The value of

Table 3.2: Complexity of the algorithms for cooperative localization with full connectivity, $|\mathcal{A}_j| = M$, $L = N(M + (N-1)/2)$. N number of source nodes, M number of anchor nodes, L total number of connections.

Algorithm	Iterations	Complexity per Iteration
ML in (3.13)	k	$O(N^3(M + N/2)^3)$
SDP in (3.17)	$\sqrt{N}\log(1/\epsilon)$	$O(N^4(M + N/2)^2)$
LLS in (3.28)	1	$O(6N^3(M + N/2)^2)$
PLE in (3.47)	1	$O(2N^3(M + N/2)^2)$



(a) The first scenario **(b)** The second scenario

Figure 3.2: The configuration the proposed networks. The solid squares and crosses represent the anchor nodes and source nodes, respectively. The value of the reference power, in dBm, for each source node is indicated next to it.

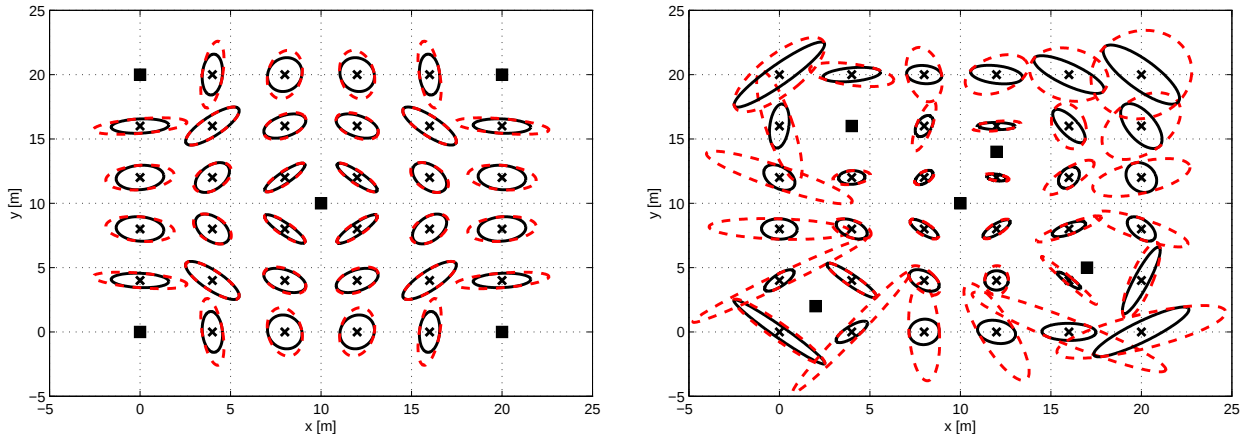
Table 3.3: The Summary of the Considered Algorithms.

Algorithm	Description
SDP-URSS	The proposed SDP algorithm in (3.17) with unknown transmit powers
ML	The ML estimator in (3.13) initialized with the true values
LLS	The proposed linear estimator in (3.41)
SDP-UNS	The SDP estimator in [29] with unknown transmit powers
SDP-RSS	The SDP estimator in [20] with true transmit powers
SDP-RSS-WP	The SDP estimator in [20] with -10 dBm ref. power assumed for all source nodes
SDP-RSS-P2	The SDP estimator in [20] with 2 dB uncertainty about transmit powers
SDP-RSS-P5	The SDP estimator in [20] with 5 dB uncertainty about transmit powers
ML-SDP-URSS	The ML estimator in (3.13) initialized with the solution of SDP-URSS
ML-LSS	The ML estimator in (3.13) initialized with the solution of LLS
ML-RAND	The ML estimator in (3.13) initialized with random values

the path loss exponent β was known and set to 4, unless otherwise noted. The standard deviation of the shadowing σ_{dB} varied from 1 to 8 dB. The ML estimator was solved by the MATLAB routine `lsqnonlin` using the Levenberg-Marquardt method. The proposed SDP was implemented by the `CVX` toolbox [44] using SeDuMi as the solver [17]. The value of the regularization parameter δ was set to 0.1 for the linear estimator. A summary of the considered algorithms is given in Table 3.3.

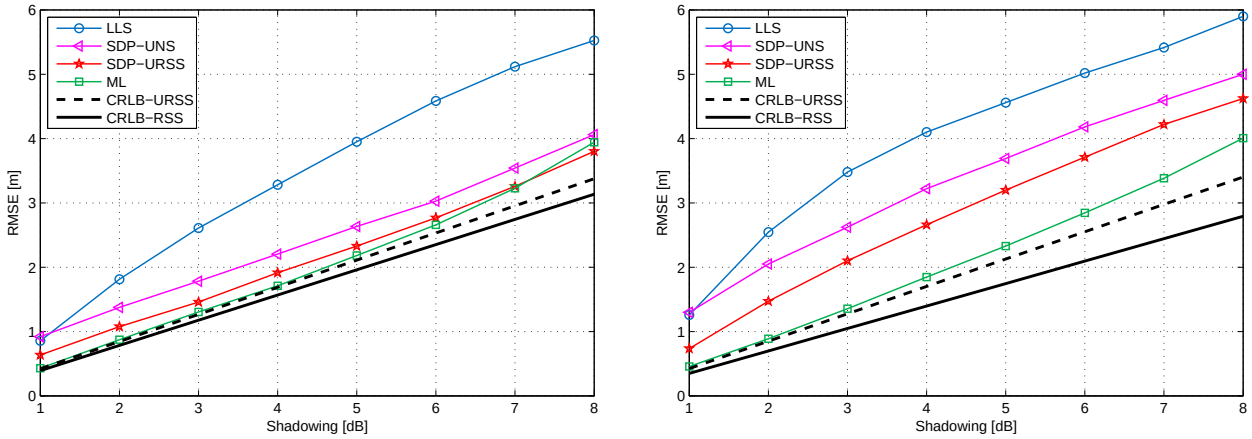
3.8.1 Cramér-Rao Lower Bound (CRLB)

In this section, we show the effect of unknown transmit powers on the CRLB accuracy, evaluated on a regular grid. Figs. 3.3a and 3.3b show the CRLB ellipse [45] of non-cooperative RSS localization with either known or unknown transmit power when the anchor nodes are placed based on the first and second scenarios, respectively. Fig. 3.3a illustrates that there is no significant difference between the CRLB of RSS localization with known and unknown transmit power of a source located inside the convex hull of the anchor nodes, whereas, when the source node is relatively close to the



(a) The first scenario **(b)** The second scenario

Figure 3.3: The CRLB of non-cooperative RSS localization. Black (solid) and red (dashed) circles represent the CRLB ellipse with known transmit power and unknown transmit power, respectively. The solid squares and crosses indicate the anchor nodes and source nodes, respectively. The standard deviation of shadowing term is 3 dB. The difference between the two CRLBs in the second scenario is more significant.



(a) The first scenario **(b)** The second scenario

Figure 3.4: The RMSE of the proposed algorithms versus the standard deviation of shadowing. The proposed SDP (SDP-URSS) performs very well and its performance is very close to the original ML estimator, especially in regular networks.

anchor nodes or between two adjacent anchor nodes, the effect of unknown transmit power on the CRLB is more significant. On the other hand, as depicted in Fig. 3.3b, the differences between the CRLBs for the second scenario are much larger, especially when the source node is located outside the convex hull of the anchor nodes. Thus, when the source node is outside the convex hull, the impact of unknown transmit power is significant. However, the impact is minor when the source node is inside the convex hull. The CRLBs for the cooperative case are more or less similar to the non-cooperative case.

3.8.2 Root-Mean-Square-Error (RMSE)

Now consider the configuration of the network given in Fig. 3.2a. Full connectivity was initially assumed, meaning that each source node was connected to all anchor nodes and also to all other source nodes. The locations of the anchor and source nodes were fixed and 500 measurement noise realizations were drawn. The RMSE of each algorithm is calculated by averaging over all estimated source locations and noise realizations. The RMSE performance of the proposed algorithms is depicted in Fig. 3.4. The solver for the ML estimator was initialized with the true value of the source location. The term URSS is used for RSS localization when the transmit powers of the source nodes are not known and the term RSS is used for the classic RSS localization with known transmit powers as studied in [2, 20]. CRLB-RSS is the classic CRLB with known transmit powers derived in [20]. As expected, in Fig. 3.4a the difference between the CRLB with known transmit power in [2] and the CRLB with unknown transmit power is small, since the source nodes were located inside the convex hull of the anchor nodes. Fig. 3.4a shows that the ML estimator has superior performance for low shadowing standard deviations, but its performance degrades for shadowing standard deviation higher than 5 dB, even though it is initialized with the true values. The proposed SDP (SDP-URSS) performs significantly better than LLS algorithm and is very close to the ML estimator. When shadowing is very high (7 dB and more), the proposed SDP performs slightly better than the ML estimator. The ML estimator is optimal when the number of measurements tends to infinity meaning that no unbiased estimator can have lower RMSE than the ML estimator asymptotically [10, Ch. 7]. However, here the number of measurements is limited, therefore, the optimality of the ML estimator is not guaranteed. Moreover, since both the ML estimator [2] and the proposed SDP are biased (especially when either the number of measurements is limited or shadowing is high), we cannot expect that the ML estimator provides the best accuracy. Note that this behavior has also been observed in [20]. Besides the proposed algorithms, an SDP algorithm given in [29] (labeled as SDP-UNS) is also included in the simulations. It should be noted that SDP-UNS is generally based on the fact that the transmit powers are available. Therefore, we modified the algorithm to estimate jointly transmit powers and source locations. There are many weighting terms in SDP-UNS which depend highly on the transmit powers [29]. To rectify this problem, we first set the value of the weighting terms to 1 and estimate transmit powers and source locations. Then, we updated the weighting terms using the estimated transmit powers and ran the algorithm again. The performance of SDP-UNS falls between LLS and SDP-URSS, even though its computational complexity is higher than SDP-URSS.

The comparison of the proposed algorithms versus the standard deviation of shadowing for the second scenario is illustrated in Fig. 3.4b. Here, there is a considerable gap between CRLB-RSS and CRLB-URSS. The performance order of the algorithms is the same as in the first scenario. In comparison with the previous scenario, the difference between the accuracy of the ML estimator and the proposed SDP is larger. However, the proposed SDP-URSS is still much better than LLS and SDP-UNS estimators.

In Table 3.4, we compare the average running time of the considered algorithms. The running

Table 3.4: The Average Running Time of the Considered Algorithms. $N = 10$, $|\mathcal{A}_j| = 5, L = 95$.

Algorithm	Iterations	Time [ms]
ML	22	268.72
SDP-URSS	19	218.34
SDP-UNS	35	462.16
LLS	1	11.81

time is measured by averaging over 500 noise realizations when the first network with full connectivity is considered and the standard deviation of the shadowing is set to 3 dB. LLS has the lowest complexity and therefore the fastest running time. The required running time of the proposed SDP-URSS is about 20% lower than the ML estimator. From Table 3.2, the complexity per iteration for all algorithms is almost identical for this network size ($N = 10, |\mathcal{A}_j| = 5$). However, they require different numbers of iterations. Moreover, we empirically computed the average number of iterations for the ML and SDP algorithms. The average number of iterations for the ML and SDP-URSS were 22 and 19, respectively. Consequently, the running time of the ML and SDP-URSS is approximately 22 and 19 times higher than LLS (which requires only one iteration). As mentioned in Section 3.7, the running time of the ML estimator depends highly on the initialization. Here, the ML estimator has been initialized with the true values which cuts down its running time and improves its performance. Later, we will discuss the effect of initialization on the complexity and performance of the ML estimator. The computational of complexity of SDP-UNS in [29] is almost the same as that of the proposed SDP-URSS. However, as mentioned above, SDP-UNS should be run twice because of the estimation of the weighting terms. Therefore, the required running time of SDP-UNS is nearly as twice as that of SDP-URSS. Thus, complexity and performance are directly related for the considered algorithms, except for SDP-UNS which was designed for known transmit powers.

Fig. 3.5 illustrates the performance of the proposed SDP compared with the previously studied algorithm in [20] (labeled as SDP-RSS) for the first scenario. Since SDP-RSS is based on the availability of source transmit powers, the true values of the source nodes transmit powers were given to SDP-RSS in Fig. 3.5. In SDP-RSS-WP, the same algorithm in [20] was used, however, the algorithm did not have the exact value of the transmit powers and the value of -10 dBm was given to the algorithm as the reference power for all source nodes. We also proposed a condition where there is uncertainty about the transmit power. Assume that approximate values of the transmit powers, $\bar{P}_{0j}$, are given as

$$\bar{P}_{0j} = P_{0j} + v_j, \tag{3.53}$$

where $v_j \sim \mathcal{N}(0, \sigma_{P_0})$ are Gaussian random variables with standard deviation σ_{P_0}. Instead of the true value of transmit power P_{0j}, the value of $\bar{P}_{0j}$ was given to the SDP algorithm in [20]. In this simulation, the value of σ_{P_0} was set to 2 and 5 dB for SDP-RSS-S2 and SDP-RSS-S5, respectively. SDP-RSS (exact knowledge) outperforms the proposed approach, mainly because the true value of transmit power is available to it. SDP-RSS-S2 performs worse than SDP-RSS especially when

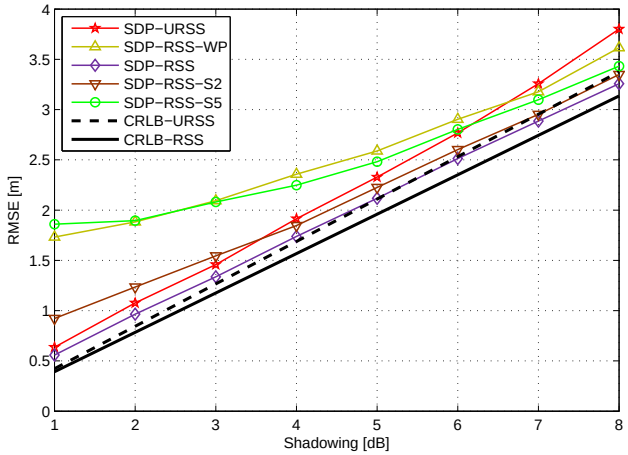
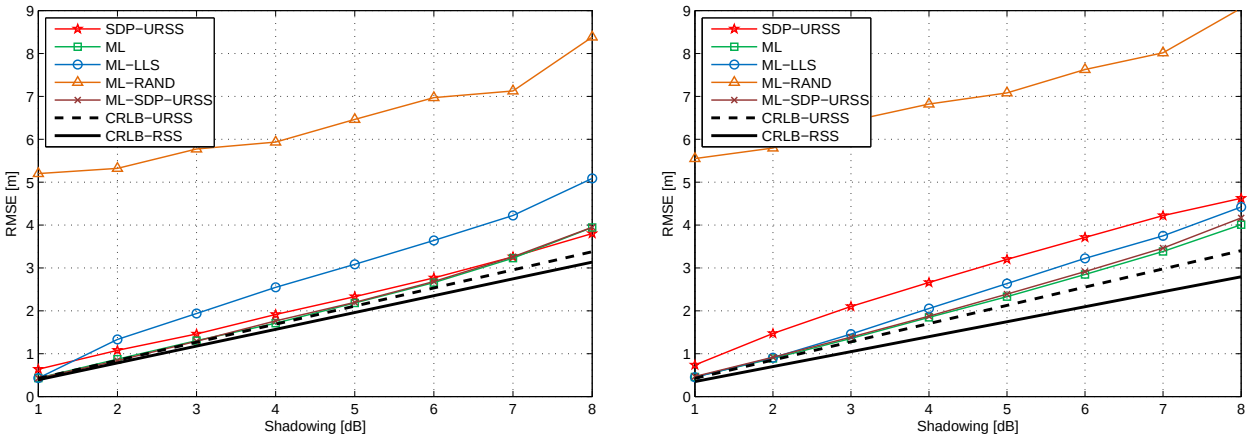


Figure 3.5: The RMSE of the proposed algorithms versus the standard deviation of shadowing for the first scenario with uncertainty in the transmit power. Although in the first scenario unknown transmit powers do not have a huge impact on the CRLB accuracy, employing the proposed SDP (handling unknown transmit powers) is very useful and strongly improves performance.



(a) The first scenario **(b)** The second scenario

Figure 3.6: The RMSE of proposed algorithms versus the standard deviation of shadowing and different initializations for the ML estimator. The initialization has a huge impact on the ML estimator performance. The proposed SDP algorithm provides a suitable initial point for the ML estimator if better accuracy is required.

the shadowing is low. SDP-RSS-WP and SDP-RSS-S5 have poor performance for $\sigma_{dB} < 6dB$. As can be seen, as the standard deviation of the shadowing increases, the effect of transmit power uncertainty on the estimation accuracy decreases since performance is dominated by shadowing. The performance of the proposed SDP-URSS is close to SDP-RSS and has better accuracy than SDP-RSS-S5 and SDP-RSS-WP, especially for low shadowing standard deviation. Hence, although in the first scenario, unknown transmit powers do not have a huge impact on the CRLB, employing an algorithm handling this uncertainty is very useful and strongly improves performance.

Table 3.5: The Average Running Time of the ML Estimator with Different Initializations.

Algorithm	Iterations	Time [ms]
ML	22	268.72
ML-SDP	23	282.51
ML-LLS	26	316.16
ML-RAND	43	514.58

3.8.3 Initialization

It is well-known that the performance of the ML estimator depends heavily on its initial solution, since most implementations of the ML estimator are iterative. To further compare the proposed SDP and the ML estimator, we initialized the solver of the ML estimator with different values. Fig. 3.6a and 3.6b illustrate the RMSE of the proposed algorithms as a function of the standard deviation of the shadowing with different initial points for the solver of the ML estimator for the first and second scenarios, respectively. The curve labeled as ML stands for the ML estimator initialized with the true values, whereas ML-SDP, ML-LLS, and ML-RAND stand for the ML estimator when its solver was initialized with the SDP solution, LLS solution, and random values, respectively. Fig. 3.6 shows that in both scenarios, the ML estimator initialized with the true values outperforms other algorithms as expected. However, the initial point can have a large effect on ML accuracy. The proposed SDP algorithm offers better performance compared to ML-LLS and ML-RAND, especially for large shadow fading standard deviation. It is interesting that when the ML estimator is initialized with the solution of the proposed SDP (ML-SDP-URSS), we can achieve almost the same accuracy as the ML algorithm initialized with the true solution. Therefore, although the proposed SDP algorithm has excellent standalone performance, it can also be used as an initial point for the ML estimator when even better accuracy is required. Fig. 3.6 also shows that randomly initializing the ML estimator yields extremely poor performance even at low σ_{dB} values. Table 3.5 shows the running time of the ML estimator for different initializations. We can see that the initial point has a huge impact on not only the performance of the ML estimator, but also the required number of iterations and the running time. When the initial point is not sufficiently close to the global minimum, the number of iterations increases significantly. Moreover, there is no guarantee that the algorithm converges to the global minimum. This is the major drawback of the ML estimator. In this case, an increase in running time (complexity) does not result in performance improvement because of initialization; in fact the opposite is true.

3.8.4 Connectivity

In the previous simulations, we assumed that the network has full connectivity. However, this assumption is not valid in all practical cases where the connections among sensors are limited. In this section, we examine the first scenario but with limited connectivity. Fig. 3.7 shows the RMSE of the proposed algorithms versus the number of connections when the shadowing standard deviation is 4 dB. For instance, the value of 10 on the x-axis means that each source node is connected to

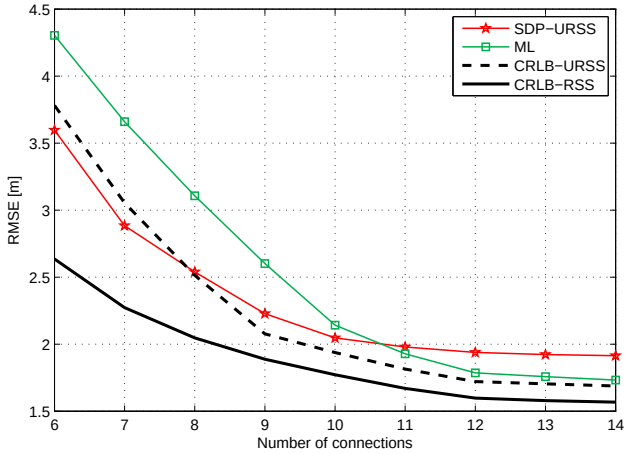


Figure 3.7: The RMSE of the proposed algorithms versus the number of connections. The standard deviation of the shadowing is 4 dB. As the number of connections increases, the estimation accuracy improves. The proposed SDP algorithm has better performance than the ML estimator when the connectivity in the network is limited.

the 10 closest sensors, either anchor nodes or source nodes. The LLS estimator in (3.41) is no longer applicable here because in its formulation, each source node should be connected to at least four noncollinear anchor nodes, but in low connectivity there is no guarantee that each source node can communicate with four neighboring anchor nodes. Fig. 3.7 shows that by increasing the number of connections, the estimation accuracy improves, as expected. Comparing the two CRLBs, we observe that the unknown transmit power has more effect on the CRLB accuracy when the network connectivity is limited. The proposed SDP performs better than the ML estimator when the number of connections in the network is very low. Potential reasons were previously mentioned in the description of Fig. 3.4. When the number of connections is 6 or 7, the proposed SDP is slightly lower than CRLB. As mentioned previously in Section 3.8.2, both SDP-URSS and ML estimators are biased (especially when the number of connections is limited) and thus the CRLB cannot provide an absolute lower bound for them [46, 20, 2].

3.8.5 Impact of Path Loss Exponent

In this section, we investigate the effect of imperfect PLE knowledge on the performance of the proposed algorithms. The first scenario was considered for this simulation. However, we assume that the estimators do not have the exact value of the PLE and instead they have an approximate value modeled as

$$\bar{\beta} = \beta + \mathcal{N}(0,\, 1), \tag{3.54}$$

where $\beta = 4$ in our simulations. Fig. 3.8 shows the performance of the proposed algorithms in this condition. The order of the algorithms remains unchanged compared to Fig. 3.4a. It is obvious that the performance of all the algorithms degrades, especially at low values of σ_{dB}. The proposed SDP-URSS performs better than the ML estimator in the presence of PLE uncertainty when the

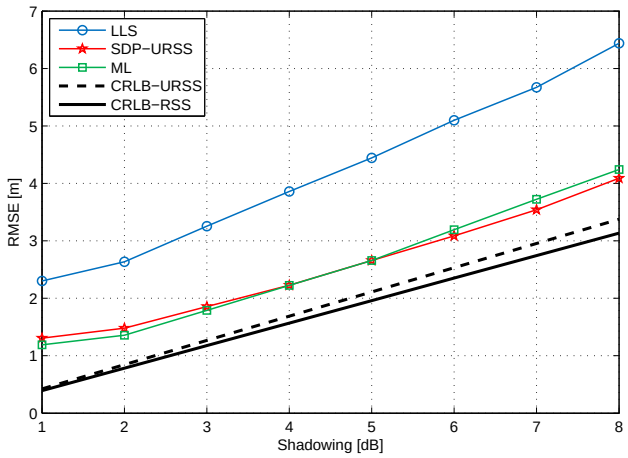


Figure 3.8: The RMSE of the proposed algorithms versus the standard deviation of shadowing for the first scenario with uncertainty in the PLE. Imperfect PLE decreases the performance of all algorithms, especially at low σ_{dB} values. When the shadowing is larger than 6 dB, the proposed SDP is more robust against imperfect PLE than the ML estimator.

shadowing standard deviation is larger than 5 dB. Now, assume that we have an imperfect value of the PLE and the proposed algorithms are computed based on the approach given in (3.47). Fig. 3.9 shows the estimate of the PLE versus the number of iterations and different shadowing standard deviations for the proposed SDP in (3.17). The true value of the PLE was set to 4 and we started the algorithm with the PLE of 3 and 5. It can be seen from Fig. 3.8 that the RMSE performance of the proposed algorithms is poor at the low shadowing standard deviations. However, as depicted in Fig. 3.9, by exploiting the iterative approach given in (3.47), the imperfect PLE estimate converges to a value very close to the true value for a small number of iterations. It is worth mentioning that we have only plotted the convergence of the PLE for the proposed SDP in (3.17). However, the PLE convergence for the ML and LLS algorithms is more or less the same. Our simulations show that once the PLE has converged, the RMSE of the proposed algorithms for the source locations is almost identical to Fig. 3.4a.

3.9 Conclusion

In this work, cooperative RSS-based sensor localization with unknown and different transmit powers was examined. The CRLB of the measurement model was derived and through computer simulations it was shown that the effect of unknown transmit powers on the CRLB accuracy of the source locations depends strongly on the network deployment geometry. A novel SDP technique was derived to estimate the source transmit powers jointly with the source node locations. The complexity analyses of the considered algorithms were presented. Simulation results demonstrated that the proposed SDP algorithm has excellent performance, close to the original ML estimator in most conditions. Not only does the proposed SDP exhibit good accuracy alone but also gives a good initial point for the ML estimator if further refinement is required. We also introduced an

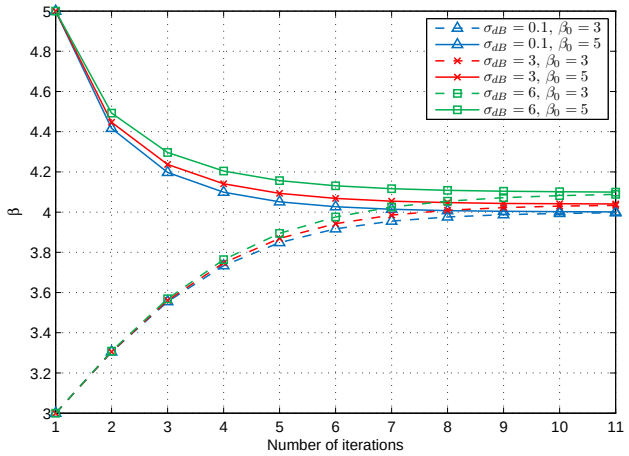


Figure 3.9: The convergence rate of the PLE for the proposed SDP. The algorithm starts with the initial value β_0. The true value of β is 4. Once the PLE has converged, the proposed algorithms achieve performance nearly identical to what is achieved with perfect knowledge.

approach for dealing with imperfect information about the path loss exponent which can strongly impact the performance of RSS-based localization algorithms.

Chapter 4

Cooperative TOA-Based Localization in NLOS Environments

4.1 Introduction

Among different types of measurement methods, TOA is one of the most popular ones used for localization in wireless sensor networks and cellular systems [1]. Cooperative localization is more useful in indoor environments where GPS does not work well. On the other hand, in indoor environments, most connections are NLOS which severely degrades the localization accuracy. In TOA localization, NLOS bias errors are always positive and much larger than the measurement noise. Therefore, they cause large estimation errors if they are either treated as line-of-sight (LOS) connections or not used properly.

Although NLOS scenarios in sensor localization have been widely considered in the literature, we believe that the problem is still not sufficiently solved, especially when the majority of the available connections are NLOS. NLOS mitigation techniques in general are divided into three scenarios based on how much information from NLOS propagation is available. In the first scenario, the estimator knows which connections are NLOS and the distribution of NLOS biases are available [47]. This scenario is rarely practical as determining the distribution of NLOS biases is difficult. In the second scenario, the estimator knows which connections are NLOS, while the distribution of the NLOS biases is not available. In this case, NLOS connections first need to be identified [48]. An identification technique based on ML estimation is described in [49]. Once NLOS connections are identified, they can be either ignored [50] or used in a way which improves performance [51, 52]. In [53], after identifying NLOS connections, the location of the source node is estimated using a weighted linear least squares algorithm in which NLOS connections are assigned lower weights, since they typically have larger errors. In [51], a linear programming method is derived under the assumption that the NLOS connections are perfectly identifiable. The performance of the algorithms in this scenario is dependent on accurate identification of NLOS connections. However, the probability of incorrect identification is always non-negligible. It is possible that a

LOS connection is misidentified as a NLOS connection (false alarm) or vice versa (missed detection) which may significantly degrade estimation accuracy [54].

In most networks, it is very difficult to distinguish NLOS connections from LOS connections and no information about the statistics of the NLOS error may be available. In third scenario, the estimator knows neither which connections are NLOS nor the distribution of NLOS biases. Studies show that in the absence of NLOS information, robust estimators such as the M-estimator [55] and least median of squares (LMS) [56, 57] are more beneficial. In fact, robust estimators are more resistant to large errors and outliers [58]. In [55], an estimator based on the robust Huber function is derived. The Huber function is a piecewise function which is quadratic for small residual errors and linear for large ones [55]. The estimator in [55] is severely nonlinear and nonconvex and requires initialization to find its solution. LMS is another robust estimator used for NLOS localization [56]. However, when more than half of the connections are NLOS, the performance of LMS degrades dramatically [56]. A summary of different NLOS identification and mitigation techniques used in noncooperative localization is provided in [54].

In this work, we examine TOA-based sensor localization in NLOS environments for the second and third scenarios. First, we assume that we are able to distinguish the NLOS connections from LOS connections and propose a novel NLOS mitigation technique for cooperative networks. Second, we assume that it is difficult to identify the NLOS connections and no statistical information about NLOS propagation is available.

4.2 Part I: Known NLOS Connections and Unknown Distribution

Recently, cooperative localization has been studied in the literature [13, 3]. However, most studies on NLOS localization mentioned above have considered only the noncooperative case and there are only a few studies of cooperative localization in NLOS environments [59]. Cooperative localization can improve the accuracy of location estimation, especially when connectivity in the network is limited and source nodes do not have access to a sufficient number of anchor nodes. The algorithms for noncooperative localization in [53, 60, 51, 57] are basically based on the linear estimators in which the measurement model is approximately linearized. However, the measurement model of cooperative localization cannot be easily linearized and most previously considered algorithms for noncooperative localization in NLOS environments are not applicable to cooperative localization. Therefore, further studies on cooperative localization in NLOS environments are warranted.

In this work, we propose a novel SDP estimator for cooperative localization in NLOS environments. We assume that the the estimator can distinguish NLOS connections from LOS connections [48]. However, the distribution of NLOS bias errors is not known to the estimator. It can be shown that the ML estimator discarding NLOS connections and using only LOS connections generally has the optimal performance for sufficiently high signal-to-noise-ratio (SNR). However, the objective function of the ML estimator is highly nonlinear and has no closed-form solution. Thus, the ML problem is converted into a convex SDP optimization problem using only LOS connections which

can be solved effectively. Since the proposed SDP is sub-optimal, it cannot deliver good performance in some conditions (especially if the connectivity is low and the number of LOS is small). Therefore, we exploit NLOS connections to improve the performance of the proposed SDP.

4.2.1 System Model

This section describes the measurement model of cooperative TOA localization in NLOS environments. Denote by $\mathbf{x}_j = [x_j, y_j] \in \mathbb{R}^2$, $j = 1, 2, \ldots, N$ the coordinates of the jth source node with unknown location and by $\mathbf{a}_i = [a_i, b_i]^T \in \mathbb{R}^2$, $i = 1, 2, \ldots, M$ the coordinates of the ith anchor node with known location. The measurements are divided into four sets based on their type of connectivity. Therefore, each connected pair of sensors is placed in one of the following sets:

$$\begin{aligned}
\mathcal{L}_a &:= \{(i,j) : \text{LOS source-anchor connection}\},\\
\mathcal{L}_x &:= \{(k,j) : \text{LOS source-source connection}\},\\
\mathcal{N}_a &:= \{(i,j) : \text{NLOS source-anchor connection}\},\\
\mathcal{N}_x &:= \{(k,j) : \text{NLOS source-source connection}\}.
\end{aligned} \tag{4.1}$$

Assuming perfect synchronization, the measurement model of cooperative TOA localization is defined as [1]

$$\begin{aligned}
r_{ij} &= d_{ij} + n_{ij}, && (i,j) \in \mathcal{L}_a,\\
r_{kj} &= d_{kj} + n_{kj}, && (k,j) \in \mathcal{L}_x,\\
r_{ij} &= d_{ij} + b_{ij} + n_{ij}, && (i,j) \in \mathcal{N}_a,\\
r_{kj} &= d_{kj} + b_{kj} + n_{kj}, && (k,j) \in \mathcal{N}_x
\end{aligned} \tag{4.2}$$

where $d_{ij} = \|\mathbf{a}_i - \mathbf{x}_j\|$, $d_{kj} = \|\mathbf{x}_k - \mathbf{x}_j\|$ are the true distances. b_{ij} and b_{kj} are unknown positive NLOS biases, generally larger than the measurement noise. $n_{ij} \sim N(0, \sigma_{ij}^2)$ and $n_{kj} \sim N(0, \sigma_{kj}^2)$ are the measurement noise modeled as independent identically distributed zero-mean Gaussian random variables with a variance of $\sigma_{ij}^2 = K_E d_{ij}^{\beta_{ij}}$ and $\sigma_{kj}^2 = K_E d_{kj}^{\beta_{kj}}$, respectively. K_E is a constant depending on the environment where the network is set. β_{ij} and β_{kj} are the path loss exponents where we assume

$$\begin{aligned}
\beta_{ij} = \beta_{kj} &= \beta_L \quad (i,j) \in \mathcal{L}_a,\ (k,j) \in \mathcal{L}_x,\\
\beta_{ij} = \beta_{kj} &= \beta_N \quad (i,j) \in \mathcal{N}_a,\ (k,j) \in \mathcal{N}_x.
\end{aligned} \tag{4.3}$$

At the end, we assume that the estimator knows which connections are NLOS, but the distribution of NLOS bias errors are not available.

4.2.2 Performance Analysis

The CRLB of NLOS TOA localization for single-node localization is given [47]. It has been stated in [47] that the CRLB of source location depends only on LOS connections if we can identify all NLOS connections exactly. Therefore, by employing the ML estimator using only LOS connections, we can achieve the CRLB accuracy for sufficiently high SNR. However, if we have any side information about the NLOS bias distribution, we can obtain better accuracy by using the maximum a-posteriori (MAP) estimator which asymptotically reaches the generalized CRLB (G-CRLB). The above results given in [47] for single-node localization can be easily extended to cooperative localization. As mentioned before, we assume that the distribution of the NLOS bias is not available and we can identify the NLOS connections perfectly. Therefore, the ML estimator of measurement model in (4.14) is obtained by the following minimization problem [10]

$$\hat{\mathbf{X}}_{\mathrm{ML}} = \underset{\mathbf{X}\in\mathbb{R}^{2\times 2N}}{\arg\min} \sum_{(i,j)\in\mathcal{L}_a} \frac{1}{\sigma_{ij}^2}\left(r_{ij} - d_{ij}\right)^2 + \sum_{(k,j)\in\mathcal{L}_x} \frac{1}{\sigma_{kj}^2}\left(r_{kj} - d_{kj}\right)^2 \tag{4.4}$$

where $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]$ is the matrix of unknown source locations to be estimated. Unfortunately, the objective function in (4.19) is severely nonlinear and nonconvex and we cannot find a closed-form solution. However, it can be approximately solved by iterative numerical techniques such as the Gauss-Newton method [10, 14]. Iterative methods require a good initial point to make sure that the algorithm converges to the global minimum of the objective function. However, even with a good initial point, it is still possible that the algorithm returns a local minimum resulting in a large estimation error. To avoid the ML convergence problem, we introduce a sub-optimal estimator based on convex optimization.

4.2.3 Semidefinite Programming

In this section, we describe a convex SDP approach for NLOS cooperative localization. By defining some auxiliary variables, the minimization problem in (4.19) can be rewritten as

$$\begin{aligned}
\underset{\substack{\mathbf{X},\mathbf{Z},t_{ij},t_{kj}\\ d_{ij},d_{kj},h_{ij},h_{kj}}}{\text{minimize}} \quad & \sum_{(i,j)\in\mathcal{L}_a} \frac{t_{ij}}{\sigma_{ij}^2} + \sum_{(k,j)\in\mathcal{L}_x} \frac{t_{kj}}{\sigma_{kj}^2} \\
\text{subject to} \quad & t_{ij} = h_{ij} - 2r_{ij}d_{ij} \\
& h_{ij} = z_{jj} + \mathbf{a}_i^T\mathbf{a}_i - \mathbf{a}_i^T\mathbf{x}_j - \mathbf{x}_j^T\mathbf{a}_i, \\
& h_{ij} = d_{ij}^2, \quad (i,j)\in\mathcal{L}_a, \\
& t_{kj} = h_{kj} - 2r_{kj}d_{kj}, \\
& h_{kj} = z_{kk} + z_{jj} - z_{kj} - z_{jk}, \\
& h_{kj} = d_{kj}^2, \quad (k,j)\in\mathcal{L}_x, \\
& \mathbf{Z} = \mathbf{X}^T\mathbf{X}
\end{aligned} \tag{4.5}$$

where $z_{kj} = \mathbf{x}_k^T \mathbf{x}_j$ is the (k,j)th element of the matrix $\mathbf{Z}$. In the next step, we convert (4.20) into a convex optimization problem by using an appropriate relaxation. The minimization problem (4.20) can be relaxed into a convex optimization problem as [43, 21, 15, 27]

$$\underset{\substack{\mathbf{X},\mathbf{Z},t_{ij},t_{kj} \\ d_{ij},d_{kj},h_{ij},h_{kj}}}{\text{minimize}} \quad \sum_{(i,j)\in\mathcal{L}_a} \frac{1}{\sigma_{ij}^2} t_{ij} + \sum_{(k,j)\in\mathcal{L}_x} \frac{1}{\sigma_{kj}^2} t_{kj}, \tag{4.6a}$$

$$\text{subject to } t_{ij} = h_{ij} - 2r_{ij}d_{ij}, \tag{4.6b}$$

$$h_{ij} = z_{jj} + \mathbf{a}_i^T \mathbf{a}_i - \mathbf{a}_i^T \mathbf{x}_j - \mathbf{x}_j^T \mathbf{a}_i, \tag{4.6c}$$

$$h_{ij} \geq d_{ij}^2, \quad (i,j) \in \mathcal{L}_a, \tag{4.6d}$$

$$t_{kj} = h_{kj} - 2r_{kj}d_{kj}, \tag{4.6e}$$

$$h_{kj} = z_{kk} + z_{jj} - z_{kj} - z_{jk}, \tag{4.6f}$$

$$h_{kj} \geq d_{kj}^2, \quad (k,j) \in \mathcal{L}_x, \tag{4.6g}$$

$$\mathbf{Z} \succeq \mathbf{X}^T \mathbf{X}. \tag{4.6h}$$

The solution of (4.6) can be found effectively with interior point methods [15, 16]. Standard SDP solvers such as SeDuMi or SDPT3 [17, 18] can be employed to solve the SDP optimization problem of (4.21) in MATLAB. Note that inequalities given in (4.6d) and (4.6h) can be expressed as a linear matrix inequality (LMI) using the Schur complement [16]

$$\begin{bmatrix} h_{ij} & d_{ij} \\ d_{ij} & 1 \end{bmatrix} \succeq \mathbf{0}_2, \tag{4.7a}$$

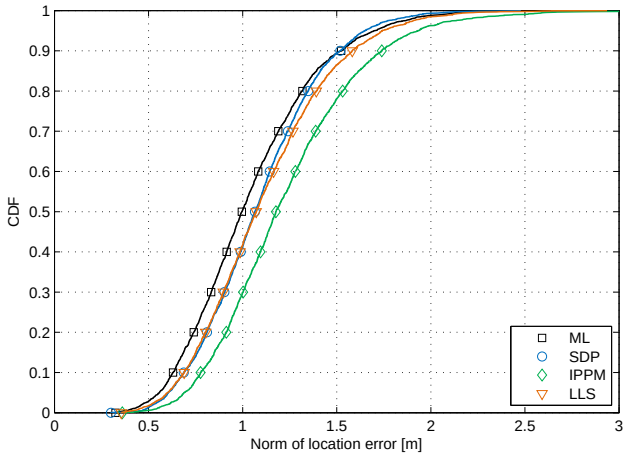
$$\begin{bmatrix} \mathbf{Z} & \mathbf{X}^T \\ \mathbf{X} & \mathbf{I}_2 \end{bmatrix} \succeq \mathbf{0}_{N+2}. \tag{4.7b}$$

As mentioned before, the solution of SDP minimization problem in (4.6) is sub-optimal and cannot achieve the same accuracy as the ML minimization problem in (4.4), especially when either the SNR is low or the number of LOS connections is limited. To enhance the performance of the SDP in (4.6), we try to also use NLOS connections in the minimization problem. NLOS connections basically have large errors and degrade the performance of the algorithm if they are used directly in the objective function in (4.6a). However, they can be used as a constraint on the objective function to limit the feasible region of the solution [51]. Using the fact that the bias of NLOS connections in (4.2) are positive and larger than the measurement noise, we can write

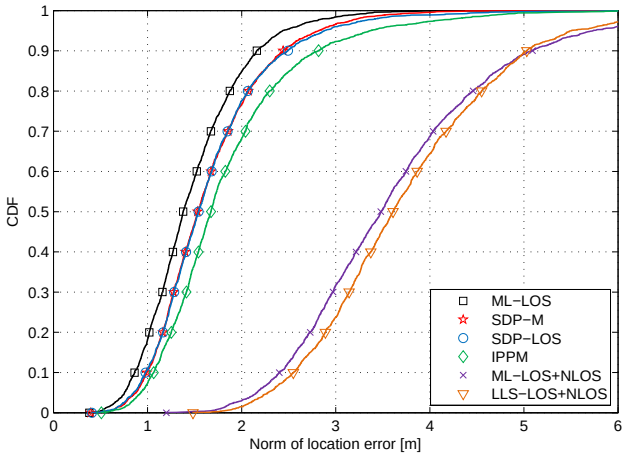
$$\begin{aligned} r_{ij} &\geq d_{ij}, \quad (i,j) \in \mathcal{N}_a, \\ r_{kj} &\geq d_{kj}, \quad (k,j) \in \mathcal{N}_x. \end{aligned} \tag{4.8}$$

Consequently,

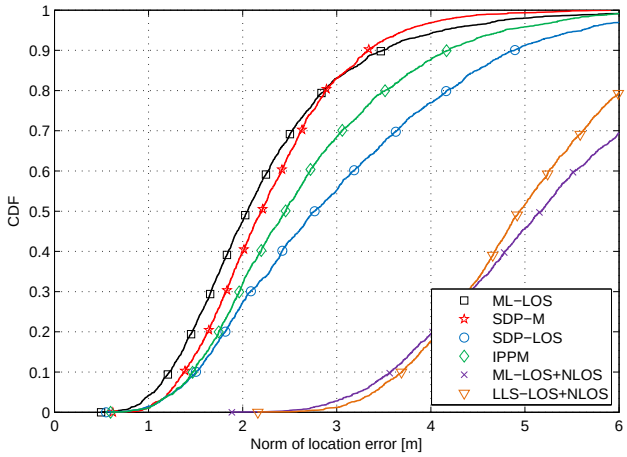
$$r_{ij}^2 \geq \|\mathbf{a}_i - \mathbf{x}_j\|^2, \quad (i,j) \in \mathcal{N}_a,$$



(a) LOS = 8, NLOS = 0



(b) LOS = 5, NLOS = 3



(c) LOS = 3, NLOS = 5

Figure 4.1: The CDF of the proposed algorithms. The network has four anchor nodes and five source nodes. As the number of NLOS connections increases, SDP-LOS degrades substantially as compared to ML-LOS, mainly because the number of LOS connections becomes limited. However, the proposed SDP-M performs nearly as well as ML-LOS.

$$r_{kj}^2 \geq \|\mathbf{x}_k - \mathbf{x}_j\|^2, \quad (k,j) \in \mathcal{N}_x. \tag{4.9}$$

Now, we force the algorithm to minimize the objective function and also to satisfy the constraints in (4.24), meaning that the solution of (4.6) should be located inside the circular regions on NLOS connections defined in (4.9) and also minimize the objective function in (4.6a). Combining the constraints in (4.24) with the minimization problem in (4.21) yields

$$\begin{aligned} \underset{\substack{\mathbf{X},\mathbf{Z},t_{ij},t_{kj} \\ d_{ij},d_{kj},h_{ij},h_{kj}}}{\text{minimize}} \quad & \sum_{(i,j)\in\mathcal{L}_a} \frac{1}{\sigma_{ij}^2} t_{ij} + \sum_{(k,j)\in\mathcal{L}_x} \frac{1}{\sigma_{kj}^2} t_{kj}, \\ \text{subject to} \quad & t_{ij} = h_{ij} - 2r_{kj}d_{ij}, \\ & h_{ij} = z_{jj} + \mathbf{a}_i^T\mathbf{a}_i - \mathbf{a}_i^T\mathbf{x}_j - \mathbf{x}_j^T\mathbf{a}_i, \end{aligned}$$

$$
\begin{aligned}
& h_{ij} \geq d_{ij}^2, \quad (i,j) \in \mathcal{L}_a \\
& t_{kj} = h_{kj} - 2r_{kj}d_{kj}, \qquad (4.10) \\
& h_{kj} = z_{kk} + z_{jj} - z_{kj} - z_{jk}, \\
& h_{kj} \geq d_{kj}^2, \quad (k,j) \in \mathcal{L}_x, \\
& \mathbf{Z} \succeq \mathbf{X}^T\mathbf{X}, \\
& r_{ij}^2 \geq \|\mathbf{a}_i - \mathbf{x}_j\|^2, \quad (i,j) \in \mathcal{N}_a, \\
& r_{kj}^2 \geq \|\mathbf{x}_k - \mathbf{x}_j\|^2, \quad (k,j) \in \mathcal{N}_x. \qquad (4.11)
\end{aligned}
$$

In summary, in the proposed SDP in (4.10), we have used LOS connections to define the objective function to be minimized and NLOS connections to restrict the feasible region of the solution. It should be noted that the last two constraints in (4.10) can be expressed as an LMI as:

$$
\begin{aligned}
& \begin{bmatrix} r_{ij}^2 & (\mathbf{a}_i - \mathbf{x}_j)^T \\ (\mathbf{a}_i - \mathbf{x}_j) & \mathbf{I}_2 \end{bmatrix} \succeq \mathbf{0}_3, \quad (i,j) \in \mathcal{N}_a, \\
& \begin{bmatrix} r_{kj}^2 & (\mathbf{x}_k - \mathbf{x}_j)^T \\ (\mathbf{x}_k - \mathbf{x}_j) & \mathbf{I}_2 \end{bmatrix} \succeq \mathbf{0}_3, \quad (k,j) \in \mathcal{N}_x. \qquad (4.12)
\end{aligned}
$$

4.2.4 Simulation Results

The performance of the proposed algorithm is examined through computer simulations. TOA measurements were generated based on the measurement model in (4.2). The value of path-loss exponents β_L and β_N were set to 2 and 3, respectively [1, 51]. The NLOS bias errors were uniformly distributed between 2 and 8 meters. The ML estimator was solved by MATLAB routine `lsqnonlin`, which uses the Levenberg-Marquardt method. The iterative solver of the ML estimator was initialized with the true value of the source location. The proposed SDP algorithm was solved by the SeDuMi solver [17]. A cooperative NLOS mitigation algorithm called iterative parallel projection method (IPPM) given in [59], a linear least squares estimator (LLS) similar to the algorithm given in [19, 36], and an SDP using LOS only [43] were used for comparison.

A regular network with four anchor nodes and five source nodes was considered. The anchor nodes were located at the vertices of a 10m$\times$10m square and the source nodes were placed randomly inside the square. The value of K_E was set to 0.02 which leads to around 12% error on average. Fig. 4.1a depicts the cumulative distribution function (CDF) of the proposed algorithms when there are no NLOS connections. All of the algorithms perform more or less the same in a LOS environment. In Fig. 4.1b, we increased the number of NLOS connections to 3. The ML-LOS and SDP-LOS stand for the ML estimator in (4.4) and SDP algorithm in (4.6) (which is also studied in [43]) using only LOS connections. The SDP-M is the proposed algorithm in (4.10) exploiting NLOS connections. The SDP-M and SDP-LOS algorithms have the same performance. Hence, the SDP-M has no advantage over SDP-LOS in this case. In ML-LOS+NLOS and LLS-LOS+NLOS algorithms no NLOS identification and mitigation is used and we treated all connections as LOS. It is crystal

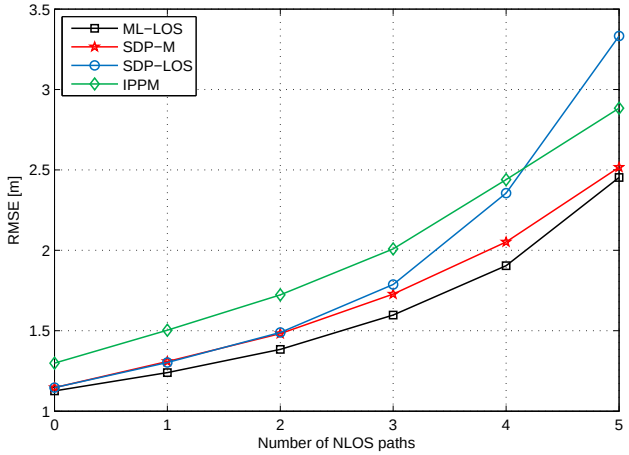


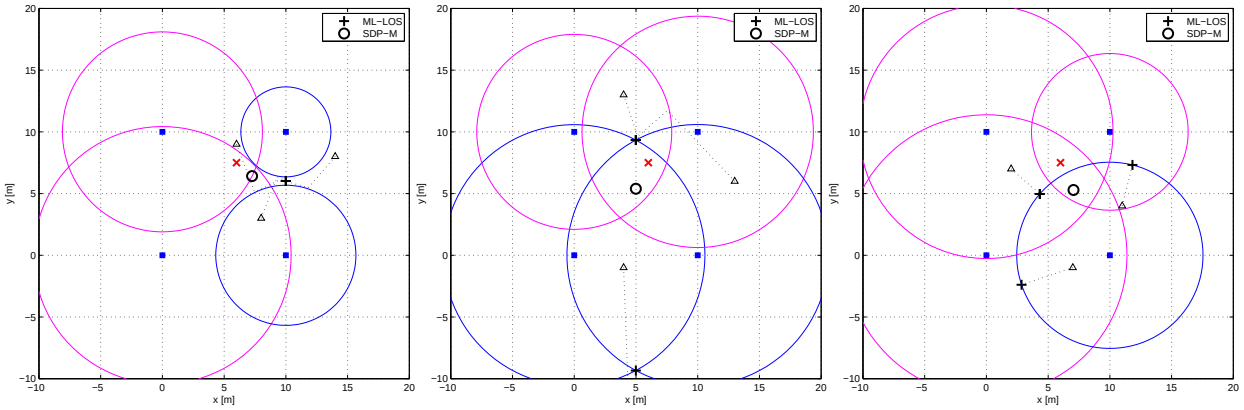
Figure 4.2: The RMSE performance of the proposed algorithms versus the number of NLOS connections. The results are consistent with Fig. 4.1.

clear that NLOS connections degrade the performance of localization considerably if they cannot be identified. Fig. 4.1c shows the performance of the proposed algorithms when we have increased the number of NLOS connections to 5. The performance of SDP-LOS degrades substantially as compared to ML-LOS. Using only LOS connections does not achieve enough accuracy for SDP-LOS, because the number of LOS connections is limited. However, the proposed SDP-M exploiting NLOS connections still has excellent performance. The CDF curve of ML-LOS has a large tail, mainly because of either bad geometry of sensors with LOS connections or large measurement noise. On the other hand, the proposed SDP-M is better than ML-LOS for location error larger than 3 m, because the constraints protect the solution from large estimation errors. It is worth mentioning that ML-LOS algorithm is initialized with the true value, therefore, it is far too optimistic that one can achieve the plotted performance with other initial points.

In Fig. 4.2, we plot the root mean square error (RMSE) performance of the algorithms versus the number of NLOS connections. The results are consistent with Fig. 4.1. As the number of NLOS connections increases, a limited number of LOS connections is available and the performance of SDP-LOS degrades significantly. However, SDP-NLOS exhibits good performance even with a low number of LOS connections.

Degenerate Cases

Since the LOS connections define the objective function of the ML problem in (4.4), we need at least three LOS connections to have a unique solution for the ML estimator. However, sometimes (perhaps even often) this requirement will be not fulfilled in an indoor environment. Under this condition, the ML estimator in (4.4) which has discarded all NLOS connections and uses only LOS connections is not able to find a unique estimate for the source location. On the other hand, this requirement can be omitted for the proposed SDP in (4.10), because the proposed SDP has not only an objective function to be minimized but also has constraints which limit the feasible region



(a) LOS = 2, NLOS = 2 **(b)** LOS = 2, NLOS = 2 **(c)** LOS = 1, NLOS = 3

Figure 4.3: The comparison of SDP-M and ML-LOS algorithms in degenerate cases. The solid blue squares indicate anchor nodes and the red cross indicates the true location of the source node. The blue and magenta circles represent LOS and NLOS connections, respectively. The triangles represent the initial point of the ML estimator algorithm. Unlike the ML estimator, the proposed SDP-M is applicable in degenerate cases when the number of LOS connections is less than three.

which often eliminate the ambiguity in source location. Therefore, even when we have no LOS connection, the proposed SDP returns a point that satisfies only the constraints (in the case that there are many points satisfying the constraints, the proposed SDP returns the analytical center of the feasible region [61]). Fig. 4.3 shows a comparison of the proposed SDP with the ML-LOS estimator in degenerate cases. We have considered the single-node case in the figures to make them less complicated, however, the results are also valid for the cooperative case. In Fig. 4.3a, the number of both LOS and NLOS connections is 2. Since the ML-LOS estimator only uses LOS connections, it returns $[10, 6]^T$ which is the minimum of its objective function. SDP-M tries to minimize the objective function (consisting of LOS connections) and also satisfies the constraints which forces the solution inside the circular regions of the NLOS constraints. In Fig. 4.3b we again have two LOS and two NLOS connections, while the objective function has two local minima ($[5, 9]^T$ and $[5, -9]^T$). In this case, the ML estimator returns two different values for the source location depending on its initialization. For instance, if the ML estimator is initialized with the point $[13, 7]^T$, the algorithm returns $[5, 9]^T$ as a solution, while if it is initialized with the point $[4, -1]^T$, it returns $[5, -9]^T$ which is far away from the true location of the source node causing large estimation error. On the other hand, SDP-M does not need an initialization and gives only one solution. The number of NLOS connections was increased to 3 in Fig. 4.3c. Any point on the blue circle is the minimum of the objective function. The ML-LOS estimator is not reliable in this case, because it returns different values for different initial points. It should be noted that in cooperative localization case, the number of local minima and saddle points in the objective function of the ML estimator is very high, even if the number of LOS connections is larger than 3, and it is highly probable that the algorithm gets stuck in one which is not the global minimum.

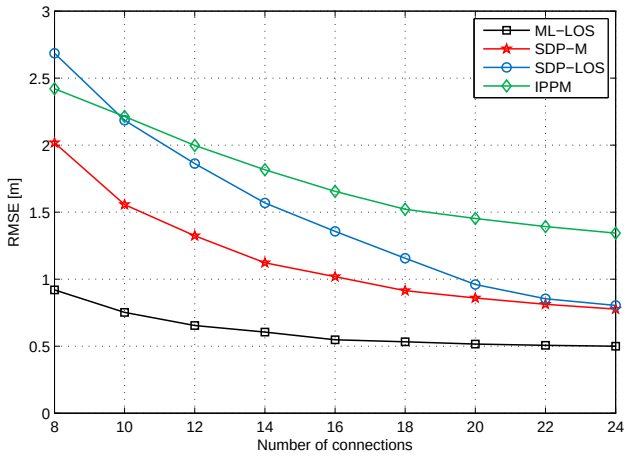


Figure 4.4: The RMSE performance of the proposed algorithms versus the number of connections. Sixty percent of the connections are NLOS. The proposed SDP-M significantly outperforms SDP-LOS when the connectivity is low and a limited number of LOS connections is available.

Connectivity

A network with five anchor nodes and twenty source nodes placed randomly inside a 10m $\times$ 10m square is considered. Fig. 4.4 shows the RMSE performance of the proposed algorithm versus the number of connections. For instance, 10 on the x-axis means that each source node is connected to the 10 closest sensors either anchor nodes or other source nodes. Moreover, 60% of the connections are NLOS. The performance of all algorithms improves as the number of connections increases. The reduction in the RMSE of ML-LOS is not great, mainly because the algorithm was initialized with the true values and thus even in low connectivity it has a low RMSE. The proposed SDP-M exploiting NLOS connections performs better than SDP-LOS in low connectivity (about 0.7 m when the number of connections is 8). As the number of connections increases, the number of LOS connections increases and the accuracy in both SDP-M and SDP-LOS is dominated by LOS connections. For instance, when the number of connections is 24 (meaning that we have full connectivity in the network), both SDP-M and SDP-LOS have nearly the same RMSE of 0.75 m and NLOS connections do not help SDP-M to have better accuracy.

4.3 Part II: Unknown NLOS Connections and Distribution

In this work, TOA-based sensor localization in unknown NLOS environments is studied. We assume that the estimator has no information about the distribution of the NLOS measurements and that NLOS signal identification is also infeasible. The performance analysis of the model is investigated through the ML estimator. It will be shown that the corresponding ML estimator of the measurement model is a constrained underdetermined nonlinear problem which is difficult to solve. As a result, a NLOS mitigation technique based on convex SDP optimization is proposed. As opposed to the ML estimator, the proposed SDP formulation is efficiently solvable without requiring any initialization. The performance of the proposed estimator is compared with several previously proposed techniques through computer simulations. The average running times of the different estimators is also considered.

4.3.1 System Model

The measurement model of NLOS TOA localization will now be described. Consider a network of M anchor nodes and one source node to be localized. Let $\mathbf{x} \in \mathbb{R}^2$ be the unknown coordinates of the source node and $\mathbf{y}_i \in \mathbb{R}^2$, $i \in \mathcal{A}$ be the known coordinates of the ith anchor node, where $\mathcal{A} = \{1, 2, \ldots, M\}$. The TOA measurements are divided into two sets as follows

$$\begin{aligned} \mathcal{L} &:= \{i \mid i \in \mathcal{A}, i\text{th link is LOS}\}, \\ \mathcal{N} &:= \{i \mid i \in \mathcal{A}, i\text{th link is NLOS}\}. \end{aligned} \tag{4.13}$$

In TOA, the travel times of signals between the source node and the anchor nodes are measured [1]. The distance measurement between the source node and the ith anchor node in TOA localization can be modeled as [47]

$$\begin{aligned} r_i &= d_i + n_i, && i \in \mathcal{L}, \\ r_i &= d_i + b_i + n_i, && i \in \mathcal{N} \end{aligned} \tag{4.14}$$

where $r_i = ct_i$, t_i is the signal propagation time between those nodes, and c is the propagation velocity of the signal (e.g., $c \approx 3 \times 10^8$ m/s in free space). d_i is the true distance between the source node and the ith anchor node

$$d_i = \|\mathbf{y}_i - \mathbf{x}\|_2, \quad i \in \mathcal{A}. \tag{4.15}$$

The n_is represent the measurement errors due to thermal noise at the receivers. These errors are modeled as independent zero-mean Gaussian random variables with variances of σ_i^2. This variance in the TOA measurements is related to the captured signal-to-noise ratio (SNR) [1, 11]. Furthermore, the captured SNR is a function of the distance and path-loss exponent [12, 2]. Therefore,

the variance in TOA measurements due to thermal noise can be modeled as [13]

$$\sigma_i^2 = \alpha d_i^{\beta_i} \tag{4.16}$$

where α is a constant defining the relationship between the noise variance and the true distance raised to the path loss exponent β_i. α depends on the environment where the sensor network is placed and β_i is defined as [13]

$$\begin{aligned} \beta_i &= \beta_L, \quad i \in \mathcal{L}, \\ \beta_i &= \beta_N, \quad i \in \mathcal{N}. \end{aligned} \tag{4.17}$$

Last, but not least, b_i are unknown positive NLOS biases which are typically much larger than the measurement noise, n_i. There are different models proposed for NLOS biases in the literature [54]. Gaussian [62], exponential [52], and uniform [51] distributions are usually applied to model the NLOS propagation biases in a network. In this paper, the NLOS biases are drawn from an exponential distribution. However, we assume that the distribution of NLOS biases is unknown to the estimator. Moreover, the estimator does not even know which connections are NLOS.

4.3.2 Maximum Likelihood Estimation

The performance analysis of NLOS localization for TOA measurements is given in [47]. It was shown that when the distribution of NLOS bias errors is not available, the Cramér-Rao lower bound (CRLB) of the source location estimate depends only on the LOS connections [47]. This means that, under this condition, the best accuracy can be achieved if the NLOS connections are identified and discarded, and the source location is estimated by using only the LOS measurements. The CRLB can be achieved asymptotically (when the number of measurements tends to infinity) by the ML estimator. Therefore, assuming a sufficient number of measurements is available, the ML estimator using only LOS connections can be used as a benchmark for comparing the performance of different estimators. The ML estimator is obtained by maximizing the likelihood function [10]. After some calculations and manipulations, the ML estimator using only LOS connections is obtained by the following minimization problem [10]

$$\hat{\mathbf{x}}_{\mathrm{ML-LOS}} = \underset{\mathbf{x}\in\mathbb{R}^2}{\arg\min} \sum_{i\in\mathcal{L}} w_i \left(r_i - d_i\right)^2 \tag{4.18}$$

where $w_i = 1/\sigma_i^2$ in the cost function are the weighting elements. Since in this work, we assume that NLOS connections are not distinguishable, the ML estimator in (4.18) is not applicable here, but will serve as a lower bound on the accuracy of the considered estimators whenever there are a sufficient number of LOS connections. When the number of LOS connections in the network is fewer than three, we can no longer use the ML estimator in (4.18), since in this case, no unique solution can be found for the estimator[22].

Now we derive an estimator assuming no information about NLOS connections is available. Because we do not know which connections are NLOS, we assume all measurements are NLOS and estimate NLOS biases jointly with the source location. Let $\mathbf{b} = [b_1, b_2, \ldots, b_M]^T$ be the unknown vector of NLOS biases. The ML estimator of the unknown parameters $[\mathbf{x}; \mathbf{b}]$ is obtained by following optimization problem

$$\begin{aligned}[\hat{\mathbf{x}}; \hat{\mathbf{b}}] = \underset{[\mathbf{x};\mathbf{b}]\in\mathbb{R}^{2+M}}{\arg\min} & \sum_{i\in\mathcal{A}} w_i \left(r_i - d_i - b_i\right)^2, \\ \text{s. t. } & b_i \geq 0, \end{aligned} \tag{4.19}$$

which is a constrained nonlinear underdetermined problem (i.e., there are more unknown parameters than measurements) and finding its solution is computationally difficult. The solution of (4.19) can be approximately found with iterative algorithms such as the Gauss-Newton method [63]. However, iterative algorithms require a good starting point to guarantee that the algorithm converges to the global minimum of the cost function. Moreover, since the problem is underdetermined, there might be an infinite number of solutions [63]. In the next section, we derive a NLOS mitigation technique based on convex SDP optimization. The advantage of the SDP technique over the ML technique is that the SDP technique requires no initialization [15].

4.3.3 Semidefinite Programming

In this section, the proposed SDP estimator is described. Again, since we do not know whether a connection is NLOS or not, we assume all connections are NLOS. Then, the measurement model of (4.14) can be rewritten as

$$r_i = d_i + b_i + n_i, \quad i \in \mathcal{A}, \tag{4.20}$$

Squaring both sides of (4.20) yields

$$\begin{aligned} r_i^2 &= (d_i + b_i + n_i)^2, \\ &= d_i^2 + c_i + 2n_i(b_i + d_i) + n_i^2 \end{aligned} \tag{4.21}$$

where $c_i = b_i^2 + 2b_i d_i$. For sufficiently small measurement noise, the quadratic form of the noise n_i^2 can be neglected, and we have

$$r_i^2 \simeq d_i^2 + c_i + \epsilon_i \tag{4.22}$$

where $\epsilon_i = 2\bar{d}_i n_i$ and

$$\bar{d}_i = b_i + d_i. \tag{4.23}$$

The nonlinear least squares estimator of the unknown parameters $[\mathbf{x}; \mathbf{c}]$ is obtained as [10, Ch. 8]

$$\begin{aligned}[\hat{\mathbf{x}}; \hat{\mathbf{c}}] = \underset{[\mathbf{x};\mathbf{c}]\in\mathbb{R}^{2+M}}{\arg\min} & \sum_{i\in\mathcal{A}} v_i \left(r_i^2 - d_i^2 - c_i\right)^2, \\ \text{s. t. } & c_i = b_i^2 + 2b_i d_i \end{aligned} \tag{4.24}$$

where $v_i = (\bar{d}_i^2\sigma_i^2)^{-1}$ is the weighting element which is equal to the inverse of the variance of ϵ_i in (4.22). It should be noted that v_i depends on the unknown parameter $\bar{d}_i^2$. However, it can be approximated using r_i^2 instead of $\bar{d}_i^2$ as $v_i \simeq (r_i^2\sigma_i^2)^{-1}$. Unfortunately, the cost function of the minimization problem in (4.24) is still nonlinear and nonconvex. Next, we will convert the problem in (4.24) into a convex optimization problem. However, prior to that, we will impose another constraint on the estimate of the source node location which will considerably increase the performance, especially in severe NLOS environments [52]. Since the biases of NLOS connections in (4.14) are positive and typically much larger than the measurement noise, we can write [52, 51]

$$r_i \geq d_i = \|\mathbf{y}_i - \mathbf{x}\|. \tag{4.25}$$

Equivalently,

$$r_i^2 \geq d_i^2 = (\mathbf{y}_i - \mathbf{x})^T(\mathbf{y}_i - \mathbf{x}). \tag{4.26}$$

The constraint in (4.26) forces the estimated position of the source node to be within the circles determined by the anchor nodes and the radii r_i. However, for LOS connections, the constraint in (4.26) may not be valid, because LOS connections do not have the large NLOS biases and also the measurement noises n_i may have negative values. Therefore, the position of the source node may be located outside of a circle defined by a LOS connection. Since we do not know which connections are LOS, we have to make the constraint in (4.26) slightly looser for all connections. About 95% of a Gaussian random variable's probability density lies within two standard deviations of its mean. Hence, the constraint in (4.26) can be modified as

$$r_i^2 + u_i \geq d_i^2 = (\mathbf{y}_i - \mathbf{x})^T(\mathbf{y}_i - \mathbf{x}) \tag{4.27}$$

where $u_i = 2\sqrt{\mathrm{Var}[r_i^2]}$. To progress, we need to calculate the variance of r_i^2. Assuming d_i and b_i are the unknown deterministic variables, we can rewrite (4.20) as

$$r_i = \bar{d}_i + n_i \tag{4.28}$$

where $\bar{d}_i$ is defined in (4.23). Therefore, we have

$$\begin{aligned} r_i^2 &= (\bar{d}_i + n_i)^2 \\ &= \bar{d}_i^2 + 2\bar{d}_i n_i + n_i^2. \end{aligned} \tag{4.29}$$

Hence, the variance of r_i^2 is calculated as

$$\begin{aligned} \mathrm{Var}[r_i^2] &= \mathrm{E}[r_i^4] - (\mathrm{E}[r_i^2])^2 \\ &= 4\bar{d}_i^2\sigma_i^2 \end{aligned} \tag{4.30}$$

Table 4.1: A summary of the considered estimators.

Estimator	Description
SDP-M	The proposed SDP estimator in (4.31)
ML-LOS	The ML estimator in (4.18) which uses only LOS connections, initialized with true values
ML	The ML estimator in (4.18) which uses all connections, initialized with true values
HUBER	The Huber estimator in [55], initialized with true values
SDP-H	A semidefinite programming estimator in [62]
QP	A quadratic programming estimator in [60]
LMS	A least median squares estimator in [56]
POCS	A projection onto convex sets estimator in [64]

where $\mathrm{E}[\cdot]$ denotes the expected value. Again, since the variance of r_i^2 depends on the unknown parameter d_i^2, we need to approximate the variance by using r_i^2 instead. One of the most useful advantages of the SDP estimator is that one can simply apply constraints to the solution [15]. Relaxing the cost function in (4.24) [15, 52] and using the constraint in (4.26), we express the following convex optimization problem as [15, 27]

$$\begin{aligned}
\underset{\mathbf{x},z,h_i,c_i}{\text{minimize}} \quad & \sum_{i\in\mathcal{A}} v_i(r_i^2 - h_i - c_i)^2 + \delta \sum_{i\in\mathcal{A}} c_i^2 \\
\text{subject to} \quad & h_i = \begin{bmatrix} \mathbf{y}_i \\ -1 \end{bmatrix}^T \begin{bmatrix} \mathbf{I}_2 & \mathbf{x} \\ \mathbf{x}^T & z \end{bmatrix} \begin{bmatrix} \mathbf{y}_i \\ -1 \end{bmatrix}, \\
& \begin{bmatrix} \mathbf{I}_2 & (\mathbf{y}_i - \mathbf{x}) \\ (\mathbf{y}_i - \mathbf{x})^T & r_i^2 + u_i \end{bmatrix} \succeq \mathbf{0}_3, \\
& \begin{bmatrix} \mathbf{I}_2 & \mathbf{x} \\ \mathbf{x}^T & z \end{bmatrix} \succeq \mathbf{0}_3, \quad c_i \geq 0.
\end{aligned} \tag{4.31}$$

which is an SDP optimization problem and can be solved effectively with interior point methods [15], [16, Ch. 11]. δ is a penalization factor which is required when the problem is ill-posed [16, Ch. 6]. It also constrains c_i to a reasonable size [16, Ch. 6]. Note that in (4.31), we neglect the relation of d_i and b_i in c_i which is defined in (4.21) and estimate c_i as an independent unknown parameter.

It is worth mentioning that in the proposed SDP formulation in (4.31), when most of the connections are LOS, the cost function plays an important role and the constraint in (4.26) barely has an effect on the solution. However, when most of the connections are NLOS, the constraint is more effective and keeps the solution from exhibiting a large estimation error.

4.3.4 Simulation Results

In this section, the performance of the proposed estimator is evaluated through computer simulations. A network with eight anchor nodes and one source node is considered. The locations of the anchor nodes are fixed and 441 different locations for the source node are generated uniformly in a square region of 10m $\times$ 10m. Fig. 4.5 shows the configuration of the simulated network. The TOA measurements were generated according to the model given in (4.14). The values of the path loss

exponent for LOS paths β_L and NLOS paths β_N were set to 2 and 3, respectively. The value of α was set to 0.04. The average range measurement error for the network was 0.8 m. The NLOS bias errors were exponentially distributed and their means were randomly selected from 2 to 5 m. The ML estimator in (4.18) was solved by the MATLAB routine `lsqnonlin` with the default setting using the Levenberg-Marquardt method. The proposed SDP estimator was implemented with the `CVX` toolbox [44] using SeDuMi as a solver [17]. The value of δ for the proposed SDP estimator was set to 0.1. In addition to the proposed estimator, five previously studied estimators were selected for comparison. A robust estimator based on the Huber function was studied in [55]. Similar to the ML estimator, the estimator in [55] is highly nonlinear and nonconvex and requires initialization. In our simulation, the estimator in [55] was solved by the MATLAB routine `fminunc` and initialized with the true values, meaning that we provide very optimistic estimates of its performance. Another robust NLOS mitigation technique based on the LMS technique given in [56] is also used for comparison. The estimator in [56] uses linear least squares which does not require initialization. Further, we included the robust method of projection onto convex sets (POCS) introduced in [64] in the simulations. POCS requires simple computations and shows strong robustness against NLOS propagation. An estimator based on quadratic programming (QP) was derived in [60]. The estimator in [60] does not have a closed-form solution. In our simulations, the estimator in [60] was solved by the MATLAB routine `quadprog`. Another estimator included in the comparisons is a recently developed SDP estimator in [62]. The proposed SDP estimator and the SDP estimator in [62] are different in two aspects. First, the proposed SDP estimates NLOS biases along with the source location, while the SDP in [62] only estimates the source location. Second, an upper bound is used as a constraint in the proposed SDP estimator, while the SDP in [62] is forced to find the source location in the middle between an upper and a lower bound. A summary of the considered estimators is given in Table 4.1. The considered estimators are compared in three different NLOS environments; mild, moderate, and severe, where 25%, 62%, and 87% of the connections are NLOS, respectively.

Fig. 4.6 shows the cumulative distribution function (CDF) of the location errors for the considered estimators. Each source node is connected to 8 anchor nodes, as shown in Fig. 4.5. In Fig. 4.6a, 2 out of 8 connections are NLOS. As expected, the ML estimator using only LOS connections (labeled as ML-LOS) demonstrates superior performance. However, it should be noted that we only plot ML-LOS as a benchmark for comparison, since we assume that the estimators do not know which connections are NLOS. Achieving this performance requires NLOS connections to be identified perfectly and the ML solver to be initialized appropriately. The ML estimator assuming all links are LOS (labeled as ML) has the worst performance, because no NLOS mitigation technique was used. It is obvious that NLOS propagation degrades the accuracy of localization considerably, even in mild NLOS environments. The performance of the proposed SDP estimator (labeled as SDP-M) is very close to that of ML-LOS, although unlike ML-LOS, SDP-M does not know which connections are NLOS. HUBER and LMS also perform well in comparison with ML. The performances of the POCS and QP estimators are almost the same and fall between the ML

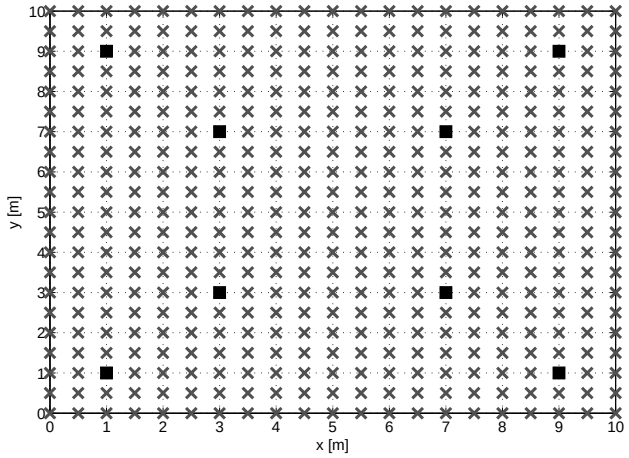
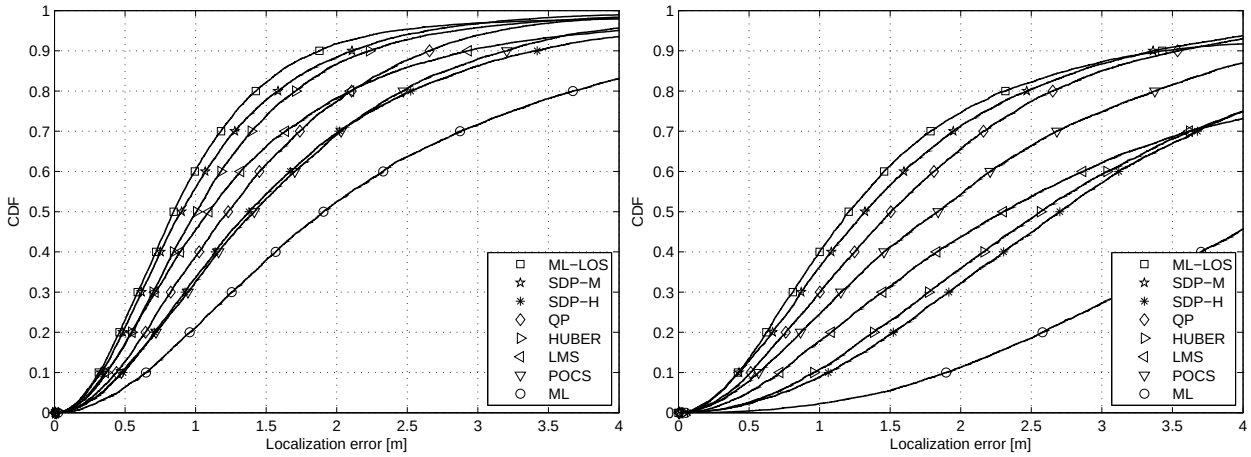


Figure 4.5: The configuration of the simulated network. The solid squares indicate the anchor nodes and the crosses indicate the locations of the source node. There are 8 fixed anchor nodes and 441 different locations for the source node.

and LMS curves.

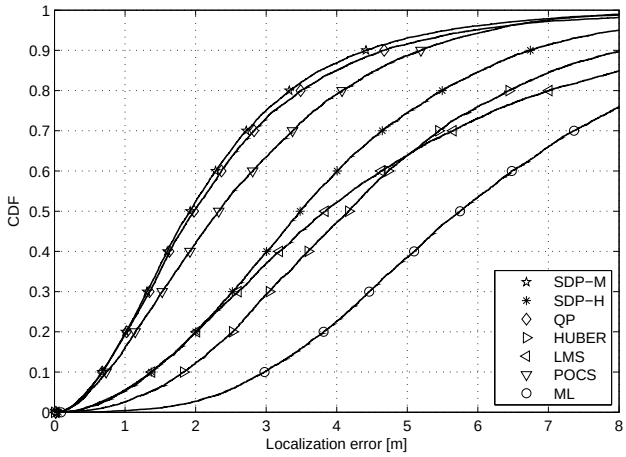
In Fig. 4.6b, the number of NLOS connections is increased to 5. Hence, there are only 3 LOS connections. Again, ML-LOS provides the best accuracy. On the other hand, ML performs very poorly, mainly because most of the connections are NLOS. SDP-M outperforms the other estimators and is still fairly close to ML-LOS. HUBER and LMS cannot tolerate many NLOS errors and degrade substantially. It is worth mentioning that the performances of QP and POCS were worse than those of HUBER and LMS in mild NLOS environments, as depicted in Fig. 4.6a. However, in moderate NLOS environments, QP and POCS perform better than HUBER and LMS. It is well-known that POCS is not particularly accurate in LOS environments, but it is robust to NLOS measurements [64]. In both situations, SDP-H does not provide acceptable performance.

In indoor environments, there are many obstacles between the source node and the anchor nodes. Therefore, most connections are NLOS. Moreover, the number of LOS connections is sometimes fewer than three. In this case, the ML estimator in (4.18) (ML-LOS) does not have a unique solution and, depending on the starting point, the solver of (4.18) returns different estimates for the source location which may have large estimation errors [22]. Unlike the ML estimator, the proposed SDP estimator requires no initialization and always has a unique solution [52]. In Fig. 4.6c, we increase the number of NLOS connections to 7, meaning that there is only one LOS link. Since ML-LOS does not have a unique solution in this case, we did not plot its performance. QP and POCS estimators show strong robustness in severe NLOS environments. LMS and HUBER perform poorly when the number of NLOS connections is very high. The reason is that the LMS estimator in [56] searches for a subset of anchor nodes which has the lowest residual error. Since the number of LOS connections are fewer than three, it cannot find a subset that does not have at least one NLOS connection, which is sufficient to degrade its performance considerably. HUBER generally moderates outliers and large measurements errors [55], but when most of the measurements have large errors, it cannot moderate outliers and degrades substantially. On the other hand, the NLOS



(a) Mild, LOS = 6, NLOS = 2

(b) Moderate, LOS = 3, NLOS = 5



(c) Severe, LOS = 1, NLOS = 7

Figure 4.6: The location error CDF of the considered estimators. HUBER and LMS perform well when the number of NLOS connections is low. QP and POCS perform well when the number of NLOS is moderate. However, the proposed SDP estimator outperforms other estimators in all situations. When the number of NLOS is high, both SDP-M and QP provide the best accuracy, while in this case, ML-LOS is not applicable.

mitigation technique in SDP-M is different from HUBER and LMS. In SDP-M, we assume that all links are NLOS and estimate the NLOS biases for all of them. Therefore, no matter how many measurements have a NLOS bias, the estimator tries to estimate NLOS biases jointly with the source location and find a solution that has the lowest cost function value.

In Table 4.2, the computational complexity of the considered estimators in terms of average running time is compared. The same network as in Fig 4.5 was considered when the number of NLOS connections is 5. SDP-M requires a longer running time than ML-LOS and ML. This is the cost paid for significantly better accuracy and avoiding the need for initialization. However, it should be noted that the ML and ML-LOS solvers are initialized with the true values which reduces their running times. Although the running time of SDP-M is higher than that of ML-LOS, the performance of ML-LOS strongly depends on perfect identification of NLOS connections (which

Table 4.2: The average running time of the considered estimators.

Estimator	Time [ms]
SDP-M	68.89
ML-LOS	21.65
ML	24.53
HUBER	27.77
SDP-H	63.64
LMS	29.36
POCS	26.50
QP	23.69

itself requires extra preprocessing time) and appropriate initialization. The running time of the other estimators falls between ML-LOS and SDP-M.

4.3.5 Conclusion

TOA-based sensor localization in NLOS environments was examined. NLOS connections add a large positive bias to the measurements which usually causes large estimation error. We examined NLOS propagation under two different scenarios. In the first scenario, we assumed that the estimator is able to distinguish NLOS and LOS connections. We showed that discarding NLOS connections is not always a good approach if the estimator is able to identify them. A novel SDP technique was introduced in which NLOS connections are used to improve the accuracy of location estimation. Simulation results showed that the performance of the proposed algorithm approaches optimal accuracy in most conditions. The proposed algorithm is also immune to large estimation errors. Unlike the ML estimator, the proposed SDP is also applicable in degenerate cases when the number of LOS connections is less than three.

In the second scenario, we neither assumed that the estimator had information about the statistics of the NLOS distribution nor did we assume knowledge of which connections are LOS and which are NLOS. In this case, the ML estimator which uses only the measurements from LOS connections provides a lower bound for estimation error if there are a sufficient number of LOS connections available. In this work, we derived a novel SDP technique in order to mitigate the estimation errors introduced due to large NLOS biases. The performance of the proposed SDP estimator was compared with that of the ML estimator and several previously considered algorithms through computer simulations. The algorithms were compared in mild, moderate, and severe NLOS environments. Simulation results showed that the proposed SDP estimator outperforms the previously considered algorithms in all NLOS situations at the cost of greater processing time.

Chapter 5

Cooperative Joint Synchronization and Localization

5.1 Introduction

Synchronization and localization are two main components of WSNs [1, 65]. Many operations in a WSN such as power management, data fusion, spectrum allocation, and localization are dependent on proper synchronization among nodes [66]. Within the body of WSN research lies significant work pertaining to the accurate positioning of the individual sensor nodes. This is due in no small part to the fact that in order to apply meaning to the sensor measurements, it is important to know at which locations the measurements were obtained.

Generally, WSNs are viewed as networks consisting of both anchor nodes, whose positions are known, and source nodes, whose positions must be estimated. While it may be reasonable to assume that the anchor nodes' clocks are synchronized with each other, it is typically not reasonable to assume that the source nodes are synchronized with each other or the anchor nodes. The lack of synchronization among nodes in a WSN is mainly due to their different clock parameters (i.e., clock skew and offset). Since sensors have different hardware implementations, each sensor has its own specific clock. Moreover, the clocks inside sensors are highly sensitive and their parameters may change over time.

In WSNs, clock synchronization is typically performed by transferring a series of time stamp messages among the nodes, called TOA measurements. Moreover, the accuracy of clock synchronization impacts the localization performance in the network. Small differences between the clocks at the nodes may lead to significant localization errors. The reason is that the location of a source node is estimated from a series of time stamps which are a function of the source node clock parameters. If exact information about the clock parameters is not available, the measured time stamps will deviate from their true values leading to significant errors in location estimation. Typically in asynchronous networks, the clocks at the nodes are first synchronized [65, 66, 67] and then localization is performed [2, 43]. However, this approach can lead to poor synchronization

performance which dramatically impacts localization accuracy [68]. Recently, several studies have focused on joint synchronization and localization in a *noncooperative* network, since there is a close relationship between the two and they can be performed simultaneously. It has been shown that joint synchronization and localization can provide significant improvements over the two-step approach [68, 69]. This is mainly due to the fact that in the two-step approach, each link between a source node and anchor node is treated independently, while in the joint approach the relationship between multiple links (which is the location of the source node) is used in the estimation.

The ML estimator of joint synchronization and localization in noncooperative networks is derived in [68]. The ML estimator is a well-known estimator because it provides asymptotically optimal accuracy [68, 69]. A problem which arises, however, is that the ML estimator has a severely nonlinear and nonconvex cost function, and thus a closed-form solution does not exist. In the case of localization alone, it can be solved approximately by iterative methods which require appropriate initializations (c.f., [22, 23, 2]). In the localization-only problem, several approaches have been proposed to deal with this limitation such as linear approximations [70, 19] and convex relaxation techniques [24, 21, 71]. Linear estimators, which have closed-form solutions, are desirable due to their low complexities. However, their performance is very poor when the signal-to-noise ratio (SNR) is low. Several linear estimators have also been proposed for joint synchronization and localization in *noncooperative networks* in the literature [68, 72, 73, 69]. Convex relaxation techniques such as semidefinite programming (SDP) provide another approach for handling the ML convergence problem. Using this technique, the nonlinear and nonconvex ML problem is relaxed into a convex optimization problem. The advantage of SDP problems is that the cost function has no local minima or saddle points and thus convergence to the global minimum is guaranteed [15, 16].

All of the studies mentioned above have only considered a *noncooperative network*. However, cooperation among nodes in WSNs provides many advantages in terms of efficiency and performance [1]. The problem of joint synchronization and localization for cooperative networks has not yet been examined in the literature. Therefore in this work, joint synchronization and localization for *cooperative networks* is studied for the first time. Synchronization and localization for all unlocalized source nodes are performed simultaneously, which is expected to provide better accuracy. Previous works have proposed linear solutions as simple, but less accurate approximations to the ML estimator for noncooperative networks. Unfortunately, such linear approximations can not be extended to the cooperative case. Thus, we propose a nonlinear but convex estimator for this problem. It will be shown that the proposed estimator which is in the form of an SDP has performance very close to the ML, with substantially lower complexity. To make the expressions easier to understand, the problem is first introduced and solved for noncooperative networks. The system model of joint localization and synchronization is introduced and the corresponding ML estimator is formulated. A novel SDP estimator is derived by relaxing the nonconvex and nonlinear ML estimator into a convex one. The CRLB of the system model is also derived as a benchmark for comparing the performance of the proposed estimator. It is shown that the SDP-based estimator

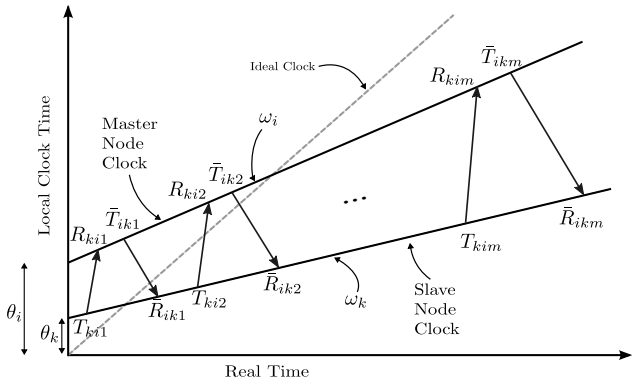


Figure 5.1: Synchronization messages between a master and slave node for the two-way technique.

provides better performance than previously proposed suboptimal estimators for the noncooperative case. The system model and the proposed SDP estimator are then extended to cooperative networks. Computer simulations are conducted to evaluate the performance of the proposed SDP estimator and it is found that cooperation provides a substantial benefit in terms of performance as compared to noncooperative networks. A complexity analysis of the proposed estimator is also provided where it is found that running times are roughly an order of magnitude lower than the ML estimator.

5.2 System Model

In this section, the system model of joint TOA-based synchronization and localization is introduced. The model is first described for a general case of two nodes transmitting synchronization messages. Then, the system models of the noncooperative and cooperative networks including multiple nodes are presented in the subsequent sections.

The internal clocks of the nodes are assumed to be imperfect, a trait which causes the internal time to drift away from the reference time. The relationship between the internal time of a node and the reference time is modeled as [66, 68, 65]

$$t_i = \omega_i t + \theta_i \tag{5.1}$$

where t_i and t are the internal time of the ith node and the reference time, respectively. ω_i is the *clock skew* and θ_i represents the *clock offset*. The lack of synchronization at node may be attributed to the fact that each node has a unique clock, meaning that the clock skew and offset are different for each node.

Consider a slave node k at coordinates $\mathbf{x}_k = [x_k, y_k]^{\mathrm{T}} \in \mathbb{R}^2$ and a master node i at coordinates $\mathbf{x}_i = [x_i, y_i]^{\mathrm{T}} \in \mathbb{R}^2$. The slave node refers to the node that initiates the transmission and wants to synchronize its own clock with that of the master node. There are two common clock synchronization techniques in WSNs: one-way message dissemination and two-way message exchanges [66]. In the one-way message dissemination technique, either the slave node or the master node transmits

the synchronization messages, while in the two-way technique, both slave and master nodes transmit the synchronization messages, as shown in Fig. 5.1. Several rounds of the messages are usually transferred between the nodes to achieve higher accuracy. At the mth round of transmission, the slave node k transmits the forward signal at time stamp T_{kim} and master node i receives the signal at time stamp R_{kim}. Master node i then sends back a signal at $\bar{T}_{ikm}$ and slave node k captures the backward signal at $\bar{R}_{ikm}$. Time stamps T_{kim} and $\bar{R}_{ikm}$ are reported based on the internal clock of node k, while R_{kim} and $\bar{T}_{ikm}$ are reported based on the internal clock of node i. The measured time stamps at the receivers are modeled as

$$\begin{aligned} R_{kim} &= \frac{\omega_i}{\omega_k} T_{kim} + \omega_i (t_{ki} + n_{kim}) - \frac{\omega_i}{\omega_k}\theta_k + \theta_i \\ \bar{R}_{ikm} &= \frac{\omega_k}{\omega_i} \bar{T}_{ikm} + \omega_k (t_{ik} + \bar{n}_{ikm}) - \frac{\omega_k}{\omega_i}\theta_i + \theta_k \end{aligned} \tag{5.2}$$

where $t_{ki} = t_{ik}$ is the propagation time between nodes k and i

$$\begin{aligned} t_{ki} &= \frac{1}{c} d_{ki} \\ &= \frac{1}{c} \sqrt{(\mathbf{x}_k - \mathbf{x}_i)^{\mathrm{T}} (\mathbf{x}_k - \mathbf{x}_i)}. \end{aligned} \tag{5.3}$$

The terms n_{kim} and $\bar{n}_{ikm}$ represent the measurement errors which are modeled as independent and identically distributed (i.i.d.) Gaussian random variables with variance σ_{ki}^2 and $\bar{\sigma}_{ik}^2$, respectively. The measurement errors represent several parameters such as the effect of multipath, the accuracy of the algorithm used for time estimation, the ability to detect the leading edge, and the receiver thermal noise [74]. The CRLB analyses show that the optimal accuracy of the TOA estimate depends on SNR, bandwidth, signal duration, and carrier frequency [75, 1]. The variances of the measurement errors can be modeled as

$$\sigma_{ki}^2 = \xi_i d_{ki}^{\gamma}, \quad \bar{\sigma}_{ik}^2 = \xi_k d_{ik}^{\gamma} \tag{5.4}$$

where ξ_i and ξ_k define the relationship between the noise variance and the true distance and their values depend on the propagation environment and hardware implementations. γ is the path-loss exponent whose value depends on the propagation environment and typically varies between 2 (free space) and 4 (harsh environments) [13]. In most studies, the variances of the measurements are assumed to be the same for all links. However, in this work, each link has a unique variance which is modeled as described above. All four time stamps $\{T_{kim}, R_{kim}, \bar{T}_{ikm}, \bar{R}_{ikm}\}$ are available to the nodes: T_{kim} and $\bar{T}_{ikm}$ are known and predefined, while R_{kim} and $\bar{R}_{ikm}$ are measured at nodes k and i, respectively. Note that the model is based on infinite or at least a very high time resolution of the clock.

In an asynchronous wireless network, we assume that the clock parameters and locations of the anchor nodes are known, while those of the source nodes are unknown. Therefore, the clock parameters and locations of the source nodes must be determined from the synchronization messages

transferred among the nodes. An anchor node is always considered a master node, while a source node is either a master or slave node, depending on whether or not it triggers the transmission.

Sensor networks can be divided into two groups: noncooperative and cooperative [1, 76]. In noncooperative networks, source nodes can only communicate with anchor nodes [77]. In such networks, even communications between two neighboring source nodes is performed through intermediate anchor nodes. Although it is sufficient for a source node to connect to only one anchor node for data transmissions, localization typically requires more anchor nodes. Therefore, a high density of anchor nodes and longer communication ranges are required to guarantee that each source node is connected to a sufficient number of anchor nodes. More specifically, for a synchronous network, each source node must be connected to at least three anchor nodes. The reason is that the problem of positioning includes three degrees of freedom: translation, rotation, and reflection. Therefore, in order to estimate the source node location without ambiguity, at least one connection for each degree of freedom is required. The situation is worse in an asynchronous network where there are two additional degrees of freedom corresponding to the clock offset and skew. In this case, the source node requires five measurements to have a unique solution. Two extra measurements can be provided to the estimator either by adding more connections to the anchor nodes or by adding more rounds of measurements between neighboring nodes. If we are dealing with a dynamic network where the nodes change their locations, the latter option is not possible and the source nodes need to connect to more anchor nodes. Therefore, if a one-way message dissemination technique is used, five anchor nodes are required for each source node, while if two-way message exchange technique is used, four anchor nodes are required. This requirement leads to large anchor node density. To deal with such situations, cooperative networks are introduced, in which source nodes communicate directly with neighboring source nodes in addition to anchor nodes [1, 76, 78]. In this case, the need for dense anchor nodes and longer communication ranges is eliminated. It will be shown later that cooperation among nodes is also highly beneficial for the accuracy of localization.

5.3 Noncooperative Localization

This section describes the system model for noncooperative asynchronous TOA-based synchronization and localization model. In noncooperative networks, the source nodes only communicate with the anchor nodes. Since source nodes do not collect measurements from each other in this case, the location of each source node can be determined independently by using only source-anchor measurements. Let us consider a network with M anchor nodes. To simplify the expressions, assume there is only one source node in the network. Since the source nodes are treated independently, all expressions can be easily developed for multiple source nodes. Let $\mathbf{x} = [x, y]^T \in \mathbb{R}^2$, ω_x, and θ_x be the unknown coordinates and the clock parameters of the source node, respectively. Let $\mathbf{y}_i = [x_i, y_i]^T \in \mathbb{R}^2$, $i = 1, \ldots, M$, ω_i, and θ_i be the known coordinates and clock parameters of the ith anchor node connected to the source node, respectively. From (5.2), the system model of a

noncooperative network can be written as

$$
\begin{aligned}
R_{im} &= \frac{\omega_i}{\omega_x} T_{im} + \omega_i (t_i + n_{im}) - \frac{\omega_i}{\omega_x} \theta_x + \theta_i \\
\bar{R}_{im} &= \frac{\omega_x}{\omega_i} \bar{T}_{im} + \omega_x (t_i + \bar{n}_{im}) - \frac{\omega_x}{\omega_i} \theta_i + \theta_x
\end{aligned} \tag{5.5}
$$

where the subscript i indexes the ith anchor and the subscript x refers to the source node. Note that m refers to the mth round of measurements. It is assumed that during L rounds of the measurements the clock parameters and location of the source node do not change. To further simplify (5.5), L rounds of the message exchanges can be averaged as

$$
\begin{aligned}
R_i &= \frac{\omega_i}{\omega_x} T_i + \omega_i (t_i + n_i) - \frac{\omega_i}{\omega_x} \theta_x + \theta_i \\
\bar{R}_i &= \frac{\omega_x}{\omega_i} \bar{T}_i + \omega_x (t_i + \bar{n}_i) - \frac{\omega_x}{\omega_i} \theta_i + \theta_x
\end{aligned} \tag{5.6}
$$

where

$$
\begin{aligned}
T_i &= \frac{1}{L} \sum_{m=1}^{L} T_{im}, \quad R_i = \frac{1}{L} \sum_{m=1}^{L} R_{im} \\
\bar{T}_i &= \frac{1}{L} \sum_{m=1}^{L} \bar{T}_{im}, \quad \bar{R}_i = \frac{1}{L} \sum_{m=1}^{L} \bar{R}_{im}
\end{aligned} \tag{5.7}
$$

and n_i and $\bar{n}_i$ are i.i.d. Gaussian random variables with variance σ_i^2/L and $\bar{\sigma}_i^2/L$, respectively. Note that the second subscript is now unnecessary and dropped. Note that collecting more rounds of the measurements decreases the effect of the measurement noise which leads to higher localization and synchronization accuracy. Note that this assumption is valid for static networks in which the locations and clock parameters of source nodes do not change during L rounds of measurements.

5.3.1 Maximum Likelihood Estimator

The ML estimator is a popular estimator because of its attractive properties. The ML estimator is asymptotically efficient meaning that it reaches the CRLB as the number of measurements tends to infinity [10, Ch. 7]. The CRLB provides a lower bound on the variance of any unbiased estimator [10, Ch. 3]. In other words, the CRLB determines the best accuracy that can be achieved by an unbiased estimator and is based on the derivative of the log likelihood function of the measurements with respect to the parameters of interest. The CRLB is typically used as a benchmark for comparing different estimators. The derivation of the CRLB is left to Section 5.5. The ML estimator is obtained by maximizing the likelihood function over the unknown parameters. It can be seen from (5.6) that when the likelihood function is formulated, the unknown parameter ω_x will appear in the variance of the noise measurements. Therefore, applying the ML estimation directly to the model in (5.6) is slightly difficult. According to the invariance property, the ML estimator of transformed parameters can be determined by applying the transformation to the ML estimator

of the original parameters [10, Ch. 7]. Introducing two new variables, the model in (5.6) can be alternatively written as

$$\begin{aligned} n_i &= R_i\beta_i - t_i - T_i\beta_x + \alpha_x - \alpha_i \\ \bar{n}_i &= \bar{R}_i\beta_x - t_i - \bar{T}_i\beta_i - \alpha_x + \alpha_i \end{aligned} \tag{5.8}$$

where $\beta_i = 1/\omega_i$, $\alpha_i = \theta_i/\omega_i$, $\beta_x = 1/\omega_x$, and $\alpha_x = \theta_x/\omega_x$. Let $\varphi_x = [\mathbf{x}^{\mathrm{T}}, \beta_x, \alpha_x]^{\mathrm{T}}$ be the vector unknown parameters to be estimated. Since the distribution of n_i and $\bar{n}_i$ is Gaussian, the ML estimator of the model in (5.8) is simply obtained by the following minimization problem

$$\begin{aligned} \hat{\varphi}_x = \underset{\varphi_x \in \mathbb{R}^4}{\arg\min} \; & \sum_{i=1}^{M} \sigma_i^{-2} \left(R_i\beta_i - t_i - T_i\beta_x + \alpha_x - \alpha_i\right)^2 \\ & + \sum_{i=1}^{M} \bar{\sigma}_i^{-2} \left(\bar{R}_i\beta_x - t_i - \bar{T}_i\beta_i - \alpha_x + \alpha_i\right)^2 \end{aligned} \tag{5.9}$$

where the estimates of clock parameters would be: $\hat{\omega}_x = 1/\hat{\beta}_x$ and $\hat{\theta}_x = \hat{\alpha}_x/\hat{\beta}_x$.

The problem in (5.9) is highly nonlinear and nonconvex and its closed-form solution is not available. However, it can be approximately solved by iterative numerical techniques [10, 14, 22]. Since the cost function is nonconvex, there is no guarantee that the solver converges to the global minimum. The difficulty in finding the solution of the ML estimator leads us to employ suboptimal estimators, such as SDP and linear least squares (LLS) estimators. Several linear estimators for asynchronous noncooperative networks are derived in [73, 69, 79]. A novel SDP estimator is derived in the next section.

5.3.2 Semidefinite Programming

The proposed SDP estimator for noncooperative networks is derived in this section. An SDP problem includes a linear cost function and a series of convex constraints which are defined by positive semidefinite matrices [15]. SDP is a form of convex optimization problems, since its cost function and constraints (which might be nonlinear) are convex. To formulate the proposed SDP estimator for the model in (5.8), the cost function of the ML estimator needs to first be formulated as an affine function of the unknown variables, subject to several nonlinear constraints. Then, the constraints are relaxed into a series of positive semidefinite matrices. Finally, the problem is formulated as an SDP optimization problem. To progress, (5.8) is written in vector form as

$$\begin{aligned} \mathbf{n} &= \mathbf{R}\boldsymbol{\beta}_y - \mathbf{t} - \beta_x\mathbf{T}\mathbf{1}_M + \alpha_x\mathbf{1}_M - \boldsymbol{\alpha}_y \\ \bar{\mathbf{n}} &= \bar{\mathbf{T}}\boldsymbol{\beta}_y - \mathbf{t} + \beta_x\bar{\mathbf{R}}\mathbf{1}_M - \alpha_x\mathbf{1}_M + \boldsymbol{\alpha}_y \end{aligned} \tag{5.10}$$

where

$$\mathbf{T} = \mathrm{diag}\{T_1, \ldots, T_M\}, \quad \mathbf{R} = \mathrm{diag}\{R_1, \ldots, R_M\}$$

$$\begin{aligned}
\bar{\mathbf{T}} &= \mathrm{diag}\{\bar{T}_1, \ldots, \bar{T}_M\}, \quad \bar{\mathbf{R}} = \mathrm{diag}\{\bar{R}_1, \ldots, \bar{R}_M\} \\
\mathbf{n} &= [n_1, \ldots, n_M]^{\mathrm{T}}, \qquad \bar{\mathbf{n}} = [\bar{n}_1, \ldots, \bar{n}_M]^{\mathrm{T}} \\
\boldsymbol{\beta}_y &= [\beta_1, \ldots, \beta_M]^{\mathrm{T}}, \qquad \boldsymbol{\alpha}_y = [\alpha_1, \ldots, \alpha_M]^{\mathrm{T}}.
\end{aligned}$$

Defining a new variable $\mathbf{h} = [\mathbf{t}^{\mathrm{T}}, \beta_x, \alpha_x]^{\mathrm{T}}$, (5.10) is expressed as

$$\begin{aligned}
\mathbf{n} &= \mathbf{b} - \mathbf{A}\mathbf{h} \\
\bar{\mathbf{n}} &= \bar{\mathbf{b}} - \bar{\mathbf{A}}\mathbf{h}
\end{aligned} \tag{5.11}$$

where

$$\begin{aligned}
\mathbf{A} &= \begin{bmatrix} \mathbf{I}_M & \mathbf{T}\mathbf{1}_M & -\mathbf{1}_M \end{bmatrix}, \ \mathbf{b} = \mathbf{R}\boldsymbol{\beta}_y - \boldsymbol{\alpha}_y \\
\bar{\mathbf{A}} &= \begin{bmatrix} \mathbf{I}_M & -\bar{\mathbf{R}}\mathbf{1}_M & \mathbf{1}_M \end{bmatrix}, \ \bar{\mathbf{b}} = \bar{\mathbf{T}}\boldsymbol{\beta}_y + \boldsymbol{\alpha}_y.
\end{aligned}$$

By using (5.11), the cost function of the ML estimator in (5.9) can be alternatively written as

$$\begin{aligned}
\mathbf{n}^{\mathrm{T}}\mathbf{Q}\mathbf{n} + \bar{\mathbf{n}}^{\mathrm{T}}\bar{\mathbf{Q}}\bar{\mathbf{n}} &= \mathrm{Trace}\{\mathbf{Q}\mathbf{n}\mathbf{n}^{\mathrm{T}} + \bar{\mathbf{Q}}\bar{\mathbf{n}}\bar{\mathbf{n}}^{\mathrm{T}}\} \\
&= \mathrm{Trace}\{\mathbf{Q}(\mathbf{A}\mathbf{H}\mathbf{A}^{\mathrm{T}} - 2\mathbf{A}\mathbf{h}\mathbf{b}^{\mathrm{T}} + \mathbf{b}\mathbf{b}^{\mathrm{T}}) \\
&\quad + \bar{\mathbf{Q}}(\bar{\mathbf{A}}\mathbf{H}\bar{\mathbf{A}}^{\mathrm{T}} - 2\bar{\mathbf{A}}\mathbf{h}\bar{\mathbf{b}}^{\mathrm{T}} + \bar{\mathbf{b}}\bar{\mathbf{b}}^{\mathrm{T}})\}
\end{aligned} \tag{5.12}$$

where $\mathbf{Q} = \mathrm{diag}\{\sigma_1^{-2}, \ldots, \sigma_M^{-2}\}$, $\bar{\mathbf{Q}} = \mathrm{diag}\{\bar{\sigma}_1^{-2}, \ldots, \bar{\sigma}_M^{-2}\}$, and $\mathbf{H} = \mathbf{h}\mathbf{h}^{\mathrm{T}}$. The first M diagonal elements of the matrix $\mathbf{H}$ are

$$\begin{aligned}
[\mathbf{H}]_{ii} &= t_i^2 \\
&= \frac{1}{c^2}(\mathbf{x} - \mathbf{y}_i)^{\mathrm{T}}(\mathbf{x} - \mathbf{y}_i) \\
&= \frac{1}{c^2}(\mathbf{x}^{\mathrm{T}}\mathbf{x} - 2\mathbf{y}_i^{\mathrm{T}}\mathbf{x} + \mathbf{y}_i^{\mathrm{T}}\mathbf{y}_i) \\
&= \frac{1}{c^2}(z - 2\mathbf{y}_i^{\mathrm{T}}\mathbf{x} + \mathbf{y}_i^{\mathrm{T}}\mathbf{y}_i), \quad i = 1, 2, \ldots, M,
\end{aligned} \tag{5.13}$$

where $z = \mathbf{x}^{\mathrm{T}}\mathbf{x}$. (5.13) can be alternatively written as

$$[\mathbf{H}]_{ii} = \frac{1}{c^2}\begin{bmatrix} \mathbf{y}_i \\ -1 \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \mathbf{I}_2 & \mathbf{x} \\ \mathbf{x}^{\mathrm{T}} & z \end{bmatrix} \begin{bmatrix} \mathbf{y}_i \\ -1 \end{bmatrix}, \ i = 1, 2, \ldots, M, \tag{5.14}$$

By using the auxiliary variables above, the ML estimator in (5.9) is rewritten as

$$\begin{aligned}
&\underset{\mathbf{x}, z, \mathbf{h}, \mathbf{H}}{\text{minimize}} \ \mathrm{Tr}\{\mathbf{Q}(\mathbf{A}\mathbf{H}\mathbf{A}^{\mathrm{T}} - 2\mathbf{A}\mathbf{h}\mathbf{b}^{\mathrm{T}}) + \bar{\mathbf{Q}}(\bar{\mathbf{A}}\mathbf{H}\bar{\mathbf{A}}^{\mathrm{T}} - 2\bar{\mathbf{A}}\mathbf{h}\bar{\mathbf{b}}^{\mathrm{T}})\} \\
&\text{subject to} [\mathbf{H}]_{ii} = \frac{1}{c^2}\begin{bmatrix} \mathbf{y}_i \\ -1 \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \mathbf{I}_2 & \mathbf{x} \\ \mathbf{x}^{\mathrm{T}} & z \end{bmatrix} \begin{bmatrix} \mathbf{y}_i \\ -1 \end{bmatrix}, i = 1, \ldots, M
\end{aligned}$$

$$\mathbf{H} = \mathbf{h}\mathbf{h}^{\mathrm{T}}, \quad z = \mathbf{x}^{\mathrm{T}}\mathbf{x}. \tag{5.15}$$

The terms $\mathbf{b}\mathbf{b}^{\mathrm{T}}$ and $\bar{\mathbf{b}}\bar{\mathbf{b}}^{\mathrm{T}}$ are removed from the cost function, since they are constant and do not affect the minimization problem. Now, the problem in (5.15) has an affine cost function in terms of the individual elements of the matrix $\mathbf{H}$, the vector $\mathbf{h}$, and the variable z. However, there are still two nonconvex constraints on $\mathbf{H}$ and z which make the whole problem nonconvex. To convert the problem in (5.15) into a convex SDP problem, the nonconvex constraints need to be relaxed. Using the Schur complement [16], the matrix $\mathbf{H}$ and the variable z are relaxed and written as a linear matrix inequality (LMI)

$$\begin{aligned} z = \mathbf{x}^{\mathrm{T}}\mathbf{x} \;\; &\Rightarrow \begin{bmatrix} \mathbf{I}_2 & \mathbf{x} \\ \mathbf{x}^{\mathrm{T}} & z \end{bmatrix} \succeq \mathbf{0}_3 \\ \mathbf{H} = \mathbf{h}\mathbf{h}^{\mathrm{T}} \;\; &\Rightarrow \begin{bmatrix} \mathbf{H} & \mathbf{h} \\ \mathbf{h}^{\mathrm{T}} & 1 \end{bmatrix} \succeq \mathbf{0}_{M+3}. \end{aligned} \tag{5.16}$$

Although the above relaxations may degrade the performance of the estimator, they enable us to convert the nonconvex ML estimator into a convex estimator. Using the above relaxations, the original nonlinear and nonconvex ML estimator can be reformulated as

$$\begin{aligned} &\underset{\mathbf{x},z,\mathbf{h},\mathbf{H}}{\text{minimize}} \; \mathrm{Tr}\{\mathbf{Q}(\mathbf{A}\mathbf{H}\mathbf{A}^{\mathrm{T}} - 2\mathbf{A}\mathbf{h}\mathbf{b}^{\mathrm{T}}) + \bar{\mathbf{Q}}(\bar{\mathbf{A}}\mathbf{H}\bar{\mathbf{A}}^{\mathrm{T}} - 2\bar{\mathbf{A}}\mathbf{h}\bar{\mathbf{b}}^{\mathrm{T}})\} \\ &\text{subject to} [\mathbf{H}]_{ii} = \frac{1}{c^2}\begin{bmatrix} \mathbf{y}_i \\ -1 \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \mathbf{I}_2 & \mathbf{x} \\ \mathbf{x}^{\mathrm{T}} & z \end{bmatrix} \begin{bmatrix} \mathbf{y}_i \\ -1 \end{bmatrix}, i = 1, \ldots, M \\ &\qquad \begin{bmatrix} \mathbf{H} & \mathbf{h} \\ \mathbf{h}^{\mathrm{T}} & 1 \end{bmatrix} \succeq \mathbf{0}_{M+3}, \; \begin{bmatrix} \mathbf{I}_2 & \mathbf{x} \\ \mathbf{x}^{\mathrm{T}} & z \end{bmatrix} \succeq \mathbf{0}_3, \end{aligned} \tag{5.17}$$

which is an SDP optimization problem. The solution of (5.17) can be found with iterative algorithms such as the interior point method [15, 16]. Although an iterative algorithm is employed to solve an SDP problem, the algorithm is self-initialized and requires no initialization from the user. Unlike the ML estimator, the SDP problem in (5.17) is convex which ensures that there is only one global minimum. The most important feature of SDP problems is that they can be solved very effectively in polynomial time [15]. In MATLAB simulations, the standard SDP solvers such as SeDuMi or SDPT3 [17, 18] are used to solve SDP problems. If the equality holds in the constraints in (5.16), the solution of the SDP and ML estimators will be the same.

5.4 Cooperative Localization

This section extends the model described in sections II and III to cooperative networks. In cooperative networks, two or more source nodes are involved. The source nodes can communicate not only with anchor nodes but also with each other [1]. Therefore, two sets of measurements

Table 5.1: Complexities of the considered estimators with full connectivity.

Estimator	Description	Iterations	Complexity per Iteration
SDP	The proposed SDP algorithm in (5.26)	k_1	$O(N^3(M+N-1)^3)$
ML	The ML estimator in (5.23)	k_2	$O(N^4(M+3N)^4)$
LLS I	A linear joint estimator in [68]	1	$O(56NM^2)$
LLS II	A linear joint estimator in [73]	1	$O(64NM^2)$
LLS III	A linear joint estimator in [79]	1	$O(28NM^2)$
SEP	A two-step approach: sync. in [80], local. in [70]	1	$O(8NM(2L^3+L^2+M))$

are collected in the network: source-anchor and source-source measurements. Assume there are N source nodes and M anchor nodes in the network. Denote by $\mathcal{S}=\{1,\ldots,N\}$ the set of indices of the source nodes, by $\mathcal{A}=\{N+1,\ldots,N+M\}$ the set of indices of the anchor nodes, and by $\mathbf{x}_k=[x_k,y_k]\in\mathbb{R}^2$, $k\in\mathcal{S}$, the coordinates of the kth source location. In cooperative networks, connectivity is typically limited, meaning that the source nodes are not connected to all anchor nodes and other source nodes. To consider the connection availability, two sets are defined for each source node as follows

$$\begin{aligned}\mathcal{A}_k &= \{\ i \mid \text{anchor node } i \text{ is connected to source node } k\ \}\\ \mathcal{B}_k &= \{\ i \mid \text{source node } i \text{ is connected to source node } k\ \}\end{aligned} \tag{5.18}$$

where $\mathcal{A}_k$ is the set of the indices of the anchor nodes connected to the kth source node and $\mathcal{B}_k$ is the set of indices of the source nodes connected to the kth source node. The received time stamps for the kth source node can be expressed as

$$\begin{aligned}R_{kim} &= \frac{\omega_i}{\omega_k}T_{kim}+\omega_i(t_{ki}+n_{kim})-\frac{\omega_i}{\omega_k}\theta_k+\theta_i,\\ \bar{R}_{ikm} &= \frac{\omega_k}{\omega_i}\bar{T}_{ikm}+\omega_k(t_{ik}+\bar{n}_{ikm})-\frac{\omega_k}{\omega_i}\theta_i+\theta_k\end{aligned} \tag{5.19}$$

where $i\in\mathcal{D}_k$ and $\mathcal{D}_k=\mathcal{A}_k\cup\mathcal{B}_k$ refers to all nodes (either anchor nodes or other source nodes) connected to the kth source node. Note that $i\in\mathcal{A}_k$ refers to the source-anchor measurements which are the same as the noncooperative case, while $i\in\mathcal{B}_k$ refers to the source-source measurements which are only collected in cooperative networks. Furthermore, recall that the clock skews ω_i and clock offsets θ_i are unknown for source nodes ($i\in\mathcal{S}$) and known for anchor nodes ($i\in\mathcal{A}$). Similar to the noncooperative case, after averaging over L rounds of measurements, the above model can be further simplified as

$$\begin{aligned}R_{ki} &= \frac{\omega_i}{\omega_k}T_{ki}+\omega_i(t_{ki}+n_{ki})-\frac{\omega_i}{\omega_k}\theta_k+\theta_i\\ \bar{R}_{ik} &= \frac{\omega_k}{\omega_i}\bar{T}_{ik}+\omega_k(t_{ik}+\bar{n}_{ik})-\frac{\omega_k}{\omega_i}\theta_i+\theta_k\end{aligned} \tag{5.20}$$

where

$$T_{ki} = \frac{1}{L}\sum_{m=1}^{L} T_{kim}, \quad R_{ki} = \frac{1}{L}\sum_{m=1}^{L} R_{kim}$$
$$\bar{T}_{ik} = \frac{1}{L}\sum_{m=1}^{L} \bar{T}_{ikm}, \quad \bar{R}_{ik} = \frac{1}{L}\sum_{m=1}^{L} \bar{R}_{ikm}$$

and n_{ki} and $\bar{n}_{ik}$ are i.i.d. Gaussian random variables with variance σ_{ki}^2/L and $\bar{\sigma}_{ik}^2/L$, respectively. In most cooperative studies [1, 43, 81], only one link between two source nodes is considered. However in this work, we assume that there are two links between two neighboring source nodes: one from source node k to source node i and one from source node i to source node k.

5.4.1 Maximum Likelihood Estimator

Similar to the noncooperative case, two auxiliary variables are introduced and the model in (5.20) is rewritten as

$$\begin{aligned} n_{ki} &= R_{ki}\beta_i - t_{ki} - T_{ki}\beta_k + \alpha_k - \alpha_i \\ \bar{n}_{ik} &= \bar{R}_{ik}\beta_k - t_{ki} - \bar{T}_{ik}\beta_i - \alpha_k + \alpha_i \end{aligned} \tag{5.21}$$

where $\beta_i = 1/\omega_i$, $\alpha_i = \theta_i/\omega_i$, $\beta_k = 1/\omega_k$, and $\alpha_k = \theta_k/\omega_k$. Let us define a new vector $\boldsymbol{\varphi}$ as

$$\boldsymbol{\varphi} = [\mathbf{x}_1^{\mathrm{T}}, \ldots, \mathbf{x}_N^{\mathrm{T}}, \omega_1, \ldots, \omega_N, \theta_1, \ldots, \theta_N]^{\mathrm{T}} \tag{5.22}$$

which is the vector of unknown parameters to be estimated. The corresponding ML estimator of the system model in (5.19) is obtained by the following optimization problem [10, Ch. 7]

$$\begin{aligned} \hat{\boldsymbol{\varphi}} = \underset{\boldsymbol{\varphi}\in\mathbb{R}^{4N}}{\arg\min} & \sum_{k\in\mathcal{S}}\sum_{i\in\mathcal{D}_k} \sigma_{ki}^{-2}\left(R_{ki}\beta_i - t_{ki} - T_{ki}\beta_k + \alpha_k - \alpha_i\right)^2 \\ & + \sum_{k\in\mathcal{S}}\sum_{i\in\mathcal{D}_k} \bar{\sigma}_{ik}^{-2}\left(\bar{R}_{ik}\beta_k - t_{ki} - \bar{T}_{ik}\beta_i - \alpha_k + \alpha_i\right)^2 \end{aligned} \tag{5.23}$$

where $\mathcal{D}_k = \mathcal{A}_k \cup \mathcal{B}_k$. The same statements mentioned for the ML estimator of noncooperative networks are also valid here. Similar to the noncooperative case, the cost function of the ML estimator in (5.23) is highly nonlinear and nonconvex. Although it is very difficult to express mathematically, our simulations results show that the cost function in this case typically has many more local minima and saddle points than the noncooperative cost function, mainly due to the additional unknown variables involved in the problem [82]. As a result, the solver is much more sensitive to initialization. The proposed SDP estimator for cooperative networks, which is indifferent to initialization, is derived in the next section.

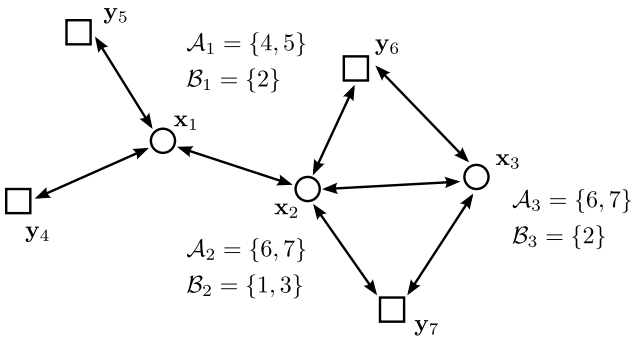


Figure 5.2: An example of a cooperative network.

5.4.2 Semidefinite Programming

Using the same approach as the noncooperative case, the proposed SDP estimator for cooperative networks is derived in this section. First, the system model in (5.21) should be written in vector form

$$\begin{aligned} \mathbf{n}_k &= \mathbf{b}_k - \mathbf{A}_k \mathbf{h}_k \\ \bar{\mathbf{n}}_k &= \bar{\mathbf{b}}_k - \bar{\mathbf{A}}_k \mathbf{h}_k \end{aligned} \tag{5.24}$$

where $\mathbf{h}_k = [\mathbf{t}_k^{\mathrm{T}}, \boldsymbol{\beta}, \boldsymbol{\alpha}]^{\mathrm{T}}$. The rows of matrices $\mathbf{A}_k$ and $\bar{\mathbf{A}}_k$ and the elements of $\mathbf{b}_k$ and $\bar{\mathbf{b}}_k$ are composed of the terms in (5.21). To simplify the derivations, these variables are defined for an example cooperative network. Consider the cooperative network depicted in Fig. 5.2. There are three source nodes and four anchor nodes. The variables in (5.24) for this specific network are defined as

$$\mathbf{A}_1 = \begin{bmatrix} 1 & 0 & 0 & T_{12} & -R_{12} & -1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\mathbf{A}_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & T_{21} & -R_{21} & 0 & -1 & 1 & 0 \\ 0 & 1 & 0 & 0 & T_{23} & 0 & -R_{23} & -1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\mathbf{A}_3 = \begin{bmatrix} 1 & 0 & 0 & T_{32} & -R_{32} & -1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\bar{\mathbf{A}}_1 = \begin{bmatrix} 1 & 0 & 0 & -\bar{R}_{21} & \bar{T}_{21} & 1 & -1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\bar{\mathbf{A}}_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & -\bar{R}_{12} & \bar{T}_{12} & 0 & 1 & -1 & 0 \\ 0 & 1 & 0 & 0 & -\bar{R}_{32} & 0 & \bar{T}_{32} & 1 & 0 & -1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\bar{\mathbf{A}}_3 = \begin{bmatrix} 1 & 0 & 0 & -\bar{R}_{23} & \bar{T}_{23} & 1 & -1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\mathbf{b}_1 = \begin{bmatrix} 0 \\ R_{14}\beta_4 - \alpha_4 \\ R_{15}\beta_5 - \alpha_5 \end{bmatrix}, \ \mathbf{b}_2 = \begin{bmatrix} 0 \\ 0 \\ R_{26}\beta_6 - \alpha_6 \\ R_{27}\beta_7 - \alpha_7 \end{bmatrix}$$

$$\mathbf{b}_3 = \begin{bmatrix} 0 \\ R_{36}\beta_6 - \alpha_6 \\ R_{37}\beta_7 - \alpha_7 \end{bmatrix}, \ \bar{\mathbf{b}}_1 = \begin{bmatrix} 0 \\ \bar{T}_{41}\beta_4 + \alpha_4 \\ \bar{T}_{51}\beta_5 + \alpha_5 \end{bmatrix}$$

$$\bar{\mathbf{b}}_2 = \begin{bmatrix} 0 \\ 0 \\ \bar{T}_{62}\beta_6 + \alpha_6 \\ \bar{T}_{72}\beta_7 + \alpha_7 \end{bmatrix}, \ \bar{\mathbf{b}}_3 = \begin{bmatrix} 0 \\ \bar{T}_{63}\beta_6 + \alpha_6 \\ \bar{T}_{73}\beta_7 + \alpha_7 \end{bmatrix}$$

To progress, another auxiliary matrix needs to be defined as

$$\mathbf{Z} = \mathbf{X}^{\mathrm{T}}\mathbf{X} \tag{5.25}$$

where $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$. Using the same relaxations used in developing (5.16), the nonlinear and nonconvex ML estimator in (5.23) can be expressed as a SDP estimator as follows

$$\begin{aligned} \underset{\mathbf{X},\mathbf{Z},\mathbf{h}_k,\mathbf{H}_k}{\text{minimize}} \ & \sum_{k\in\mathcal{S}} \text{Trace}\{\mathbf{Q}_k(\mathbf{A}_k\mathbf{H}_k\mathbf{A}_k^{\mathrm{T}} - 2\mathbf{A}_k\mathbf{h}_k\mathbf{b}_k^{\mathrm{T}}) + \bar{\mathbf{Q}}_k(\bar{\mathbf{A}}_k\mathbf{H}_k\bar{\mathbf{A}}_k^{\mathrm{T}} - 2\bar{\mathbf{A}}_k\mathbf{h}_k\bar{\mathbf{b}}_k^{\mathrm{T}})\} \\ \text{subject to } [\mathbf{H}_k]_{ii} &= \frac{1}{c^2}\begin{bmatrix} \mathbf{y}_i \\ -\mathbf{e}_k \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \mathbf{I}_2 & \mathbf{X} \\ \mathbf{X}^{\mathrm{T}} & \mathbf{Z} \end{bmatrix} \begin{bmatrix} \mathbf{y}_i \\ -\mathbf{e}_j \end{bmatrix}, \ i \in \mathcal{A}_k, \\ [\mathbf{H}_k]_{ii} &= \frac{1}{c^2}\begin{bmatrix} \mathbf{0}_2 \\ \mathbf{e}_{ik} \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \mathbf{I}_2 & \mathbf{X} \\ \mathbf{X}^{\mathrm{T}} & \mathbf{Z} \end{bmatrix} \begin{bmatrix} \mathbf{0}_2 \\ \mathbf{e}_{ik} \end{bmatrix}, \ i \in \mathcal{B}_k, \\ & \begin{bmatrix} \mathbf{I}_2 & \mathbf{X} \\ \mathbf{X}^{\mathrm{T}} & \mathbf{Z} \end{bmatrix} \succeq \mathbf{0}_{N+2}, \ \begin{bmatrix} \mathbf{H}_k & \mathbf{h}_k \\ \mathbf{h}_k^{\mathrm{T}} & 1 \end{bmatrix} \succeq \mathbf{0}_{M+2N+1}, \end{aligned} \tag{5.26}$$

where $\mathbf{e}_i$ is an $N \times 1$ vector in which the ith element is one and the other elements are zero and $\mathbf{e}_{ik} = \mathbf{e}_i - \mathbf{e}_k$. If the equality holds for the last two constraints in (5.26), the solutions of the proposed SDP estimator and the original ML estimator coincide.

5.5 Cramér-Rao Lower Bound

The CRLB of asynchronous TOA-based localization is derived in this section. Here the CRLB of cooperative networks is derived. However, the expressions can be also used for noncooperative networks. For each source node, four parameters need to be estimated: $x_k, y_k, \omega_k, \theta_k$. Therefore, for a network with N source nodes, $4N$ unknown parameters need to be determined. The CRLB of unknown parameters are calculated from the natural logarithm of the probability density function (log-likelihood function) of the system model. The log-likelihood function of the model in (5.20) is written as

$$\begin{aligned}
&\ln p\left(\{T_{ki}, R_{ki}, \bar{T}_{ik}, \bar{R}_{ik}\}_{k\in\mathcal{S}, i\in\mathcal{D}_k}; \{\mathbf{x}_k, \omega_k, \theta_k\}_{k\in\mathcal{S}}\right) = \\
&- \ln\left(\prod_{k\in\mathcal{S}}\prod_{i\in\mathcal{D}_k}\sqrt{2\pi\omega_i^2\sigma_{ki}^2 2\pi\omega_k^2\bar{\sigma}_{ik}^2}\right) \\
&- \frac{1}{2}\sum_{k\in\mathcal{S}}\sum_{i\in\mathcal{D}_k}\sigma_{ki}^{-2}\left(\frac{R_{ki}}{\omega_i} - t_{ki} - \frac{T_{ki}}{\omega_k} + \frac{\theta_k}{\omega_k} - \frac{\theta_i}{\omega_i}\right)^2 \\
&- \frac{1}{2}\sum_{k\in\mathcal{S}}\sum_{i\in\mathcal{D}_k}\bar{\sigma}_{ik}^{-2}\left(\frac{\bar{R}_{ik}}{\omega_k} - t_{ik} - \frac{\bar{T}_{ik}}{\omega_i} - \frac{\theta_k}{\omega_k} + \frac{\theta_i}{\omega_i}\right)^2.
\end{aligned} \tag{5.27}$$

Let $\boldsymbol{\varphi} = [\mathbf{x}_1^{\mathrm{T}}, \ldots, \mathbf{x}_N^{\mathrm{T}}, \omega_1, \ldots, \omega_N, \theta_1, \ldots, \theta_N]^{\mathrm{T}}$ be the vector of unknown parameters to be estimated. The CRLB of $\boldsymbol{\varphi}$ is the diagonal elements of the inverse of the Fisher information matrix [10, Ch. 3], where it is obtained by

$$[\mathbf{F}]_{ij} = -\mathbb{E}\left[\frac{\partial^2 \ln p}{\partial\varphi_i \partial\varphi_j}\right], \quad i, j = 1, \ldots, 4N. \tag{5.28}$$

The elements of the Fisher information matrix are calculated as follows:

$$\mathbf{F} = \begin{bmatrix} \mathbf{F}_{xx} & \mathbf{F}_{x\omega} & \mathbf{F}_{x\theta} \\ \mathbf{F}_{x\omega}^{\mathrm{T}} & \mathbf{F}_{\omega\omega} & \mathbf{F}_{\omega\theta} \\ \mathbf{F}_{x\theta}^{\mathrm{T}} & \mathbf{F}_{\omega\theta}^{\mathrm{T}} & \mathbf{F}_{\theta\theta} \end{bmatrix}. \tag{5.29}$$

The elements of the matrix $\mathbf{F}$ are provided at the top of the next page.

$$[\mathbf{F}_{xx}]_{kl} = \begin{cases} \displaystyle\sum_{i\in\mathcal{D}_k}(\sigma_{ki}^{-2} + \bar{\sigma}_{ik}^{-2})\mathbf{a}_{ki}\mathbf{a}_{ki}^{\mathrm{T}} + \sum_{i\in\mathcal{S}}\delta_{ik}(\sigma_{ik}^{-2} + \bar{\sigma}_{ki}^{-2})\mathbf{a}_{ik}\mathbf{a}_{ik}^{\mathrm{T}} & k = l \\ -\delta_{kl}(\sigma_{kl}^{-2} + \bar{\sigma}_{lk}^{-2})\mathbf{a}_{kl}\mathbf{a}_{kl}^{\mathrm{T}} - \delta_{lk}(\sigma_{lk}^{-2} + \bar{\sigma}_{kl}^{-2})\mathbf{a}_{lk}\mathbf{a}_{lk}^{\mathrm{T}} & k \neq l \end{cases}$$

$$[\mathbf{F}_{x\omega}]_{kl} = \begin{cases} \displaystyle\sum_{i\in\mathcal{D}_k}(\sigma_{ki}^{-2}\mu_{ki} + \bar{\sigma}_{ik}^{-2}\bar{\lambda}_{ik})\mathbf{a}_{ki} + \sum_{i\in\mathcal{S}}\delta_{ik}(\sigma_{ik}^{-2}\lambda_{ik} + \bar{\sigma}_{ki}^{-2}\bar{\mu}_{ki})\mathbf{a}_{ik} & k = l \\ -\omega_k^{-1}\left[\delta_{kl}\left(\sigma_{kl}^{-2} + \bar{\sigma}_{lk}^{-2}\right)\mathbf{a}_{kl} + \delta_{lk}\left(\sigma_{lk}^{-2} + \bar{\sigma}_{kl}^{-2}\right)\mathbf{a}_{lk}\right] & k \neq l \end{cases}$$

$$[\mathbf{F}_{x\theta}]_{kl} = \begin{cases} \omega_k^{-1} \sum_{i\in\mathcal{D}_k} (-\sigma_{ki}^{-2} + \bar{\sigma}_{ik}^{-2})\mathbf{a}_{ki} + \omega_k^{-1} \sum_{i\in\mathcal{S}} \delta_{ik}(-\sigma_{ik}^{-2} + \bar{\sigma}_{ki}^{-2})\mathbf{a}_{ik} & k = l \\ -\omega_k^{-1} \left[\delta_{kl} \left(-\sigma_{kl}^{-2} + \bar{\sigma}_{lk}^{-2}\right) \mathbf{a}_{kl} + \delta_{lk} \left(-\sigma_{lk}^{-2} + \bar{\sigma}_{kl}^{-2}\right) \mathbf{a}_{lk}\right] & k \neq l \end{cases}$$

$$[\mathbf{F}_{\omega\omega}]_{kl} = \begin{cases} \sum_{i\in\mathcal{D}_k} \sigma_{ki}^{-2}\mu_{ki}^2 + \bar{\sigma}_{ik}^{-2}\bar{\lambda}_{ik}^2 + 2\omega_k^{-2} + \sum_{i\in\mathcal{S}} \delta_{ik}(\sigma_{ik}^{-2}\lambda_{ik}^2 + \bar{\sigma}_{ki}^{-2}\bar{\mu}_{ki}^2 + 2\omega_k^{-2}) & k = l \\ \delta_{kl}(\sigma_{kl}^{-2}\mu_{kl}\lambda_{kl} + \bar{\sigma}_{lk}^2\bar{\mu}_{lk}\bar{\lambda}_{lk}) + \delta_{lk}(\sigma_{lk}^{-2}\mu_{lk}\lambda_{lk} + \bar{\sigma}_{kl}^{-2}\bar{\mu}_{kl}\bar{\lambda}_{kl}) & k \neq l \end{cases}$$

$$[\mathbf{F}_{\theta\theta}]_{kl} = \begin{cases} \omega_k^{-2} \left(\sum_{i\in\mathcal{D}_k} \sigma_{ki}^{-2} + \bar{\sigma}_{ik}^{-2} + \sum_{i\in\mathcal{S}} \delta_{ik}(\sigma_{ik}^{-2} + \bar{\sigma}_{ki}^{-2}) \right) & k = l \\ -\omega_k^{-1}\omega_l^{-1} \left[\delta_{kl} \left(\sigma_{kl}^{-2} + \bar{\sigma}_{lk}^{-2}\right) + \delta_{lk} \left(\sigma_{lk}^{-2} + \bar{\sigma}_{kl}^{-2}\right)\right] & k \neq l \end{cases}$$

$$[\mathbf{F}_{\omega\theta}]_{kl} = \begin{cases} \omega_k^{-1} \sum_{i\in\mathcal{D}_k} \sigma_{ki}^{-2}\mu_{ki} + \bar{\sigma}_{ik}^{-2}\lambda_{ik} + \omega_k^{-1} \sum_{i\in\mathcal{S}} \delta_{ik}(\sigma_{ik}^{-2}\lambda_{ik} + \bar{\sigma}_{ki}^{-2}\bar{\mu}_{ki}) & k = l \\ \omega_k^{-2}\omega_l^{-1}\delta_{kl} \left(\sigma_{kl}^{-2}\mu_{kl} - \omega_k\bar{\sigma}_{lk}^{-2}\lambda_{lk}\right) + \omega_l^{-2}\omega_k^{-1}\delta_{lk} \left(-\omega_l\sigma_{lk}^{-2}\lambda_{lk} + \bar{\sigma}_{kl}^{-2}\bar{\mu}_{kl}\right) & k \neq l \end{cases}$$

We also have

$$\mathbf{a}_{ki} = \begin{bmatrix} \frac{x_k - x_i}{cd_{ki}} & \frac{y_k - y_i}{cd_{ki}} \end{bmatrix}^{\mathrm{T}}, \; \delta_{ki} = \begin{cases} 1 & i \in \mathcal{B}_k \\ 0 & i \notin \mathcal{B}_k \end{cases}$$

$$\mu_{ki} = \omega_k^{-2}(\theta_k - T_{ki}), \qquad \lambda_{ik} = \omega_k^{-1} \left(\frac{d_{ik}}{c} + \frac{T_{ik}}{\omega_i} - \frac{\theta_i}{\omega_i} \right)$$

$$\bar{\mu}_{ki} = \omega_k^{-2}(\theta_k - \bar{T}_{ki}), \qquad \bar{\lambda}_{ik} = \omega_k^{-1} \left(\frac{d_{ik}}{c} + \frac{\bar{T}_{ik}}{\omega_i} - \frac{\theta_i}{\omega_i} \right).$$

Finally, the CRLB of the unknown parameters is obtained by

$$\mathrm{Var}[\varphi_i] = [\mathbf{F}^{-1}]_{ii}, \quad i = 1, 2, \ldots, 4N. \tag{5.30}$$

5.6 Simulation Results

This section discusses a series of computer simulations used to evaluate the performance of the proposed estimator. Both cooperative and noncooperative networks are considered. The ML estimator is solved by the MATLAB routine `lsqnonlin` using the Levenberg-Marquardt method. The solver of the ML estimator is initialized with true values. The proposed SDP estimator is implemented by the `CVX` toolbox [44] using SeDuMi as the solver [17].

5.6.1 Noncooperative

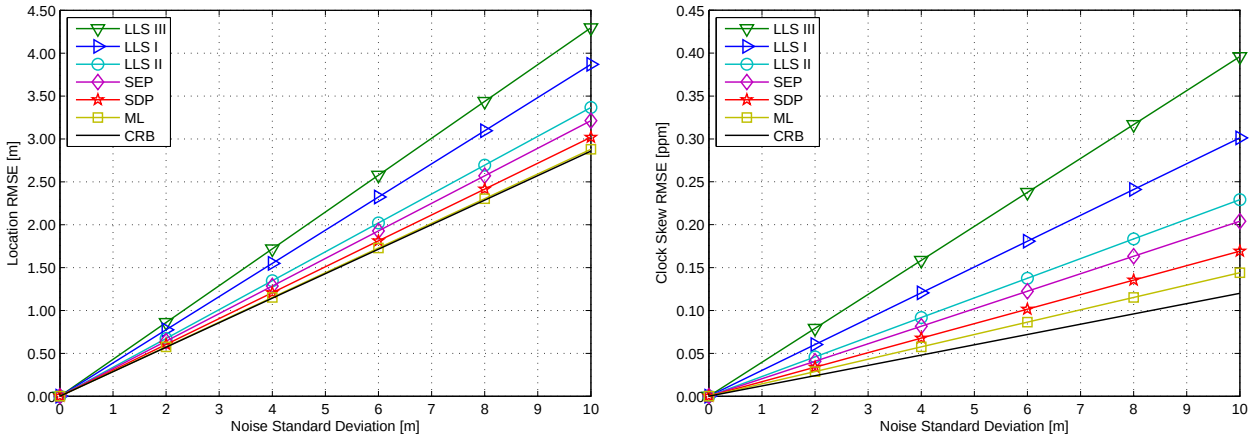
Besides the ML estimator described in the previous sections, three other estimators are selected for comparisons. To show the advantages of joint synchronization and localization, a two-step approach is also included in the simulations. In the two-step technique, synchronization is first performed in the network by using the method described in [80], where the clock parameters of the source

nodes are estimated. The synchronization method in [80] has a closed-form solution and requires no initialization. Secondly, a classic location estimator assuming perfect synchronization is employed to estimate the locations of the source nodes. The well-known linear estimator in [70], which also has a closed-form solution, is selected for localization. Three other linear estimators in [68], [73], and [79] performing joint synchronization and localization are also included in the comparisons. A summary of the considered estimators and their labels is provided in Table 5.1.

A network with 8 anchor nodes and 500 source nodes distributed uniformly in a square area of 50 m $\times$ 50 m is considered. The anchor nodes are placed at $[5,5]^{\mathrm{T}}$, $[5,45]^{\mathrm{T}}$, $[45,45]^{\mathrm{T}}$, $[45,5]^{\mathrm{T}}$, $[15,25]^{\mathrm{T}}$, $[25,15]^{\mathrm{T}}$, $[25,35]^{\mathrm{T}}$, and $[35,25]^{\mathrm{T}}$. In each experiment one source node from the 500 source nodes is selected. The clock offset and skew of the nodes are drawn from a uniform distribution $\mathcal{U}[0, 2\times10^{-6}]$ and $\mathcal{U}[0.95, 1.05]$, respectively. The transmission times of the nodes T_{kim} and $\bar{T}_{ikm}$ are drawn from a uniform distribution $\mathcal{U}[5m, 5m+1]$ and $\mathcal{U}[5m+3, 5m+4]$, respectively. The variances of the measurement noises are determined based on (5.4). The value of the path-loss exponent γ is set to 2 and assumed be the same for all nodes. The value of ξ is indicated in each figure and assumed to be the same for all nodes. The exact covariance matrices of noise are assumed to be available in order to show the optimal accuracy. However, the estimator only uses the variances for weighting and does not consider their dependency on the node positions for estimation. If the estimator does not have the exact variances, all covariance matrices can be set to an identity matrix. Our simulations show that if the exact covariance matrices are not available, the performance of the estimators degrades slightly. However, their *relative* performances do not change. Two rounds of measurements are performed between nodes, i.e., $L = 2$. For each experiment, 100 noise realizations are performed where the measurement noise, the clock parameters, and transmission times are randomly generated.

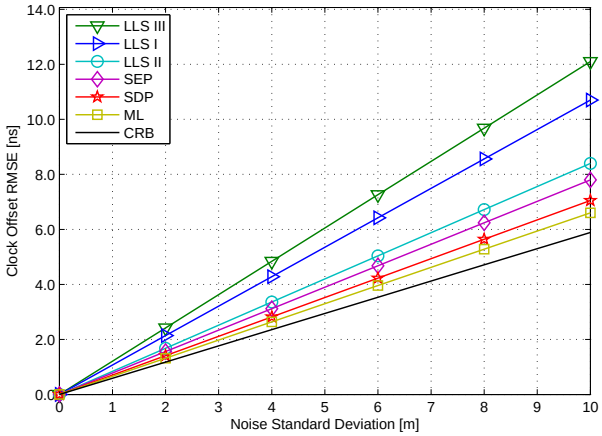
The root-mean-square error (RMSE) of the compared estimators for the location, clock skew, and clock offset is given in Fig. 5.3. The ML estimator, as expected, provides the best performance in all cases, primarily because its solver is initialized with the true values. Among the estimators, LLS I and LLS III exhibit the worst performance, mainly due to several approximations. However, as the measurement noise increases, the performance of the ML estimator degrades. LLS II performs significantly better than other linear estimators, especially in low noise levels. The reason being that, in LSS II, the effect of clock offset is eliminated by using time difference calculations and subsequent linearization is performed over three, rather than four, parameters. Therefore, there are fewer approximations in LLS II than in LLS I and LLS III which can hurt the performance. The performances of all linear estimators degrade significantly as the measurement noise increases, mainly due to the fact that the linearization assumptions are not valid in high noise measurement levels.

In all noise levels, the performance of the proposed SDP estimator is superior to the linear estimators and it performs as well as the ML estimator. The main reason is that unlike in linear estimators, in the proposed SDP estimator the original nonconvex problem is relaxed (not linearized) to a convex problem and the relaxation is tight, even in high SNR levels. Despite the fact



(a) Location

(b) Clock skew



(c) Clock offset

Figure 5.3: The RMSE performance of the considered estimators versus the standard deviation of the ranging noise for the location, clock skew, and clock offset of the source node in a noncooperative network. The proposed SDP estimator outperforms the other estimators. Moreover, it performs as well as the optimal ML estimator. The direct relationship between the synchronization and localization accuracy can be seen. The estimator which provides better accuracy in synchronization has better performance in localization too.

that the SEP estimator can only be employed in the two-way technique, its performance is better than the linear estimators but worse than SDP. The reason for the poor performance of SEP in comparison to ML and SDP is anticipated to be due to separately synchronization and localization. The SEP estimator is therefore a reasonable choice only in low measurement noise levels when a simple closed-form estimator is desired. However, with just slightly more complexity, the proposed SDP estimator provides very accurate performance in most situations.

In Fig. 5.3, the synchronization performance of the considered estimators is also compared. The order of estimators remains unchanged in comparison with the localization performance. The ML estimator accuracy for the clock parameters is slightly far from the CRLB. However, it should be noted that the ML estimator is *asymptotically* efficient, meaning that it can achieve the CRLB when the number of measurements tends to infinity. When the number of measurements is limited,

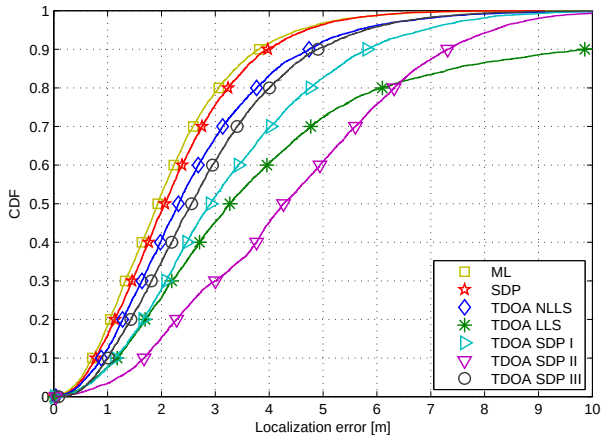


Figure 5.4: The CDF of the localization error comparing the proposed estimator with TDOA estimators in a noncooperative network. The standard deviation of the noise is 8 m.

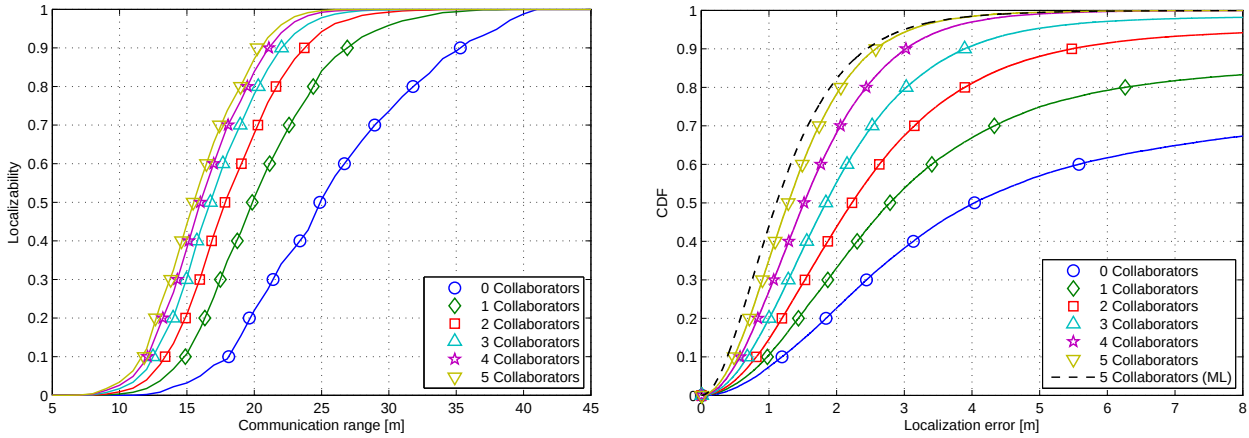
nothing about its optimality can be said. The direct relationship between the synchronization and localization accuracy can be clearly seen. In other words, the accuracy of localization is highly dependent on the accuracy of synchronization. The estimator which provides better accuracy in synchronization has better performance in localization too.

Since in this work the anchor nodes are assumed to be synchronized, one can use a classic TDOA technique [83, 35, 84, 85] to estimate the source node location. However in TDOA, it is assumed that only the clock offset of the source node is unknown, while its clock skew is perfectly available. In this technique, an anchor node is selected as a reference and its time stamp is subtracted from the time stamps of other anchor nodes. In this way, the dependency of the measurements on the source clock offset is removed. Then, the location of source nodes is estimated from TDOA measurements. In this simulation, a comparison is made between the proposed estimator considering unknown clock skew and offset, and several TDOA techniques in the literature: a TDOA nonlinear least square (NLLS) estimator in [85], a well-known linear estimator in [35], and three SDP estimators labeled as TDOA SDP I [84], TDOA SDP II [83], and TDOA SDP III [86]. In Fig. 5.4, the localization performance of the proposed SDP estimator is compared with those of the aforementioned TDOA estimators. Fig. 5.4 shows that the proposed SDP estimator outperforms other estimators (even TDOA NLLS) mainly due to the fact that the unknown clock skew is neglected in TDOA estimators.

A line-of-sight (LOS) path between nodes is assumed in the above simulations. However, the majority of the links in indoor networks are non-line-of-sight (NLOS), since the LOS path between nodes are blocked by objects and walls [87, 52]. Our simulation results show that the performance of all estimators degrades in NLOS environments [86]. However, their relative performances remain almost unchanged.

5.6.2 Cooperative

All the estimators considered in the previous section are designed only for noncooperative networks and are not applicable for cooperative networks. Therefore in this section, only the performance of



(a) Localizability **(b)** Localization Performance

Figure 5.5: The performance of the proposed SDP estimator in a cooperative network. (a) The percentage of the localizable nodes in the network versus the number of collaborators and communication range. (b) The CDF of the localization error of the proposed SDP estimator. The communication range and the standard deviation of the noise is fixed at 25 m and 8 m, respectively. Zero collaborator represent the noncooperative case.

the proposed estimator is evaluated in cooperative networks and compared with the noncooperative one.

An important advantage of cooperative networks is that, under limited connectivity, more source nodes can be localized in cooperative networks than in noncooperative networks. Node localizability refers to a situation in which the location of the source node can be localized uniquely without any ambiguity. In synchronous noncooperative networks, a source node is localizable if it is connected to at least three non-collinear anchor nodes. However in synchronous cooperative networks, a source node is localizable if it has at least three disjoint paths to the non-collinear anchor nodes and it is included in a redundantly rigid network [88]. Therefore, it is not necessary for the source node to be directly connected to an anchor node and connection can be established through intermediate source nodes. As mentioned earlier, in asynchronous networks, the source nodes typically need to have more connections, since there are more unknown variables to be estimated.

To show the advantages of cooperative networks, the communication range in the network has been limited. The source nodes are only connected to nearby nodes. For this simulation, only one round measurement is collected between nodes, hence, each source node requires at least four disjoint paths to the non-collinear anchor nodes to be localized. Fig. 5.5a depicts the localizability percentage for the same network as a function of the number of collaborators and different communication ranges. Zero collaborators represents the noncooperative case. Fig. 5.5a shows that using cooperation among nodes can significantly improve node localizability in the network at a fixed communication range. In other words, source nodes in cooperative networks require shorter communication ranges (equivalently, lower power consumption) to be localized uniquely. For instance, in a network due to power and interference limitation, the nodes can only communicate with other

nodes within their 25 m range. In this case, only 50% of the source nodes can be localized if no cooperation is done, while more than 95% of the source nodes can be localized if each source node communicates with at least two other source nodes. Note that the results in Fig. 5.5a are based on the assumption that measurements are noiseless, since the generalization for noisy measurements is difficult to determine.

In addition to increasing localizability, cooperation between nodes significantly improves localization performance. Fig. 5.5b shows the CDF of the localization error of the proposed SDP estimator. The same network as in previous simulations is considered here. However, the communication range and the standard deviation of the noise is fixed at 25 m and 8 m, respectively. Fig. 5.5a shows that at 25 m communication range only 50% of the source nodes are localizable in a noncooperative network which can be also seen in Fig. 5.5b where the estimator performs poorly in 50% of the time. If the source node collaborates with only one other source node, both localizability and localization performance are improved significantly. Higher improvement can be achieved by increasing the number of collaborators. The performance of the ML estimator for 5 collaborators is also provided in Fig. 5.5b as a benchmark which shows that how close the performance of the proposed SDP estimator is to the optimal accuracy. Note that there are some differences between the results in Fig. 5.5b and Fig. 5.5a. The differences are due to the fact that the localizability is determined merely based on the connectivity among nodes and the measurement noise and geometry are not incorporated. Therefore, there are some situations where a source node is determined to be localizable, while it cannot actually be localized and vice versa.

5.6.3 Complexity Analysis and Running Time

This section provides the computational complexities of the considered estimators. Complexities are compared based on the total number of floating-point operations or *flops*. The details of the complexity analysis for the ML, linear, and SDP estimators can be found in [81]. Table 5.1 summarizes the computational complexities of the estimators for cooperative localization with full connectivity. Note that Table 5.1 provides asymptotic complexities which means that only the dominating elements are presented. The linear estimators (LLS I, LLS II, LLS III, SEP) have lower complexity than the ML and SDP estimators, since they require only one iteration. LLS I, LLS II, and LLS III are two-step, five-step, and one-step estimators, respectively. The higher the number of steps, the higher the complexity is. Note that the complexity of SEP is dependent on the number of measurement rounds, L, since the simple preprocessing average in (5.20) cannot be performed in SEP. The proposed SDP has higher complexity per iteration than the ML estimator, since several auxiliary variables need to be included in the problem to make it convex. However, as will be shown later in the simulation results the overall complexity of the SDP is significantly lower than the ML. There is no exact expression for the number of iterations for the SDP and ML estimators, since it depends on several parameters, such as step size, initial point, and required accuracy. The number of iterations of the SDP estimator is upper bounded by $\sqrt{N(N+M)}\log(1/\epsilon)$, where ϵ is the required accuracy [15, 43]. However, some results show that the number of iterations of the

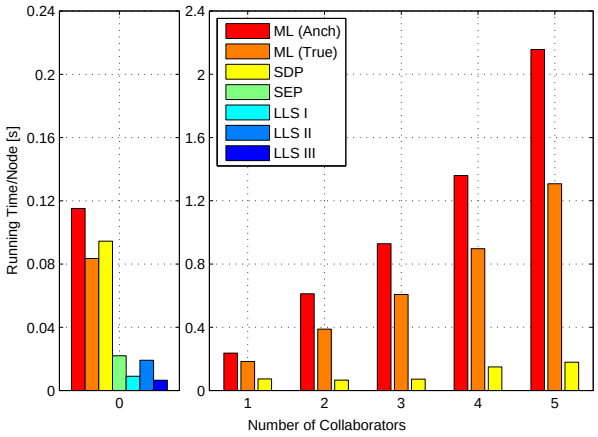


Figure 5.6: Running time per node of the considered estimators versus the number of collaborators. The proposed SDP estimator has significantly lower running time than the ML estimator in cooperative networks. ML (True) and ML (Anch) stand for the ML estimator initialized with the true value and the closest anchor node, respectively.

SDP grows much more slowly with the problem size and in many cases it is shown to be constant [15].

In Fig. 5.6, the average running time per source node as a function of the number of collaborators in both cooperative and noncooperative networks is compared for the considered estimators. As expected, in the noncooperative case the linear estimators have faster running times than the ML estimator and the proposed SDP estimator, mainly because they require only one iteration. When there is only one source node (noncooperative case), the SDP estimator requires slightly more running time than the ML estimator initialized with the true values. One reason why this is the case is that initialization with the true values cuts down the ML estimator running time. On the other hand, the proposed SDP estimator is self-initialized which requires extra computations and calculations. Fig. 5.6 shows that if the ML estimator is initialized with the closest anchor node (which is the practical case), its running time increases dramatically in comparison with the true value initialization. Moreover, as the number of collaborators increases in the network, the complexity and running time of the ML estimator increase rapidly. The proposed SDP estimator, on the other hand, requires lower running time than the ML estimator even when the ML estimator is initialized with the true values.

Fig. 5.6 shows that the SDP running time goes down slightly as the number of collaborators increases from 1 to 3. Solving an SDP problem requires two parts: a modeling part and a processing part. The modeling part which may include several transformations, dual problem formulation, and initialization are not significantly affected by the problem size. However, in the processing part where the algorithm starts updating dual and primal variables the problem size plays an important role and dominates the running time. When the number of collaborators is less than 3, the modeling part dominates and is almost constant. Therefore, when the whole running time is divided by the number of nodes, less running time per node is obtained. However, when there are more than 3 collaborating nodes, the processing time dominates and increases at a faster rate. Thus, the per

node running time increases.

5.7 Conclusion

In this work, the problem of cooperative asynchronous TOA-based sensor localization was examined. The analysis was performed under the assumption that anchor nodes of known location and with perfectly synchronized clocks were available to aid in the estimation of the source node positions. A framework for the joint synchronization and localization problem was presented, and a novel SDP estimator was derived and introduced to solve the problem for both cooperative and noncooperative networks. It was shown that despite requiring significantly less complexity than the ML estimator, especially in cooperative networks, the proposed SDP estimator provides position estimates whose accuracies are on par with the ML estimator in both cooperative and noncooperative cases. Moreover, the SDP estimator does so without requiring any initialization and with guaranteed convergence. In comparison, the ML estimator was initialized with the true source node locations and clock parameters, something which is unreasonable in real scenarios and greatly to the benefit of the ML estimator's performance results. Furthermore, the proposed SDP estimator was compared to previously proposed estimators for noncooperative source localization, and shown to provide superior positioning accuracies. However, it is in the cooperative scenario that the proposed SDP estimator truly shines, by taking advantage of source node cooperation (and the resulting improved network connectivity) to increase both localizability and localization accuracy significantly.

Chapter 6

Cooperative Source Node Tracking

6.1 Introduction

Localization and tracking have found many commercial, civilian, and military applications including tracking aircraft, tracking the behavior of animals within a herd, or positioning of firefighters inside a burning building. The Global Positioning System (GPS) is the most popular and accessible method for localization. GPS can provide a highly accurate estimate of the receiver location in clear view environments where the receiver has a direct view to a sufficient number of satellites. However in indoor and dense environments, the GPS receiver does not receive satellite signals very well, as they are attenuated strongly by walls and objects. As a result, GPS fails to provide a reliable estimate in such environments. Moreover, GPS is costly and bulky, hence not practical for all applications. Alternatively, wireless sensor networks (WSNs) can be employed for indoor environments where localization is performed using measurements captured within the network and without aid of any external resource [89, 1].

WSN localization is divided into two groups: noncooperative and cooperative [1, 81]. In noncooperative networks, source nodes can only communicate with anchor nodes. For data transmission, a source node needs to connect to only one anchor node. However, source node localization typically requires connections to at least three anchor nodes. Therefore, a high density of anchor nodes and longer communication ranges are required to ensure that each source node is connected to a sufficient number of anchor nodes. Moreover, in indoor and dense environments, the majority of connections with the anchor nodes are non-line-of-sight (NLOS) and even if the source node is connected to a sufficient number of anchor nodes, the localization performance is very poor. To deal with such situations, cooperative networks are introduced in which source nodes are able to communicate not only to the anchor nodes but also to each other [76]. In this way, the source node will have more connections leading to higher localizability and localization accuracy [90, 91]. Moreover, the need for dense anchor node deployment is eliminated in such networks.

There are many studies on NLOS mitigation techniques for localization of a static source node in noncooperative networks [48, 87, 55, 92, 57]. However, few studies have focused on source tracking in NLOS environments [93, 94]. The difference between static localization and tracking is that the

source node is assumed to be mobile in the latter and an underlying dynamic model is used to improve the estimation accuracy. One technique to deal with NLOS propagation is to estimate the NLOS biases [95]. In this technique (also called bias tracking), all measurements are assumed to be NLOS with a large positive bias. For each link, an unknown parameter is added to the state vector and estimated jointly with the location and velocity of the source node. We will later show that the performance of this technique is not as good as expected in many NLOS situations. Moreover, the complexity of the algorithm intensifies as the number of measurements increases, since for each measurement we need to track an unknown bias. Further, the complexity increases exponentially for cooperative localization.

Modeling NLOS/LOS time dependencies using a Markov chain has also been proposed in the literature [93, 94]. The interacting multiple model (IMM) algorithm is used to consider the underlying Markov chain. The problem is that the IMM algorithm needs to run 2^M filters for M measurements and hence the number of required filters grows exponentially with the number of measurements [94]. In [93], the authors introduce an approximation to decrease the number of required filters. However, the IMM algorithm still needs to know the transition probabilities of the LOS and NLOS states. The performance of the IMM algorithm is highly dependent on the transition probabilities which are very difficult to determine in practice. Using robust estimators which are very popular for static localization [54, 86] provides another solution for source tracking in NLOS propagation [92]. Unlike classic estimators, robust estimators are typically less vulnerable to outliers and NLOS biases. The main advantage of robust estimators is that they typically require no information about the NLOS propagation.

Although there are a few studies on noncooperative source node tracking in NLOS environments, this problem for cooperative networks has not yet been addressed. Therefore in this work, the problem of source node tracking in NLOS environments for cooperative networks is examined. A NLOS mitigation technique based on semidefinite programming (SDP) for cooperative networks is introduced. It should be noted that SDP techniques have not been previously proposed for the tracking problem in the literature in either noncooperative or cooperative scenarios. We show that tracking NLOS biases is not a good option. Therefore, to mitigate NLOS propagation, the proposed estimator estimates the NLOS biases jointly with the location and velocity of the source node. However, the estimator does not track the biases and instead estimates them independently at each time step. Although the complexity of the estimator increases because of the bias estimation, its complexity is lower than estimators which include the biases in the state vector and track them. The proposed estimator can be classified as a robust estimator and requires no statistical knowledge about the NLOS propagation. To make derivations simple and more clear, the proposed SDP estimator is first derived for a noncooperative network. Then, the problem is extended to cooperative networks and the proposed SDP estimator is developed. The results show that the proposed SDP estimator substantially outperforms the other previously proposed estimators for both noncooperative and cooperative networks.

6.2 Noncooperative Tracking

This section covers the system model of a noncooperative network. The extension of the model to cooperative networks is provided in the next section. Generally speaking, sensor tracking includes a series of anchor nodes with known locations estimating the locations of mobile source nodes from noisy measurements captured within the network. Time-of-arrival (TOA) measurements are highly popular for localization purposes due to simple and inexpensive implementations. However, their performance is strongly affected by NLOS propagation. An NLOS link occurs when the direct view between the source node and anchor node is blocked by an obstruction [48]. In indoor environments, most connections are NLOS because of the presence of many walls and objects.

Consider a network with M anchor nodes and a moving source node. Denote by $\mathbf{y}_i \in \mathbb{R}^2$, $i = 1, 2, \ldots, M$, the location of the ith anchor node and by $\mathbf{x}(t) \in \mathbb{R}^2$ and $\dot{\mathbf{x}}(t) \in \mathbb{R}^2$, $t = 1, 2, \ldots, T$, the location and velocity of the source node at time step t, respectively. Let $\mathbf{u}(t) = [\mathbf{x}^{\mathrm{T}}(t), \dot{\mathbf{x}}^{\mathrm{T}}(t)]^{\mathrm{T}}$ be the state vector at time step t. The tracking problem includes state and measurement models which are expressed as follows [94, 93]

$$\begin{aligned} \mathbf{u}(t) &= \mathbf{A}\mathbf{u}(t-1) + \mathbf{w}(t) \\ \mathbf{z}(t) &= \mathbf{f}(\mathbf{u}(t)) + \mathbf{n}(t). \end{aligned} \tag{6.1}$$

The first expression in (6.1) defines the evolution of the state vector over time and the second one defines the relationship between the measurements and the state vector. The matrix $\mathbf{A}$ is defined as [93]

$$\mathbf{A} = \begin{bmatrix} \mathbf{I}_2 & \Delta\mathbf{I}_2 \\ \mathbf{0}_2 & \mathbf{I}_2 \end{bmatrix}$$

where Δ is the time interval between samples. $\mathbf{w}(t)$ and $\mathbf{n}(t)$ are the process and measurement noises which are modeled as zero-mean Gaussian random variables with covariance matrices of $\mathbf{Q}_w(t)$ and $\mathbf{Q}_n(t)$, respectively. $\mathbf{f}(\mathbf{u}(t)) = [f_1(\mathbf{u}(t)), f_2(\mathbf{u}(t)), \ldots, f_M(\mathbf{u}(t))]^{\mathrm{T}}$ and $f_i(\mathbf{u}(t))$ is a nonlinear function of the state vector and defined based on the type of available measurements. In this work, we assume that only TOA measurements are available and f_i can be modeled as [96, 52]

$$f_i(\mathbf{u}(t)) = \|\mathbf{x}(t) - \mathbf{y}_i\|_2. \tag{6.2}$$

In the definition of the measurement model in (6.2), the NLOS propagation is not included. In the presence of NLOS links, the model in (6.2) can be still used. However, the performance is not as good as expected.

To mitigate the NLOS propagation, the NLOS biases can be also estimated. Since the estimator does not know which connections are NLOS, all measurements are assumed to be NLOS [95, 87]. Hence, the models in (6.1) should be modified as follows [95]

$$\mathbf{s}(t) = \mathbf{B}\mathbf{s}(t-1) + \mathbf{v}(t)$$

$$\mathbf{z}(t) = \mathbf{h}(\mathbf{s}(t)) + \mathbf{n}(t) \tag{6.3}$$

where $\mathbf{s}(t) = [\mathbf{u}^{\mathrm{T}}(t), \mathbf{b}^{\mathrm{T}}(t)]^{\mathrm{T}}$, $\mathbf{b}(t) = [b_1(t), b_2(t), \ldots, b_M(t)]^{\mathrm{T}}$, and $b_i(t)$ are large positive biases due to NLOS propagation. Since $b_i(t)$ are unknown to the estimator, they should be estimated along with the location and velocity of the source node. Moreover, the estimate of the biases can be tracked, which is why we include them in the state vector. Modeling the evolution of the NLOS biases is a difficult task. In the majority of studies which consider bias tracking, the NLOS biases are modeled as a random walk [94]. The modified state vector and the covariance matrix of the process noise will be $\mathbf{B} = \mathrm{blkdiag}\{\mathbf{A}, \mathbf{I}_M\}$ and $\mathbf{Q}_v(t) = \mathrm{blkdiag}\{\mathbf{Q}_w(t), \mathbf{Q}_b(t)\}$, respectively. $\mathbf{Q}_b(t)$ is the variance of the random walk and defines how fast the NLOS biases change from one time step to another. We will later show that $\mathbf{Q}_b(t)$ has a large impact on the accuracy of the estimator. The optimal variance is unknown to the estimator and difficult to determine, as it depends on several parameters such as the propagation environment and network configuration. Finally, we also need to include the NLOS biases in the measurement model as follows [52, 87]

$$h_i(\mathbf{s}(t)) = \|\mathbf{x}(t) - \mathbf{y}_i\|_2 + b_i(t) \tag{6.4}$$

where $b_i(t)$ is the NLOS bias of the ith anchor node at time step t, as defined above.

6.2.1 Sequential and Bayesian Estimation

The source node tracking problem includes two important concepts: sequential estimation and Bayesian estimation. In sequential estimation, data is received on-going and more data is available as time progresses. Therefore, the estimator updates its estimate every time new information is available [10, Ch. 8]. In Bayesian estimation, the parameter of interest is estimated based on some prior knowledge of the parameter available to the estimator. Exploiting these two concepts, we are able to solve the tracking problem recursively using a prediction obtained from the underlying state model and a series of on-going measurements. In Bayesian estimation, we first need to calculate the posterior probability density function (pdf). The posterior pdf of the model in (6.1) is obtained by [97]

$$p(\mathbf{u}(t)|\mathbf{z}(1:t)) = \frac{p(\mathbf{z}(t)|\mathbf{u}(t))p(\mathbf{u}(t)|\mathbf{z}(1:\,t-1))}{p(\mathbf{z}(t)|\mathbf{z}(1:t-1))} \tag{6.5}$$

where

$$p(\mathbf{u}(t)|\mathbf{z}(1:t-1)) = \int p(\mathbf{u}(t)|\mathbf{u}(t-1))p(\mathbf{u}(t-1)|\mathbf{z}(1:t-1))d\mathbf{u}(t-1) \tag{6.6}$$

$$p(\mathbf{z}(t)|\mathbf{z}(1:t-1)) = \int p(\mathbf{z}(t)|\mathbf{u}(t))p(\mathbf{u}(t)|\mathbf{z}(1:t-1))d\mathbf{u}(t) \tag{6.7}$$

and (6.7) is just a normalization factor and ensures that $p(\mathbf{u}(t)|\mathbf{z}(1:t))$ integrates to 1. The probabilities $p(\mathbf{z}(t)|\mathbf{u}(t))$ and $p(\mathbf{u}(t)|\mathbf{u}(t-1))$ are defined by the measurement model and the state model in (6.1), respectively. The minimum mean square error (MMSE) estimator is obtained by finding the mean of the posterior pdf in (6.5). However, as sometimes it is difficult to determine the mean of the posterior pdf, the maximum a posteriori (MAP) estimator is used instead which does not require integration

$$\begin{aligned}\hat{\mathbf{u}}(t) &= \arg\max\, p(\mathbf{u}(t)|\mathbf{z}(1:t)) \\ &= \arg\max\, p(\mathbf{z}(t)|\mathbf{u}(t))p(\mathbf{u}(t)|\mathbf{z}(1:t-1)).\end{aligned} \tag{6.8}$$

If the posterior pdf is Gaussian, the mean and the maximum are identical. Therefore, MMSE and MAP estimators yield the same solution [10, Ch. 9].

6.2.2 Extended Kalman Filter

If the state and measurement models are linear, the solution of the problem in (6.8) is obtained by a Kalman filter which has a closed-from solution and coincides with the optimal MMSE estimator. However in the source node tracking problem in (6.1), the measurement model is nonlinear and both MAP and MMSE estimators are intractable. In this case, the approximate solution can be obtained by linearizing the model. The nonlinear measurement model is linearized around the true solution by using the first-order Taylor series expansion which yields the extended Kalman filter (EKF). Unlike the Kalman filter, the solution of the EKF is not optimal [10, Ch. 13]. In the EKF, the matrix $\mathbf{A}$ and the non-linear function $\mathbf{f}(\mathbf{u}(t))$ are used to calculate the predicted state vector from the previous estimate and the predicted measurement from the predicted state vector, respectively. However, due to the non-linear nature of $\mathbf{f}(\mathbf{u}(t))$, it cannot be directly applied to the covariance matrix. Dealing with this problem, the Jacobian matrix including the first-order partial derivatives of $\mathbf{f}(\mathbf{u}(t))$ is formulated and evaluated with the predicted state vector. The EKF is calculated recursively at each time step where new data is available to the estimator. The details for the classic EKF for the model given in (6.1) is given in [96, 98]. The EKF estimator for the model in (6.3) which considers the NLOS biases in the state vector is also derived in [95, 92].

6.2.3 Bias Tracking

Before deriving the proposed estimator, we would like to point out an important finding. More specifically, we want to examine whether or not using EKF with the bias tracking (EKFB) is useful in NLOS environments. Consider a simple network in Fig. 6.1. There are four anchor nodes and a moving source node. The measurements are obtained based on the model given in (6.2). However, to observe the effect of NLOS propagation on the localization performance, for this simulation we did not add noise to the measurements. The NLOS biases are generated by using a ray tracing technique [99]. If the signal travels between the source node and an anchor node with more than two bounces, we assume there is no link between the source node and that anchor node. Kalman

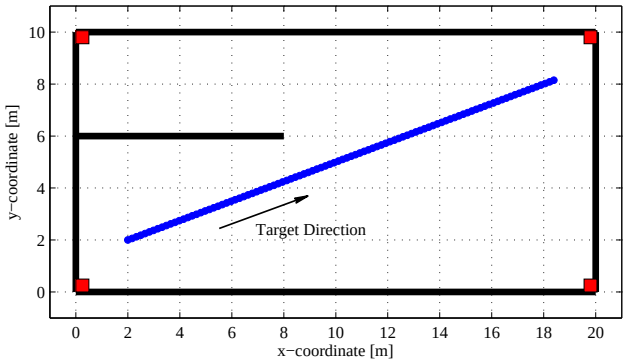


Figure 6.1: The configuration of the simulated network. The red squares represent the anchor nodes and the blue circles represent the trajectory of the source node.

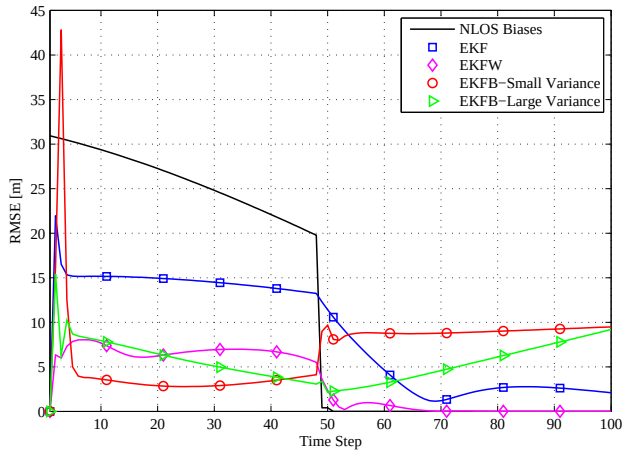


Figure 6.2: The RMSE of the EKF and EKFB estimators for the network in Fig. 6.1. No measurement noise is included to evaluate the impact of the NLOS biases on the localization performance.

filters require initialization. The initial state of the source node location is obtained by a linear least squares estimator [54] which requires no initialization. The initial state of the source node velocity is set to zero, as no information is available at the beginning. In the EKFB, the initial state of the NLOS biases is also set to zero. Another parameter that needs to be defined for the EKFB is $\mathbf{Q}_b(t)$ which has a strong effect on its performance. In our simulations, we implemented the EKFB for both large and small covariance matrices corresponding to the fast and slow changes in the NLOS biases, respectively.

Fig. 6.2 shows the RMSE of the EKF and EKFB estimators for the network in Fig. 6.1. The black solid curve represents the sum of the NLOS biases over all anchor nodes (in meters). We also include another estimator called EKFW which is the classic EKF except instead of the true covariance matrix of the measurements, the covariance matrix weighted with the range measurements is used, i.e., $\bar{\mathbf{Q}}_n(t) = \mathbf{Z}(t)\mathbf{Q}_n(t)\mathbf{Z}(t)$, $\mathbf{Z}(t) = \text{diag}\{z_1(t), \ldots, z_M(t)\}$. By using this method, the measurements that have larger values are assigned less weight, while the measurements with smaller values are assigned more weight. Comparing EKFW and EKF in Fig. 6.2, we observe that

using a simple weighting method can improve the performance significantly. As depicted in Fig. 6.2, EKFB shows little to no improvement over EKF or EKFW for either small or large variance for the NLOS biases. The main problem of the EKFB with a small variance is that it cannot respond to the fast NLOS changes effectively. The EKFB starts with a large error mainly because the initial state of the NLOS biases is zero. Then it compensates for the NLOS biases and brings itself to a steady state. However, at time step 50 when there is a sudden change in NLOS biases, the performance of the EKFB degrades substantially and it cannot return to a steady state afterwards. Using an EKFB with a large variance also does not help the estimator. Although it can track the NLOS changes faster, its performance is not steady when the NLOS links are nearly static. Based on this clear example and several other simulations, we conclude that despite a considerably higher complexity than EFK, EKFB cannot provide good performance in practical NLOS environments where the NLOS biases can change rapidly. There are two main reasons for this. First, modeling the NLOS biases is a very difficult task, especially in indoor environments where there are many objects and obstacles. Assuming a small variance for the EKFB provides good accuracy in static NLOS situations. However in this case, the EKFB cannot track the biases when they change rapidly. On the other hand, assuming a large variance may enable the EKFB to respond to NLOS changes more quickly, but its performance is unstable in static NLOS situations. Using an adaptive and smart way to change the variance might provide a solution. However, it is again highly dependent on how accurately we can model NLOS propagation in a network. Second, the EKFB is in fact an underdetermined problem (i.e., there are fewer measurements than unknown variables). Hence, there is no guarantee that the solution obtained by the EKFB is accurate in terms of RMSE. Moreover, as will be discussed later, EKFB becomes very complex in cooperative networks.

6.2.4 Semidefinite Programming-Based Estimator

In this section, the proposed SDP estimator is described. In the proposed estimator, the NLOS biases are estimated jointly with the source node location and velocity. However, given the conclusions in the previous section, the estimator will not track the NLOS biases and only estimates them as nuisance parameters at each time step independently. Therefore, the NLOS biases are not included in the state vector and their estimates at each time step are not transferred to the next step. Unlike the EKFB, the proposed SDP requires no prior information about the NLOS propagation. Suppose $p(\mathbf{u}(t-1)|\mathbf{z}(1:t-1))$ in (6.6) is Gaussian with the mean $\hat{\mathbf{u}}(t-1|t-1)$ and covariance matrix of $\mathbf{P}(t-1|t-1)$. Hence, $p(\mathbf{u}(t)|\mathbf{z}(1:t-1))$ will be Gaussian with the mean $\hat{\mathbf{u}}(t|t-1)$ and covariance matrix $\mathbf{P}(t|t-1)$, where

$$\begin{aligned}\hat{\mathbf{u}}(t|t-1) &= \mathbf{A}\hat{\mathbf{u}}(t-1|t-1)\\ \mathbf{P}(t|t-1) &= \mathbf{A}\mathbf{P}(t-1|t-1)\mathbf{A}^{\mathrm{T}} + \mathbf{Q}_w(t).\end{aligned} \tag{6.9}$$

The MAP estimator in (6.8) can be alternatively written as

$$\begin{aligned}\hat{\mathbf{u}}(t) &= \arg\max\ \ln p(\mathbf{z}(t)|\mathbf{u}(t))p(\mathbf{u}(t)|\mathbf{z}(1:t-1)) \\ &= \arg\max\ \{\ln p(\mathbf{z}(t)|\mathbf{u}(t)) + \ln p(\mathbf{u}(t)|\mathbf{z}(1:t-1))\} \\ &= \arg\min\ \{(\mathbf{z}(t)-\mathbf{h}(\mathbf{u}(t)))^{\mathrm{T}}\mathbf{Q}_n^{-1}(t)(\mathbf{z}(t)-\mathbf{h}(\mathbf{u}(t))) \\ &\quad + (\mathbf{u}(t)-\hat{\mathbf{u}}(t|t-1))^{\mathrm{T}}\mathbf{P}^{-1}(t|t-1)(\mathbf{u}(t)-\hat{\mathbf{u}}(t|t-1))\} \end{aligned} \tag{6.10}$$

where the last expression comes from the fact that both pdfs are Gaussian. The first and second terms of (6.10) define a minimization over the measurement model and the prediction likelihood functions, respectively. Now, the goal is to formulate the nonconvex and nonlinear optimization problem in (6.10) as a convex one. The first and second terms are separately formulated as convex problems and they are then combined and written as an SDP problem.

Let us start with the first term of (6.10) which is a minimization over the measurement model. The measurement model in (6.4) can be rewritten as

$$z_i(t) = d_i(t) + b_i(t) + n_i(t), \quad i \in \mathcal{A} \tag{6.11}$$

where $d_i(t) = \|\mathbf{x}(t) - \mathbf{y}_i\|$. Squaring both sides of (6.11) yields

$$\begin{aligned} z_i^2(t) &= (d_i(t) + b_i(t) + n_i(t))^2, \\ &= d_i^2(t) + c_i(t) + 2n_i(t)(b_i(t) + d_i(t)) + n_i^2(t) \end{aligned} \tag{6.12}$$

where $c_i(t) = b_i^2(t) + 2b_i(t)d_i(t)$. For sufficiently small noise, the the noise term $n_i^2(t)$ can be neglected, and we have

$$z_i^2(t) \simeq d_i^2(t) + c_i(t) + \epsilon_i(t) \tag{6.13}$$

where $\epsilon_i(t) = 2\bar{d}_i(t)n_i(t)$ and

$$\bar{d}_i(t) = b_i(t) + d_i(t). \tag{6.14}$$

The nonlinear least squares estimator of the unknown parameters $[\mathbf{x}(t); \mathbf{c}(t)]$ is obtained as [10, Ch. 8]

$$\begin{aligned} &\underset{\mathbf{x}(t),c_i(t)}{\text{minimize}} \quad \sum_{i\in\mathcal{A}} v_i(t)\left(z_i^2(t) - d_i^2(t) - c_i(t)\right)^2, \\ &\text{subject to} \quad c_i(t) = b_i^2(t) + 2b_i(t)d_i(t) \end{aligned} \tag{6.15}$$

where $v_i(t) = (\bar{d}_i^2(t)\sigma_i^2(t))^{-1}$ is the weighting element which is equal to the inverse of the variance of $\epsilon_i(t)$ in (6.13). It should be noted that $v_i(t)$ depends on the unknown parameter $\bar{d}_i^2(t)$. However, it can be approximated by using $z_i^2(t)$ instead of $\bar{d}_i^2(t)$ as $v_i(t) \simeq (z_i^2(t)\sigma_i^2(t))^{-1}$. Unfortunately, the cost function of the minimization problem in (6.15) is still nonlinear and nonconvex. Next, we will convert the problem in (6.15) into a convex optimization problem. To do that, we neglect

the relationship between $d_i(t)$, $b_i(t)$, and $c_i(i)$ which is defined above and estimate $c_i(t)$ as an independent unknown parameter. Let us define two auxiliary variables as

$$\begin{aligned}\beta_i(t) &= d_i^2(t) \\ &= \|\mathbf{x}(t)-\mathbf{y}_i\|^2 \\ &= (\mathbf{x}(t)-\mathbf{y}_i)^{\mathrm{T}}(\mathbf{x}(t)-\mathbf{y}_i) \\ &= \begin{bmatrix}\mathbf{y}_i \\ -1\end{bmatrix}^{\mathrm{T}} \begin{bmatrix}\mathbf{I}_2 & \mathbf{x}(t) \\ \mathbf{x}^{\mathrm{T}}(t) & \alpha(t)\end{bmatrix} \begin{bmatrix}\mathbf{y}_i \\ -1\end{bmatrix}\end{aligned} \tag{6.16}$$

where $\alpha(t)=\mathbf{x}^{\mathrm{T}}(t)\mathbf{x}(t)$. Plugging (6.16) in (6.15) yields

$$\begin{aligned}\underset{\mathbf{x}(t),c_i(t)}{\text{minimize}} \quad & \sum_{i\in\mathcal{A}} v_i(t)\left(z_i^2(t)-\beta_i(t)-c_i(t)\right)^2 \\ \text{subject to} \quad & \beta_i(t)=\begin{bmatrix}\mathbf{y}_i \\ -1\end{bmatrix}^{\mathrm{T}} \begin{bmatrix}\mathbf{I}_2 & \mathbf{x}(t) \\ \mathbf{x}^{\mathrm{T}}(t) & \alpha(t)\end{bmatrix} \begin{bmatrix}\mathbf{y}_i \\ -1\end{bmatrix} \\ & \alpha(t)=\mathbf{x}^{\mathrm{T}}(t)\mathbf{x}(t), \quad c_i(t)\geq 0.\end{aligned} \tag{6.17}$$

Now the cost function of (6.17) is a quadratic function of unknown parameters, hence it is convex. However, there is still a nonconvex constraint on $\alpha(t)$ in (6.17). To convert the problem in (6.17) into a convex SDP problem, the nonconvex constraint needs to be relaxed. Using the Schur complement [16], $\alpha(t)$ can be relaxed and written as a linear matrix inequality (LMI)

$$\alpha(t)=\mathbf{x}^{\mathrm{T}}(t)\mathbf{x}(t) \quad \Rightarrow \begin{bmatrix}\mathbf{I}_2 & \mathbf{x}(t) \\ \mathbf{x}^{\mathrm{T}}(t) & \alpha(t)\end{bmatrix} \succeq \mathbf{0}_3. \tag{6.18}$$

Although the cost function of (6.17) is convex, it is not in a general form of SDP, since the cost function should be a linear function of the optimization variables. This can be done by using the epigraph representation of the quadratic function, which can be formulated by adding a new scalar variable τ_1. Using the relaxation in (6.18), we can express the problem in (6.17) as [15, 27]

$$\begin{aligned}\underset{\tau_1,\mathbf{x}(t),\alpha(t),\mathbf{c}}{\text{minimize}} \quad & \tau_1 \\ \text{subject to} \quad & \begin{bmatrix}\mathbf{Z}(t)\mathbf{Q}_n(t)\mathbf{Z}(t) & (\boldsymbol{\xi}(t)-\boldsymbol{\beta}(t)-\mathbf{c}) \\ (\boldsymbol{\xi}(t)-\boldsymbol{\beta}(t)-\mathbf{c})^{\mathrm{T}} & \tau_1\end{bmatrix} \succeq \mathbf{0}_{M+1} \\ & \beta_i(t)=\begin{bmatrix}\mathbf{y}_i \\ -1\end{bmatrix}^{\mathrm{T}} \begin{bmatrix}\mathbf{I}_2 & \mathbf{x}(t) \\ \mathbf{x}^{\mathrm{T}}(t) & \alpha(t)\end{bmatrix} \begin{bmatrix}\mathbf{y}_i \\ -1\end{bmatrix}, \\ & \begin{bmatrix}\mathbf{I}_2 & \mathbf{x}(t) \\ \mathbf{x}^{\mathrm{T}}(t) & \alpha(t)\end{bmatrix} \succeq \mathbf{0}_3, \quad \mathbf{c}\geq 0.\end{aligned} \tag{6.19}$$

where $\boldsymbol{\xi}(t)=\mathbf{z}(t)\circ\mathbf{z}(t)$, $\boldsymbol{\beta}(t)=[\beta_1(t),\beta_2(t),\ldots,\beta_M(t)]^{\mathrm{T}}$. Note that $\mathbf{c}$ is a slack vector which tries

to compensate for the NLOS biases. Since we are not interested in the estimate of the NLOS biases nor tracking them, $\mathbf{c}$ is not a function of t and is overwritten at each step.

We would like to impose a constraint on the estimate of the source node location which will considerably improve the performance, especially in severe NLOS environments [52]. Since the biases of NLOS connections in (6.4) are positive and typically much larger than the measurement noise, we can write [52, 51]

$$z_i(t) \geq d_i(t) = \|\mathbf{x}(t) - \mathbf{y}_i\|. \tag{6.20}$$

Equivalently,

$$z_i^2(t) \geq d_i^2(t) = (\mathbf{x}(t) - \mathbf{y}_i)^{\mathrm{T}}(\mathbf{x}(t) - \mathbf{y}_i). \tag{6.21}$$

The constraint in (6.21) forces the estimated position of the source node to be within the circles determined by the anchor nodes and the radii $z_i(t)$. However, for LOS connections, the constraint in (6.21) may not be valid, because LOS connections do not have large NLOS biases and also the measurement noises $n_i(t)$ may have negative values. Therefore, the position of the source node may be located outside of a circle defined by a LOS connection. Since we do not know which connections are LOS, we have to make the constraint in (6.21) slightly looser for all connections. About 95% of a Gaussian random variable's probability density lies within two standard deviations of its mean. Hence, the constraint in (6.21) can be modified as

$$\begin{aligned} z_i^2(t) + 2\sqrt{\mathrm{Var}[z_i^2(t)]} &\geq d_i^2(t) \\ &\geq (\mathbf{x}(t) - \mathbf{y}_i)^{\mathrm{T}}(\mathbf{x}(t) - \mathbf{y}_i). \end{aligned} \tag{6.22}$$

Now, we need to calculate the variance of $z_i^2(t)$. Assuming $d_i(t)$ and $b_i(t)$ are the unknown deterministic variables, we can rewrite (6.11) as

$$z_i(t) = \bar{d}_i(t) + n_i(t) \tag{6.23}$$

where $\bar{d}_i(t)$ is defined in (6.14). Therefore, we have

$$\begin{aligned} z_i^2(t) &= (\bar{d}_i(t) + n_i(t))^2 \\ &= \bar{d}_i^2(t) + 2\bar{d}_i(t)n_i(t) + n_i^2(t). \end{aligned} \tag{6.24}$$

Hence, the variance of $z_i^2(t)$ is calculated as

$$\begin{aligned} \mathrm{Var}[z_i^2(t)] &= \mathrm{E}[z_i^4(t)] - (\mathrm{E}[z_i^2(t)])^2 \\ &= 4\bar{d}_i^2(t)\sigma_i^2(t), \end{aligned} \tag{6.25}$$

where $\mathrm{E}[\,\cdot\,]$ denotes the expected value. Again, since the variance of $z_i^2(t)$ depends on the unknown

parameter $\bar{d}_i^2(t)$, we need to approximate the variance by using $z_i^2(t)$ instead. One of the most useful advantages of the SDP estimator is that one can simply apply constraints to the solution [15]. The constraint in (6.22) is convex. However, it is not expressed in a general form of SDP. Plugging (6.25) into (6.22) and using the Schur complement, the constraint in (6.22) can be written as an LMI as

$$\begin{bmatrix} \mathbf{I}_2 & (\mathbf{x}(t)-\mathbf{y}_i) \\ (\mathbf{x}(t)-\mathbf{y}_i)^{\mathrm{T}} & \xi_i(t)+4z_i(t)\sigma_i(t) \end{bmatrix} \succeq \mathbf{0}_3. \tag{6.26}$$

Adding the constraint in (6.26) to the problem in (6.19) yields

$$\begin{aligned} &\underset{\tau_1,\mathbf{x}(t),\alpha(t),\mathbf{c}}{\text{minimize}} \ \ \tau_1 \\ &\text{subject to} \ \ \beta_i(t) = \begin{bmatrix} \mathbf{y}_i \\ -1 \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \mathbf{I}_2 & \mathbf{x}(t) \\ \mathbf{x}^{\mathrm{T}}(t) & \alpha(t) \end{bmatrix} \begin{bmatrix} \mathbf{y}_i \\ -1 \end{bmatrix}, \\ &\quad \begin{bmatrix} \mathbf{Z}(t)\mathbf{Q}_n(t)\mathbf{Z}(t) & (\boldsymbol{\xi}(t)-\boldsymbol{\beta}(t)-\mathbf{c}) \\ (\boldsymbol{\xi}(t)-\boldsymbol{\beta}(t)-\mathbf{c})^{\mathrm{T}} & \tau_1 \end{bmatrix} \succeq \mathbf{0}_{M+1} \\ &\quad \begin{bmatrix} \mathbf{I}_2 & \mathbf{x}(t) \\ \mathbf{x}^{\mathrm{T}}(t) & \alpha(t) \end{bmatrix} \succeq \mathbf{0}_3, \quad \mathbf{c} \geq 0 \\ &\quad \begin{bmatrix} \mathbf{I}_2 & (\mathbf{x}(t)-\mathbf{y}_i) \\ (\mathbf{x}(t)-\mathbf{y}_i)^{\mathrm{T}} & \xi_i(t)+4z_i(t)\sigma_i(t) \end{bmatrix} \succeq \mathbf{0}_3. \end{aligned} \tag{6.27}$$

which is an SDP optimization problem. It is worth mentioning that in the SDP problem in (6.27), when most of the connections are LOS, the cost function plays an important role and the constraint in (6.26) barely has an effect on the solution. However, when most of the connections are NLOS, the constraint is more effective and keeps the solution from exhibiting a large estimation error [52].

Now let us focus on the second term of (6.10) which is a minimization over the prediction likelihood. It is a form of quadratic programming which can be expressed as an SDP problem by using the epigraph representation as follows [15]

$$\begin{aligned} &\underset{\tau_2,\mathbf{u}(t)}{\text{minimize}} \ \ \tau_2 \\ &\text{subject to} \ \begin{bmatrix} \mathbf{P}(t|t-1) & (\mathbf{u}(t)-\hat{\mathbf{u}}(t|t-1)) \\ (\mathbf{u}(t)-\hat{\mathbf{u}}(t|t-1))^{\mathrm{T}} & \tau_2 \end{bmatrix} \succeq \mathbf{0}_5. \end{aligned} \tag{6.28}$$

Now, we have formulated the two terms in (6.10) as a SDP problem. Therefore, the proposed estimator is obtained by combining (6.28) and (6.27)

$$\begin{aligned} &\underset{\substack{\tau_1,\tau_2,\tau_3 \\ \mathbf{u}(t),\alpha(t),\mathbf{c}}}{\text{minimize}} \ \ \tau_1+\tau_2+\tau_3 \\ &\text{subject to} \ \ \beta_i(t) = \begin{bmatrix} \mathbf{y}_i \\ -1 \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \mathbf{I}_2 & \mathbf{x}(t) \\ \mathbf{x}^{\mathrm{T}}(t) & \alpha(t) \end{bmatrix} \begin{bmatrix} \mathbf{y}_i \\ -1 \end{bmatrix}, \end{aligned}$$

$$
\begin{bmatrix} \mathbf{Z}(t)\mathbf{Q}_{n,t}\mathbf{Z}(t) & (\boldsymbol{\xi}(t)-\boldsymbol{\beta}(t)-\mathbf{c}) \\ (\boldsymbol{\xi}(t)-\boldsymbol{\beta}(t)-\mathbf{c})^{\mathrm{T}} & \tau_1 \end{bmatrix} \succeq \mathbf{0}_{M+1},
$$

$$
\begin{bmatrix} \mathbf{I}_2 & \mathbf{x}(t) \\ \mathbf{x}^{\mathrm{T}}(t) & \alpha(t) \end{bmatrix} \succeq \mathbf{0}_3, \begin{bmatrix} \gamma\mathbf{I}_M & \mathbf{c} \\ \mathbf{c}^{\mathrm{T}} & \tau_3 \end{bmatrix} \succeq \mathbf{0}_3, \mathbf{c} \geq 0,
$$

$$
\begin{bmatrix} \mathbf{P}(t|t-1) & (\mathbf{u}(t)-\hat{\mathbf{u}}(t|t-1)) \\ (\mathbf{u}(t)-\hat{\mathbf{u}}(t|t-1))^{\mathrm{T}} & \tau_2 \end{bmatrix} \succeq \mathbf{0}_5,
$$

$$
\begin{bmatrix} \mathbf{I}_2 & (\mathbf{x}(t)-\mathbf{y}_i) \\ (\mathbf{x}(t)-\mathbf{y}_i)^{\mathrm{T}} & \xi_i(t)+4z_i(t)\sigma_i(t) \end{bmatrix} \succeq \mathbf{0}_3. \tag{6.29}
$$

The solution of (6.29) can be effectively found with numerical algorithms such as interior point methods [15, 16, 71]. In MATLAB simulations, standard SDP solvers such as SeDuMi [17] can be employed to solve the SDP in (6.29). The parameter γ is a penalization factor which is required when the problem is ill-posed [87] [16, Ch. 6]. It also constrains c_i to a reasonable size. Note that the slack variable τ_3 is introduced to represent the penalization factor as an LMI.

The estimate of the state vector, $\hat{\mathbf{u}}(t)$, at the tth time step obtained from (6.28) will be used for the next time step. However, we also need to have the variance of the estimate, $\mathbf{P}(t|t)$, which defines how accurate this estimate is and will be used for weighting in the next step. For instance, if the variance is high, it means that the estimate is not accurate. Therefore, for the next step, less weight is assigned to the previous estimate and more weight is assigned to the measurements. The problem of the estimator in (6.28) is that calculating the variance of the estimate obtained from an SDP estimator is intractable. To handle this problem, the Cramér-Rao lower bound (CRLB) is used to calculate the variance. Using the CRLB leads the estimate to be assigned more weight, as the CRLB is typically lower than the actual variance of the estimate. This problem can be resolved by regulating the variance of the measurements accordingly.

6.3 Cooperative Tracking

This section extends the model described in section II to cooperative networks. In cooperative networks, multiple source nodes are involved and they can communicate not only with anchor nodes but also with each other [1]. Therefore, two sets of measurements are collected in the network: source-anchor and source-source measurements. Assume there are N source nodes and M anchor nodes in the network. Denote by $\mathcal{B} = \{1, \ldots, N\}$ the set of indices of the source nodes, by $\mathcal{A} = \{N+1, \ldots, N+M\}$ the set of indices of the anchor nodes, and by $\mathbf{u}_k(t)$, $k \in \mathcal{S}$, the state vector of the kth source node at time step t. In cooperative networks, connectivity is typically limited, meaning that the source nodes are not connected to all anchor nodes and other source nodes. To include the connectivity in our model, two sets are defined for each source node. Let $\mathcal{A}_k(t) = \{\ i \mid \text{anchor node } i \text{ is connected to source node } k\ \}$ be the set of the indices of the anchor nodes connected to the kth source node at time step t and $\mathcal{B}_k(t) = \{\ i \mid \text{source node } i \text{ is connected to source node } k\ \}$ be the set of indices of the source nodes

connected to the kth source node at time step t. The state model for cooperative tracking is the same as noncooperative. However, the measurement model is slightly different, since we need to include both source-source and source-anchor measurements. Therefore, the tracking model for the cooperative case

$$\begin{aligned}
\mathbf{u}_k(t) &= \mathbf{A}\mathbf{u}_k(t-1) + \mathbf{w}_k(t), && k \in \mathcal{B} \\
z_{ki}(t) &= \|\mathbf{x}_k(t) - \mathbf{y}_i\| + b_{ki}(t) + n_{ik}(t), && i \in \mathcal{A}_k(t) \\
z_{ki}(t) &= \|\mathbf{x}_k(t) - \mathbf{x}_i(t)\| + b_{ki}(t) + n_{ik}(t), && i \in \mathcal{B}_k(t)
\end{aligned} \tag{6.30}$$

where $\mathbf{A}$ is defined in (6.2), $\mathbf{w}_k(t)$ is the process noise, and $n_{ik}(t)$ is the measurement noise. Although the model enables us to have a different process noise for each source node, the same process noise is considered for all source nodes in practice.

6.3.1 Extended Kalman Filter

For the noncooperative case, three estimators based on the Kalman filter were introduced. Those estimators can be extended to the cooperative case as well. Remember that for the EKFB, the NLOS biases are included in the state vector. Let us modify the state vector $\mathbf{s}(t)$ as follows:

$$\mathbf{s}(t) = [\mathbf{u}_1^{\mathrm{T}}(t), \ldots, \mathbf{u}_N^{\mathrm{T}}(t), \mathbf{b}_1^{\mathrm{T}}(t), \ldots, \mathbf{b}_N^{\mathrm{T}}(t)]^{\mathrm{T}}. \tag{6.31}$$

A problem that arises in this case is that the size of the matrices increases dramatically as the number of source node increases in the network. The reason is that the EKFB needs to track many NLOS biases over time which increases the estimator's complexity. For instance, in a cooperative network with full connectivity (i.e., each source node is connected to all anchor nodes and other source nodes), the size of the state vector is $4N + N(M + N - 1)$. In fact, our simulations show that the complexity is so high that typical computers run out of memory when simulating a network with 8 anchor nodes and 30 source nodes with full connectivity. Although, as it will be later shown, the EKFB can provide some improvement over the classic EKF, the complexity is too high which makes it impractical, especially for cooperative networks with many source nodes. Therefore for cooperative networks, we are more interested in the EKF and EKFW. Let us define an augmented state vector including the state vectors of all source nodes:

$$\mathbf{u}(t) = [\mathbf{u}_1^{\mathrm{T}}(t), \ldots, \mathbf{u}_N^{\mathrm{T}}(t)]^{\mathrm{T}}. \tag{6.32}$$

The state model considering the above state vector can be written as

$$\mathbf{u}(t) = \mathbf{D}\mathbf{u}(t-1) + \mathbf{w}(t) \tag{6.33}$$

where $\mathbf{w}(t) = [\mathbf{w}_1^{\mathrm{T}}(t), \ldots, \mathbf{w}_N^{\mathrm{T}}(t)]^{\mathrm{T}}$ and $\mathbf{D} = \mathrm{blkdiag}\{\mathbf{A}, \ldots, \mathbf{A}\}$. For a network with full connectivity, the size of the measurement vector $\mathbf{z}(t)$ is $(N(M + N - 1)) \times 1$. The size of Jacobian matrix

including the first-order derivatives of all measurements would be $(N(M+N-1)) \times 4N$. The readers are referred to [100] for more detail concerning the cooperative EKF.

6.3.2 Semidefinite Programming-Based Estimator

The proposed cooperative tracking SDP estimator is derived in this section. We follow the same approach as in the noncooperative case to relax the source-anchor measurement model. However, a different relaxation is required for source-source links. Therefore, we first describe how to relax the source-source links and then expand the noncooperative SDP estimator in (6.29) for cooperative networks. For source-source links, due to the shorter range we assume that the NLOS biases are negligible and are not estimated by the estimator to ease complexity. Therefore, the minimization problem for source-source node measurement can be obtained as

$$\underset{\mathbf{x}_k(t)}{\text{minimize}} \sum_{k\in\mathcal{B}, i\in\mathcal{B}_k(t)} v_{ki}(t)\left(z_{ki}(t)-d_{ki}(t)\right)^2. \tag{6.34}$$

where $v_{ki}(t) = z_{ki}^{-2}(t)$. The above problem can be further simplified as

$$\underset{\mathbf{x}_k(t)}{\text{minimize}} \sum_{k\in\mathcal{B}, i\in\mathcal{B}_k(t)} v_{ki}(t)\left(d_{ki}^2(t)-2d_{ki}(t)z_{ki}(t)\right). \tag{6.35}$$

As in the noncooperative case, to formulate the nonlinear and nonconvex problem in (6.35), two auxiliary variables are introduced as follows

$$\begin{aligned}\beta_{ki}(t) &= \begin{bmatrix}\mathbf{0}_2\\ \mathbf{e}_{ik}\end{bmatrix}^{\mathrm{T}} \begin{bmatrix}\mathbf{I}_2 & \mathbf{X}(t)\\ \mathbf{X}^{\mathrm{T}}(t) & \boldsymbol{\alpha}(t)\end{bmatrix} \begin{bmatrix}\mathbf{0}_2\\ \mathbf{e}_{ik}\end{bmatrix},\\ \boldsymbol{\alpha}(t) &= \mathbf{X}(t)\mathbf{X}^{\mathrm{T}}(t)\end{aligned} \tag{6.36}$$

where $\mathbf{X}(t) = [\mathbf{x}_1(t), \ldots, \mathbf{x}_N(t)]$, $\mathbf{e}_{ik} = \mathbf{e}_i - \mathbf{e}_k$ and $\mathbf{e}_i$ is an $N \times 1$ vector in which the ith element is one and the other elements are zero. Plugging the new variables in (6.36) into (6.35) yields

$$\begin{aligned}\underset{\substack{\mathbf{X}(t),\boldsymbol{\alpha}(t)\\ d_{ki}(t)}}{\text{minimize}} \quad & \sum_{k\in\mathcal{B}, i\in\mathcal{B}_k(t)} v_{ki}(t)\left(\beta_{ki}(t)-2d_{ki}(t)z_{ki}(t)\right)\\ \text{subject to} \quad & \beta_{ki}(t) = \begin{bmatrix}\mathbf{0}_2\\ \mathbf{e}_{ik}\end{bmatrix}^{\mathrm{T}} \begin{bmatrix}\mathbf{I}_2 & \mathbf{X}(t)\\ \mathbf{X}^{\mathrm{T}}(t) & \boldsymbol{\alpha}(t)\end{bmatrix} \begin{bmatrix}\mathbf{0}_2\\ \mathbf{e}_{ik}\end{bmatrix},\\ & \boldsymbol{\alpha}(t) = \mathbf{X}(t)\mathbf{X}^{\mathrm{T}}(t), \quad \beta_{ki}(t) = d_{ki}^2(t).\end{aligned} \tag{6.37}$$

By relaxing the noncovex terms, the problem in (6.37) is formulated as an SDP as follows

$$\underset{\substack{\mathbf{X}(t),\boldsymbol{\alpha}(t)\\ d_{ki}(t)}}{\text{minimize}} \sum_{k\in\mathcal{B}, i\in\mathcal{B}_k(t)} v_{ki}(t)\left(\beta_{ki}(t)-2d_{ki}(t)z_{ki}(t)\right)$$

$$\text{subject to} \quad \beta_{ki}(t) = \begin{bmatrix} \mathbf{0}_2 \\ \mathbf{e}_{ik} \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \mathbf{I}_2 & \mathbf{X}(t) \\ \mathbf{X}^{\mathrm{T}}(t) & \boldsymbol{\alpha}(t) \end{bmatrix} \begin{bmatrix} \mathbf{0}_2 \\ \mathbf{e}_{ik} \end{bmatrix},$$
$$\begin{bmatrix} \mathbf{I}_2 & \mathbf{X}(t) \\ \mathbf{X}^{\mathrm{T}}(t) & \boldsymbol{\alpha}(t) \end{bmatrix} \succeq \mathbf{0}_{N+2}, \begin{bmatrix} 1 & d_{ki}(t) \\ d_{ki}(t) & \beta_{ki}(t) \end{bmatrix} \succeq \mathbf{0}_2. \tag{6.38}$$

Now the proposed cooperative SDP estimator is obtained by expanding the problem in (6.29) for multiple source nodes and including the problem in (6.38) as follows

$$\begin{aligned}
&\underset{\substack{\mathbf{X}(t), d_{ki}(t) \\ \boldsymbol{\alpha}(t)}}{\text{minimize}} \quad \tau_2 + \sum_{k \in \mathcal{B}} \tau_1(k) + \tau_3(k) + \tau_4(k) \\
&\text{subject to} \quad \tau_4(k) = \sum_{i \in \mathcal{B}_k(t)} v_{ki}(t) \left(\beta_{ki}(t) - 2 d_{ki}(t) z_{ki}(t) \right) \\
&\begin{bmatrix} \mathbf{Z}_k(t) \mathbf{Q}_n^k(t) \mathbf{Z}_k(t) & (\boldsymbol{\xi}_k(t) - \boldsymbol{\beta}_k(t) - \mathbf{c}_k) \\ (\boldsymbol{\xi}_k(t) - \boldsymbol{\beta}_k(t) - \mathbf{c}_k)^{\mathrm{T}} & \tau_1(k) \end{bmatrix} \succeq \mathbf{0}_{M+1} \\
&\beta_{ki}(t) = \begin{bmatrix} \mathbf{y}_i \\ -\mathbf{e}_k \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \mathbf{I}_2 & \mathbf{X}(t) \\ \mathbf{X}^{\mathrm{T}}(t) & \boldsymbol{\alpha}(t) \end{bmatrix} \begin{bmatrix} \mathbf{y}_i \\ -\mathbf{e}_k \end{bmatrix}, i \in \mathcal{A}_k(t) \\
&\beta_{ki}(t) = \begin{bmatrix} \mathbf{0}_2 \\ \mathbf{e}_{ik} \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \mathbf{I}_2 & \mathbf{X}(t) \\ \mathbf{X}^{\mathrm{T}}(t) & \boldsymbol{\alpha}(t) \end{bmatrix} \begin{bmatrix} \mathbf{0}_2 \\ \mathbf{e}_{ik} \end{bmatrix}, i \in \mathcal{B}_k(t) \\
&\begin{bmatrix} \mathbf{I}_2 & \mathbf{X}(t) \\ \mathbf{X}^{\mathrm{T}}(t) & \boldsymbol{\alpha}(t) \end{bmatrix} \succeq \mathbf{0}_{N+2}, \begin{bmatrix} 1 & d_{ki}(t) \\ d_{ki}(t) & \beta_{ki}(t) \end{bmatrix} \succeq \mathbf{0}_2 \\
&\begin{bmatrix} \mathbf{P}(t|t-1) & (\mathbf{u}(t) - \hat{\mathbf{u}}(t|t-1)) \\ (\mathbf{u}(t) - \hat{\mathbf{u}}(t|t-1))^{\mathrm{T}} & \tau_2 \end{bmatrix} \succeq \mathbf{0}_{4N+1}, \\
&\begin{bmatrix} \mathbf{I}_2 & (\mathbf{x}_k(t) - \mathbf{y}_i) \\ (\mathbf{x}_k(t) - \mathbf{y}_i)^{\mathrm{T}} & \xi_{ki}(t) + 4 z_{ki}(t) \sigma_{ki}(t) \end{bmatrix} \succeq \mathbf{0}_3 \\
&\begin{bmatrix} \gamma \mathbf{I}_M & \mathbf{c}_k \\ \mathbf{c}_k^{\mathrm{T}} & \tau_3(k) \end{bmatrix} \succeq \mathbf{0}_M.
\end{aligned} \tag{6.39}$$

Similar to the noncooperative case, the SDP problem in (6.39) can be effectively solved by standard SDP solvers in polynomial time.

6.4 Simulation Results

The performance of the proposed estimator is evaluated through computer simulations. The proposed SDP estimator is implemented with the `CVX` toolbox [44] using SeDuMi as a solver [17]. The value of γ for the proposed SDP estimator was set to 10.

Consider again the noncooperative network in Fig. 6.1. Now measurement noise is also included. The variances of the range measurements, $\mathbf{Q}_n(t)$, are dependent on the true distances [51, 87]. The parameters are set such that the average error on the range measurements is about 15% of the

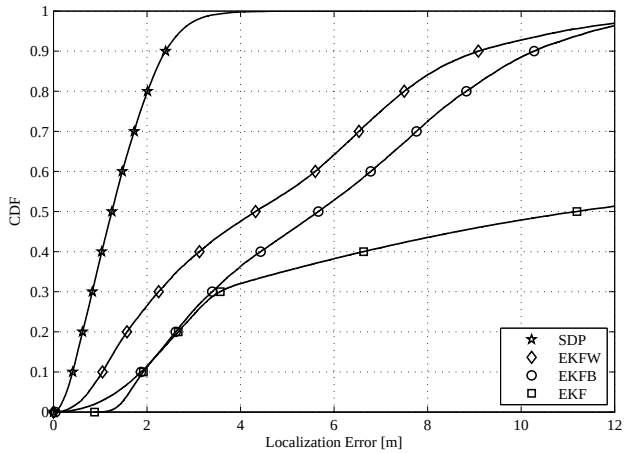


Figure 6.3: The CDF of localization error for the considered estimators and the noncooperative network in Fig. 6.1.

true ranges. Fig. 6.3 shows the cumulative distribution function (CDF) of the localization error for the considered estimators. The performance of EKF degrades substantially in a simple NLOS scenario where only one anchor node has NLOS propagation. Therefore, the classic EKF cannot be expected to provide good accuracy in practical cases where the majority of links are NLOS. Estimating and tracking the NLOS biases, the EKFB provides improvement in comparison with the classic EKF. However as also depicted in Fig. 6.2, by applying a simple weighting method in the classic EKF (labeled as EKFW), we can overcome the NLOS biases more efficiently. It is clear that in this case using EKFW is more favorable, since it provides better accuracy than EKFB with considerably lower complexity (EKFB has M more variables in the state vector to be estimated at each time step). The proposed SDP estimator outperforms the other estimators significantly. As mentioned before, the main features of the proposed SDP estimator is that it estimates the NLOS biases *independently* at each time step to mitigate NLOS propagation. It also has several constraints which diminish the effect of NLOS links on the solution.

Now consider a more complicated network given in Fig. 6.4. There are 4 anchor nodes at the corners of the area. Fig. 6.5a shows the localization performance of the noncooperative estimators. Again, the NLOS biases are simulated using a ray tracing technique. In this scenario, since there are only four anchor nodes and a lot of walls between the source node and anchor nodes, most of the time the source node has only one connection (either LOS or NLOS). However, the proposed SDP provides better accuracy than the other estimators. Again, EKFW with a simple weighting technique performs better than EKFB with bias tracking. EKF representing the performance the classic Kalman filter fails to provide reasonable accuracy. More specifically, the localization error is larger than 6 m more than 50% of the time which is relatively high for the considered network with a size of 13 m $\times$ 30 m. Although EKF is a simple estimator, it is not useful for indoor and dense environments. Therefore, more sophisticated estimators with NLOS mitigation techniques are necessary.

In Fig. 6.5b, the localization performance of the proposed estimator in a cooperative network is

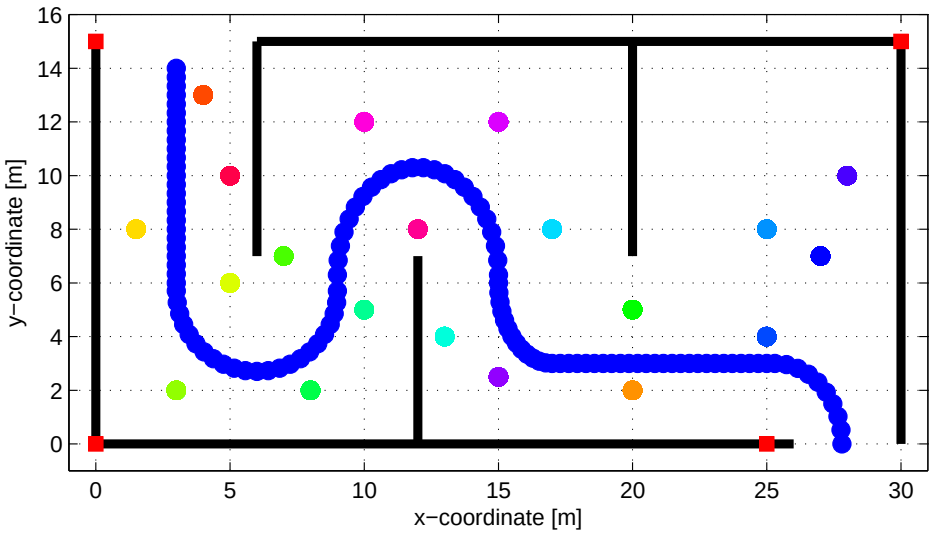
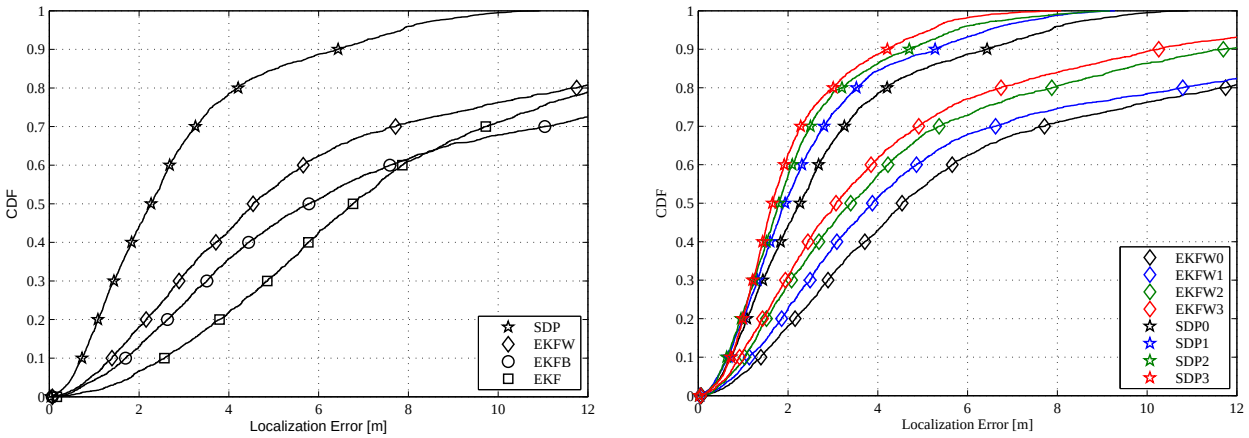


Figure 6.4: The configuration of the simulated network. The red squares and the blue circles represent the anchor nodes and the trajectory of the desired source node, respectively. The desired source node travels the path in 121 time steps. Other source nodes are static and depicted by solid colored circles.



(a) Noncooperative **(b)** Cooperative

Figure 6.5: The CDF localization error for the considered estimators and the network in Fig. 6.4. The proposed SDP estimator significantly outperforms the other estimators in both noncooperative and cooperative networks. In SDPx and EKWx, x represents the number of collaborators. Zero collaborators shows the performance of the noncooperative tracking.

provided. Only the proposed SDP estimator and EKFW are used for this simulation, since EKFB is too complex to simulate and EKF performs poorly. The same network as in Fig. 6.4 is considered, except 20 other static source nodes are placed in the network. However, the estimator does not know that the source nodes are not moving and keeps tracking them throughout the simulation. For source-source links, a source node collects TOA measurements from other source nodes which are in its LOS path. In other words, we assume that all source-source links are LOS and if source nodes do not have a direct view of each other, they are assumed not to be connected. Note that this

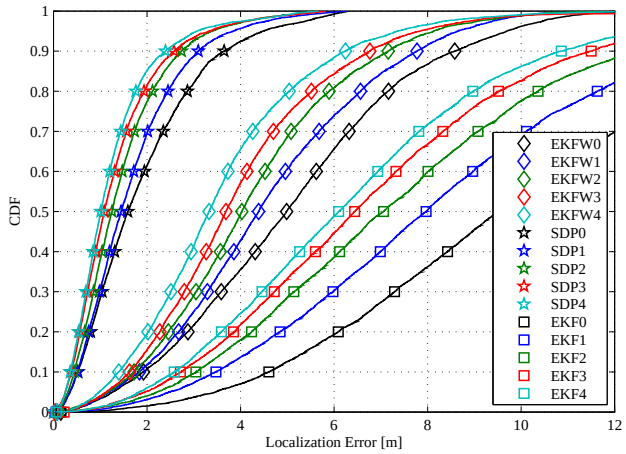


Figure 6.6: The CDF of localization error for the considered estimators and a random cooperative network. Legend: Algx stands for the specific algorithm with x collaborators. For instance, SDP3 stands for the proposed SDP estimator with 3 collaborators.

is not an impractical assumption, since the source nodes typically operate at lower power levels and the signal is rarely received at the other source nodes if it bounces multiple times. Fig. 6.5b shows the performance of the considered estimators for different number of collaborators. In SDPx and EKWx, x represents the number of collaborators. Zero collaborators represents the performance of the noncooperative case which is consistent with the results in Fig. 6.5a. For instance, SDP1 represents the performance of the proposed SDP estimator for the desired source node who is collaborating with another source node during its movement. During its movement, the desired source node randomly selects a source node from all the source nodes to whom it is connected. The localization performance improves for both estimators as the number of collaborators increases, especially at high reliability levels. For instance, at 90% CDF, the performance of the proposed SDP estimator improves from 6 m to 4 m (which is more than 33% improvement) if the desired source node collaborates with 3 other source nodes in the network. Collaboration is also useful for EKFW estimator where more than 5 m improvement is achieved at 80% CDF by 3 collaborators for the desired source node.

In the last simulation, a completely random network of size 60 m $\times$ 60 m is considered. There are eight anchor nodes in the network located at $[2,2]^{\mathrm{T}}$, $[2,58]^{\mathrm{T}}$, $[58,58]^{\mathrm{T}}$, $[58,2]^{\mathrm{T}}$, $[16,30]^{\mathrm{T}}$, $[30,16]^{\mathrm{T}}$, $[30,44]^{\mathrm{T}}$, and $[44,30]^{\mathrm{T}}$. There are 20 source nodes moving based on random walks inside the region covered by the anchor nodes. In this case, connections between the source nodes and anchor nodes are randomly assigned as either LOS or NLOS where the NLOS probability is 0.8 and 0.4 for source-anchor and source-source links, respectively. The NLOS bias errors are exponentially distributed with their means randomly selected from 2 to 20 m for source-anchor links and from 0 to 5 m for source-source links. Fig. 6.6 shows the CDF of the localization performance of the considered estimator for the considered random network for different numbers of collaborators. The same behavior as in Fig. 6.5b can be observed and the order of the estimators remains unchanged. Collaboration provides significant improvement for all estimators. The performance of EKF is

worse than the other estimators. However, collaboration is more advantageous for EKF than other estimators. The proposed SDP estimator also benefits from collaboration, although the improvement rate in the proposed SDP estimator is not as significant as in the other estimators, due to the fact that its performance is already good without collaboration.

In summary, results show the great advantage of cooperative networks over noncooperative ones. Collaboration improves the localization performance of all considered estimators. However, collaboration is more useful for the estimators which have poor performance in noncooperative networks. Moreover, the proposed SDP estimator provides superior performance to all other considered estimators in NLOS environments.

6.4.1 Remark on Complexity

This section discusses the computational complexity of the considered estimators in terms of running time. Although the running time does not say everything about the complexity, there is no other way to compare the complexity of the considered estimators, since complexity analysis is intractable. Similar to the previous case, a random network is considered. Table 6.1 summarizes the running time and root-mean-square error (RMSE) of the considered estimators for the 4 collaborators case. The running time and RMSE are expressed relative to those of the classic EKF. The EKFB provides only 20% reduction in error in comparison with EKF in exchange for 9 times more running time. Although the SDP estimator requires 5 (4) times more running time than the EKF (EKFW), its performance is 5 (3) times better. A main difference between the SDP estimator and EKF is that in each time step, EKF only runs for 1 iteration, while the SDP estimator runs multiple iterations until the required accuracy is achieved. Although Kalman filters typically iterate once in each time step [10], multiple iterations can be used to achieve lower mean square error (which is sometimes called iterative extended Kalman filer or IEKF) [101]. Therefore, we increased the number of iterations per time step for the EKF and EKFW whose results are also provided in Table 6.1. To have a fair comparison, the number of iterations is increased such that all the estimators have the same running time. As can be seen, the performance of EKFW improves by increasing the number of iterations, although its performance is still inferior to the proposed SDP estimator. On the other hand, the performance of the EKF degrades by increasing the number of iterations. The reason is that the EKF tries to minimize the mean square error between the model and measurements. Since the NLOS biases are neglected in the EFK model, a lower mean square error does not necessarily lead to better localization performance.

6.5 Conclusion

In this work, cooperative source node tracking in non-line-of-sight (NLOS) environments was studied. Specifically, we demonstrated how robust estimation (specifically semidefinite programming or SDP) can be used to develop a tracking algorithm for both cooperative and noncooperative networks. The SDP-based estimator provides improved performance (i.e., reduced localization error)

Table 6.1: Relative running time and RMSE of the considered estimators to the classic EKF.

Estimator	Description	Run Time	RMSE
EKF	Classic EKF	1.0	1.00
EKFW	EKF w/ weighting	1.2	0.59
EKFB	EKF w/ bias tracking	8.8	0.82
SDP	Proposed SDP estimator	5.1	0.21
IEKF	Classic EKF w/ multiple iter.	5.1	1.21
IEKFW	EKF w/ weighting & multiple iter.	5.1	0.49

as compared to traditional and recently proposed tracking algorithms for noncooperative networks. More specifically, it was shown through computer simulations that the proposed SDP estimator provides superior performance in a simulated indoor network. Further, it was shown that cooperation among the source nodes can be very useful by providing additional connections between source nodes and improving performance over noncooperative techniques. Specifically, results showed that cooperation among source nodes can improve the localization performance by more than 33% in the proposed estimator for a typical indoor environment. It was also found that tracking the NLOS biases is not beneficial, as modeling NLOS propagation can be very difficult. Moreover, the NLOS biases change rapidly over time which makes them difficult to track. Therefore, in the proposed estimator, NLOS biases are estimated along with the source node state (velocity and location). However, the estimator does not *track* NLOS biases and instead treats them independently at each time steps.

Chapter 7

Conclusions

In this research, we addressed several open problems within the area of cooperative positioning and tracking. Although localization seems to be trivial where a simple trilateration method can be used, there are many factors which can impact the localization accuracy if they are not treated properly. These factors, which include but are not limited to non-line-of-sight (NLOS) propagation, multipath propagation, synchronization, and unknown propagation parameters, cause several nuisance parameters to appear in the system model. Nuisance parameters must be accounted for when localization is performed, even though we are not interested in their estimates.

Cooperative localization is a recent and promising technology that can be used to address the problem of dense networks with unknown source nodes to be localized. Cooperation was shown to be highly beneficial in networks with limited connectivity and also in indoor and harsh environments. It was shown that cooperation can significantly improve the localizability and localization performance. It was shown that semidefinite programming (SDP) can be used as a powerful tool to provide an alternative solution for the optimal, but nonlinear and nonconvex maximum-likelihood (ML) estimator. Unlike the ML estimator, convergence to the global minimum is guaranteed in SDP. It was shown that the ML estimator requires appropriate initialization which is difficult to determine in networks with limited connectivity. Moreover, the complexity of the ML solver grows exponentially as the number of unknown source nodes increases. It was shown that SDP can perform nearly as well as the optimal ML estimator, has yet considerately lower complexity.

In this work, we exploited the above two important concepts to deal with network localization in the presence of nuisance parameters. In particular, we have introduced and examined four problems. First, we examined cooperative received signal strength (RSS)-based localization in the presence of unknown propagation parameters. It was shown that the accuracy of RSS-based localization is dependent on strong knowledge of the propagation parameters such as the transmit powers of the source nodes and the path loss exponent of the environment which are assumed to be unknown in this work. Considering the propagation parameters as nuisance parameters in the estimation problem, we developed a cooperative RSS-based localization technique based on SDP. Simulation results showed that the performance of the proposed SDP estimator is only 12% worse than the optimal accuracy, while it requires 20% to 60% less running time than the ML estimator.

Second, we examined cooperative range-based localization in NLOS environments. NLOS propagation was examined under two different scenarios. In the first scenario, we assumed that the estimator is able to distinguish between NLOS and LOS connections. A novel SDP technique was introduced in which NLOS connections are used to improve the accuracy of location estimation. Cooperative localization was shown to be highly beneficial in severe NLOS environments, as the effect of NLOS propagation can be mitigated by providing additional measurements to the estimator. Unlike the ML estimator, the proposed SDP is also applicable in degenerate cases when the number of LOS connections is less than three. In the second scenario, we assumed that the estimator had no information about the statistics of the NLOS distribution. We derived a novel SDP technique to mitigate NLOS propagation. Simulation results showed that the proposed SDP estimator outperforms the previously considered algorithms in all NLOS situations in exchange for slightly greater running time. More specifically, the proposed SDP estimator provides more than 200% and 100% improvement (on average) over an estimator without mitigation and other NLOS mitigation techniques, respectively.

Third, we studied cooperative TOA-based sensor localization in asynchronous networks. We presented a framework for joint synchronization and localization for both cooperative and noncooperative networks. We derived a novel SDP estimator which considers clock parameters as nuisance variables and estimates them along with the source node locations. It was shown that the proposed SDP estimator provides position estimates as accurate as the ML estimation in both cooperative and noncooperative networks. Moreover, simulation results showed that as the number of source nodes increases the complexity of the ML estimator increases rapidly, while there is only a mild growth in the complexity of the proposed SDP estimator. In a simulated cooperative network, the proposed SDP estimator requires 3 times lower running time than the ML estimator in exchange for 10% performance degradation.

Lastly, we studied cooperative source node tracking in NLOS environments. We showed that robust estimation based on SDP can be useful for tracking mobile nodes in both cooperative and noncooperative networks. The proposed SDP estimator provides improved performance in comparison with the classic Kalman filter in severe NLOS environments. It was shown that cooperation among source nodes is highly beneficial by providing additional connections between source nodes and improving performance over noncooperative techniques. More specifically, results showed that the localization performance can be improved by more than 33% using cooperation in a simulated indoor environment.

Bibliography

[1] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero III, R. L. Moses, and N. S. Correal, "Locating the nodes: Cooperative localization in wireless sensor networks," *IEEE Signal Process. Mag.*, vol. 22, no. 4, pp. 54–69, July 2005.

[2] N. Patwari, A. O. Hero III, M. Perkins, N. S. Correal, and R. J. O'Dea, "Relative location estimation in wireless sensor networks," *IEEE Trans. Signal Process.*, vol. 51, no. 8, pp. 2137–2148, August 2003.

[3] X. Li, "Collaborative localization with received-signal strength in wireless sensor networks," *IEEE Trans. Veh. Technol.*, vol. 56, no. 6, pp. 3807–3817, November 2007.

[4] H. C. So and L. Lin, "Linear least squares approach for accurate received signal strength based source localization," *IEEE Trans. Signal Process.*, vol. 59, no. 8, pp. 4035–4040, August 2011.

[5] S. Venkatesh and R. M. Buehrer, "NLOS mitigation using linear programming in ultrawideband location-aware networks," *IEEE Trans. Veh. Technol.*, vol. 56, no. 5, pp. 3182–3198, September 2007.

[6] A. Catovic and Z. Sahinoglu, "The Cramer-Rao bounds of hybrid TOA/RSS and TDOA/RSS location estimation schemes," *IEEE Commun. Lett.*, vol. 8, no. 10, pp. 626–628, October 2004.

[7] L. Yang and K. C. Ho, "An approximately efficient TDOA localization algorithm in closed-form for locating multiple disjoint sources with erroneous sensor positions," *IEEE Trans. Signal Process.*, vol. 57, no. 12, pp. 4598–4615, December 2009.

[8] M. Sun and K. C. Ho, "An asymptotically efficient estimator for TDOA and FDOA positioning of multiple disjoint sources in the presence of sensor location uncertainties," *IEEE Trans. Signal Process.*, vol. 59, no. 7, pp. 3434–3440, July 2011.

[9] L. Cong and W. Zhuang, "Hybrid TDOA/AOA mobile user location for wideband CDMA cellular systems," *IEEE Trans. Wireless Commun.*, vol. 1, no. 3, pp. 439–447, July 2002.

[10] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory.* Upper Saddle River, NJ: Prentice-Hall, 1993.

[11] B. Sadler and R. Kozick, "A survey of time delay estimation performance bounds," in *Proc. IEEE SAM Workshop*, July 2006, pp. 282–288.

[12] T. Rappaport, *Wireless Communications: Principles and Practice*, 2nd ed. Upper Saddle River, NJ: Prentice-Hall, 2001.

[13] T. Jia and R. Buehrer, "A set-theoretic approach to collaborative position location for wireless networks," *IEEE Trans. Mobile Comput.*, vol. 10, no. 9, pp. 1264–1275, September 2011.

[14] W. H. Foy, "Position-location solution by Taylor-series estimation," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 12, no. 3, pp. 187–194, March 1976.

[15] L. A. Vandenberghe and S. B. Boyd, "Semidefinite programming," *SIAM Rev.*, vol. 38, no. 1, pp. 49–95, March 1996.

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

[17] J. F. Sturm, "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones," 1998.

[18] R. H. Tütüncü, K. C. Toh, and M. J. Todd, "Solving semidefinite-quadratic-linear programs using SDPT3," *Math. Program.*, vol. 95, no. 2, pp. 189–217, February 2003.

[19] M. Sun and K. C. Ho, "Successive and asymptotically efficient localization of sensor nodes in closed-form," *IEEE Trans. Signal Process.*, vol. 57, no. 11, pp. 4522–4537, November 2009.

[20] R. Ouyang, A.-S. Wong, and C.-T. Lea, "Received signal strength-based wireless localization via semidefinite programming: Noncooperative and cooperative schemes," *IEEE Trans. Veh. Technol.*, vol. 59, no. 3, pp. 1307–1318, Mar. 2010.

[21] P. Biswas, T.-C. Liang, K.-C. Toh, Y. Ye, and T.-C. Wang, "Semidefinite programming approaches for sensor network localization with noisy distance measurements," *IEEE Trans. Autom. Sci. Eng.*, vol. 3, no. 4, pp. 360–371, October 2006.

[22] Y.-T. Chan, H. Y. C. Hang, and P. chung Ching, "Exact and approximate maximum likelihood localization algorithms," *IEEE Trans. Veh. Technol.*, vol. 55, no. 1, pp. 10–16, Januray 2006.

[23] X. Sheng and Y.-H. Hu, "Maximum likelihood multiple-source localization using acoustic energy measurements with wireless sensor networks," *IEEE Trans. Signal Process.*, vol. 53, no. 1, pp. 44–53, Januray 2005.

[24] C. Meng, Z. Ding, and S. Dasgupta, "A semidefinite programming approach to source localization in wireless sensor networks," *IEEE Signal Process. Lett.*, vol. 15, pp. 253–256, 2008.

[25] K. W. Cheung, H. C. So, W.-K. Ma, and Y. T. Chan, "A constrained least squares approach to mobile positioning: Algorithms and optimality," *EURASIP J. Appl. Signal Process.*, pp. 1–23, 2006.

[26] C. Meesookho, U. Mitra, and S. Narayanan, "On energy-based acoustic source localization for sensor networks," *IEEE Trans. Signal Process.*, vol. 56, no. 1, pp. 365–377, Januray 2008.

[27] R. M. Vaghefi, M. R. Gholami, and E. G. Ström, "RSS-based sensor localization with unknown transmit power," in *Proc. IEEE ICASSP*, May 2011, pp. 2480–2483.

[28] A. M.-C. So and Y. Ye, "Theory of semidefinite programming for sensor network localization," *Math. Program., Series B*, vol. 109, no. 2–3, pp. 367–384, October 2007.

[29] G. Wang and K. Yang, "A new approach to sensor node localization using RSS measurements in wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 10, no. 5, pp. 1389–1395, May 2011.

[30] S. Kim, H. Jeon, and J. Ma, "Robust localization with unknown transmission power for cognitive radio," in *Proc. IEEE MILICOM*, October 2007, pp. 1–6.

[31] G. Wang and K. Yang, "Efficient semidefinite relaxation for energy-based source localization in sensor networks," in *Proc. IEEE ICASSP*, 2009, pp. 2257–2260.

[32] J. H. Lee and R. M. Buehrer, "Location estimation using differential RSS with spatially correlated shadowing," in *Proc. IEEE GLOBECOM*, Dec. 2009, pp. 1–6.

[33] M. R. Gholami, M. Rydström, , and E. G. Ström, "Positioning of node using plane projection onto convex sets," in *Proc. IEEE WCNC*, April 2010, pp. 1–5.

[34] C. Liu, K. Wu, and T. He, "Sensor localization with ring overlapping based on comparison of received signal strength indicator," in *Proc. IEEE MASS*, October 2004, pp. 516–518.

[35] Y. T. Chan and K. C. Ho, "A simple and efficient estimator for hyperbolic location," *IEEE Trans. Signal Process.*, vol. 42, no. 8, pp. 1905–1915, Aug. 1994.

[36] M. R. Gholami, S. Gezici, E. G. Ström, and M. Rydström, "Positioning algorithms for cooperative networks in the presence of an unknown turn-around time," in *Proc. IEEE SPAWC*, June 2011, pp. 166–170.

[37] J. Kivinen, X. Zhao, and P. Vainikainen, "Empirical characterization of wideband indoor radio channel at 5.3 GHz," *IEEE Trans. Antennas Propag.*, vol. 49, no. 8, pp. 1192–1203, August 2001.

[38] X. Li, "RSS-based location estimation with unknown pathloss model," *IEEE Trans. Wireless Commun.*, vol. 5, no. 12, pp. 3626–3633, December 2006.

[39] G. Golub and C. F. V. Loan, *Matrix computations*, 3rd ed. Johns Hopkins University Press, 1996.

[40] M. R. Gholami, S. Gezici, and E. S. Ström, "Improved position estimation using hybrid TW-TOA and TDOA in cooperative networks," *IEEE Trans. Signal Process.*, March 2010.

[41] C. A. Floudas and P. M. Pardalos, *Encyclopedia of Optimization*, 2nd ed. New York, NY: Springer-Verlag, 2009.

[42] S. Arora and B. Barak, *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009.

[43] P. Biswas, T.-C. Lian, T.-C. Wang, and Y. Ye, "Semidefinite programming based algorithms for sensor network localization," *ACM Trans. Sen. Netw.*, vol. 2, no. 2, pp. 188–220, May 2006.

[44] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 1.21," http://cvxr.com/cvx, May 2010.

[45] D. Torrieri, "Statistical theory of passive location systems," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 20, no. 2, pp. 183–198, Jul. 1984.

[46] T. Jia and R. M. Buehrer, "On the optimal performance of collaborative position location," *IEEE Trans. Wireless Commun.*, vol. 9, no. 1, pp. 374–383, January 2010.

[47] Y. Qi, H. Kobayashi, and H. Suda, "Analysis of wireless geolocation in a non-line-of-sight environment," *IEEE Trans. Wireless Commun.*, vol. 5, no. 3, pp. 672–681, March 2006.

[48] S. Venkatesh and R. M. Buehrer, "Non-line-of-sight identification in ultra-wideband systems based on received signal statistics," *IET Microw. Antennas Propag.*, vol. 1, no. 6, pp. 1120–1130, Dec. 2007.

[49] J. Riba and A. Urruela, "A non-line-of-sight mitigation technique based on ML-detection," in *Proc. IEEE ICASSP*, May 2004, pp. 153–156.

[50] Y.-T. Chan, W.-Y. Tsui, H.-C. So, and P. chung Ching, "Time-of-arrival based localization under NLOS conditions," *IEEE Trans. Veh. Technol.*, vol. 55, no. 1, pp. 17–24, January 2006.

[51] S. Venkatesh and R. M. Buehrer, "NLOS mitigation using linear programming in ultrawideband location-aware networks," *IEEE Trans. Veh. Technol.*, vol. 56, no. 5, pp. 3182–3198, September 2007.

[52] R. M. Vaghefi and R. M. Buehrer, "Cooperative sensor localization with NLOS mitigation using semidefinite programming," in *Proc. IEEE WPNC*, Mar. 2012, pp. 13–18.

[53] I. Güvenç, C.-C. Chong, F. Watanabe, and H. Inamura, "NLOS identification and weighted least-squares localization for uwb systems using multipath channel statistics," *EURASIP J. Adv. Signal Process*, vol. 2008, no. 1, pp. 1–14, January 2008.

[54] I. Guvenc and C.-C. Chong, "A survey on TOA based wireless localization and NLOS mitigation techniques," *IEEE Commun. Surveys Tuts.*, vol. 11, no. 3, pp. 107–124, 2009.

[55] G.-L. Sun and W. Guo, "Bootstrapping M-estimators for reducing errors due to non-line-of-sight (NLOS) propagation," *IEEE Commun. Lett.*, vol. 8, no. 8, pp. 509–510, August 2004.

[56] R. Casas, A. Marco, J. J. Guerrero, and J. Falcó, "Robust estimator for non-line-of-sight error mitigation in indoor localization," *EURASIP J. Appl. Signal Process.*, pp. 156–156, January 2006.

[57] Z. Li, W. Trappe, Y. Zhang, and B. Nath, "Robust statistical methods for securing wireless localization in sensor networks," in *Proc. IEEE Int. Symp. Information Processing in Sensor Networks (IPSN)*, April 2005, pp. 91–98.

[58] P. J. Huber and E. M. Ronchetti, *Robust Statistics*, 2nd ed. Hoboken, NJ: Wiley, 2009.

[59] T. Jia and R. M. Buehrer, "Collaborative position location with NLOS mitigation," in *Proc. IEEE International Workshop on Advances in Positioning and Location-Enabled Communications (APLEC) in conjunction with the IEEE PIMRC*, September 2010, pp. 267–271.

[60] X. Wang, Z. Wang, and B. O'Dea, "A TOA-based location algorithm reducing the errors due to non-line-of-sight (NLOS) propagation," *IEEE Trans. Veh. Technol.*, vol. 52, no. 1, pp. 112–116, January 2003.

[61] P. Tseng, "Second-order cone programming relaxation of sensor network localization," *SIAM J. on Optimization*, vol. 18, no. 1, pp. 156–185, February 2007.

[62] H. Chen, G. Wang, Z. Wang, H. So, and H. Poor, "Non-line-of-sight node localization based on semi-definite programming in wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 11, no. 1, pp. 108–116, Jan. 2012.

[63] M. Macconi, B. Morini, and M. Porcelli, "A Gauss-Newton method for solving bound-constrained underdetermined nonlinear systems," *Optimization Methods Software*, vol. 24, no. 2, pp. 189–217, April 2009.

[64] M. Rydstrom, E. Str om, and A. Svensson, "Robust sensor network positioning based on projections onto circular and hyperbolic convex sets (POCS)," in *Proc. IEEE SPAWC*, July 2006, pp. 1–5.

[65] I.-K. Rhee, J. Lee, J. Kim, E. Serpedin, and Y.-C. Wu, "Clock synchronization in wireless sensor networks: An overview," *Sensors*, vol. 9, no. 1, pp. 56–85, 2009.

[66] Y.-C. Wu, Q. Chaudhari, and E. Serpedin, "Clock synchronization of wireless sensor networks," *IEEE Signal Process. Mag.*, vol. 28, no. 1, pp. 124–138, 2011.

[67] R. T. Rajan and A.-J. van der Veen, "Joint ranging and clock synchronization for a wireless network," in *Proc. IEEE CAMSAP*, 2011, pp. 297–300.

[68] J. Zheng and Y.-C. Wu, "Joint time synchronization and localization of an unknown node in wireless sensor networks," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1309–1320, 2010.

[69] M. Sun and L. Yang, "On the joint time synchronization and source localization using TOA measurements," *Int. J. Distrib. Sens. N.*, pp. 1–11, 2013.

[70] Y. Huang, J. Benesty, G. W. Elko, and R. M. Mersereati, "Real-time passive source localization: A practical linear-correction least-squares approach," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 9, no. 8, pp. 943–956, 2001.

[71] R. M. Vaghefi and R. M. Buehrer, "Joint TOA-based sensor synchronization and localization using semidefinite programming," in *Proc. IEEE ICC*, Jun. 2014, pp. 520–525.

[72] M. Gholami, S. Gezici, and E. Strom, "TDOA based positioning in the presence of unknown clock skew," *IEEE Trans. Commun.*, vol. 61, no. 6, pp. 2522–2534, 2013.

[73] R. M. Vaghefi and R. M. Buehrer, "A linear estimator for joint synchronization and localization in wireless sensor networks," in *Proc. IEEE GLOBECOM*, Dec. 2014.

[74] G. C. Carter, "Coherence and time delay estimation," *Proc. IEEE*, vol. 75, no. 2, pp. 236–255, 1987.

[75] A. Quazi, "An overview on the time delay estimate in active and passive systems for target localization," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 29, no. 3, pp. 527–533, 1981.

[76] H. Wymeersch, J. Lien, and M. Z. Win, "Cooperative localization in wireless networks," *Proc. IEEE*, vol. 97, no. 2, pp. 427–450, 2009.

[77] Y. Shen and M. Z. Win, "Fundamental limits of wideband localization-Part I: a general framework," *IEEE Trans. Inf. Theory*, vol. 56, no. 10, pp. 4956–4980, 2010.

[78] Y. Shen, H. Wymeersch, and M. Z. Win, "Fundamental limits of wideband localizatio-Part II: cooperative networks," *IEEE Trans. Inf. Theory*, vol. 56, no. 10, pp. 4981–5000, 2010.

[79] S. P. Chepuri, G. Leus, and A.-J. van der Veen, "Joint localization and clock synchronization for wireless sensor networks," in *Proc. IEEE ASILOMAR*, 2012, pp. 1432–1436.

[80] M. Leng and Y.-C. Wu, "On clock synchronization algorithms for wireless sensor networks under unknown delay," *IEEE Trans. Veh. Technol.*, vol. 59, no. 1, pp. 182–190, 2010.

[81] R. M. Vaghefi, M. R. Gholami, R. M. Buehrer, and E. G. Ström, "Cooperative received signal strength-based sensor localization with unknown transmit powers," *IEEE Trans. Signal Process.*, vol. 61, no. 6, pp. 1389–1403, Mar. 2013.

[82] B. Thompson and R. M. Buehrer, "Characterizing and improving the collaborative position location problem," in *Proc. IEEE WPNC*, 2012, pp. 42–46.

[83] E. Xu, Z. Ding, and S. Dasgupta, "Source localization in wireless sensor networks from signal time-of-arrival measurements," *IEEE Trans. Signal Process.*, vol. 59, no. 6, pp. 2887–2897, Jun. 2011.

[84] K. Yang, G. Wang, and Z.-Q. Luo, "Efficient convex relaxation methods for robust target localization by a sensor network using time differences of arrivals," *IEEE Trans. Signal Process.*, vol. 57, no. 7, pp. 2775–2784, Jul. 2009.

[85] S. R. Drake and K. Dogancay, "Geolocation by time difference of arrival using hyperbolic asymptotes," in *Proc. IEEE ICASSP*, May 2004, pp. 361–364.

[86] R. M. Vaghefi and R. M. Buehrer, "Asynchronous time-of-arrival-based source localization," in *Proc. IEEE ICASSP*, May 2013, pp. 4086–4090.

[87] R. M. Vaghefi, J. Schloemann, and R. M. Buehrer, "NLOS mitigation in TOA-based localization using semidefinite programming," in *Proc. IEEE WPNC*, Mar. 2013, pp. 1–6.

[88] Y. Liu and Z. Yang, "Understanding node localizability of wireless Ad Hoc and sensor networks," *IEEE Trans. Mobile Comput.*, vol. 11, no. 8, pp. 1249–1260, 2012.

[89] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer networks*, vol. 38, no. 4, pp. 393–422, 2002.

[90] A. Amini, R. Vaghefi, J. de la Garza, and R. Buehrer, "GPS-free cooperative mobile tracking with the application in vehicular networks," in *Proc. IEEE WPNC*, Mar. 2014, pp. 1–6.

[91] R. M. Vaghefi and R. M. Buehrer, "Improving positioning in LTE through collaboration," in *Proc. IEEE WPNC*, Mar. 2014, pp. 1–6.

[92] ——, "Target tracking in NLOS environments using semidefinite programming," in *Proc. IEEE MILCOM*, Nov. 2013, pp. 169–174.

[93] U. Hammes and A. M. Zoubir, "Robust MT tracking based on M-estimation and interacting multiple model algorithm," *IEEE Trans. Signal Process.*, vol. 59, no. 7, pp. 3398–3409, 2011.

[94] J.-F. Liao and B.-S. Chen, "Robust mobile location estimator with NLOS mitigation using interacting multiple model algorithm," *IEEE Trans. Wireless Commun.*, vol. 5, no. 11, pp. 3002–3006, 2006.

[95] M. Nájar, J. M. Huerta, J. Vidal, and J. A. Castro, "Mobile location with bias tracking in non-line-of-sight," in *Proc. IEEE ICASSP*, 2004, pp. 956–959.

[96] X. R. Li and V. P. Jilkov, "Survey of maneuvering target tracking: III. measurement models," in *Proc. SPIE*, vol. 4473, 2001, pp. 423–446.

[97] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, 2002.

[98] R. S. Baheti, "Efficient approximation of kalman filter for target tracking," *IEEE Trans. Aerosp. Electron. Syst.*, no. 1, pp. 8–14, 1986.

[99] S. J. Fortune, D. M. Gay, B. W. Kernighan, O. Landron, R. A. Valenzuela, and M. H. Wright, "WISE design of indoor wireless systems: practical computation and optimization," *IEEE Comput. Sci. Eng.*, vol. 2, no. 1, pp. 58–68, 1995.

[100] R. M. Vaghefi and R. M. Buehrer, "Improving sensor tracking performance in NLOS environments using cooperation," in *IEEE ICC*, 2015.

[101] T. Lefebvre, H. Bruyninckx, and J. De Schutter, "Kalman filters for non-linear systems: a comparison of performance," *International Journal of Control*, vol. 77, no. 7, pp. 639–653, 2004.