Localization Algorithms for Passive Sensor Networks

by

Darya Ismailova
B.Eng., University of Astrakhan, 2010

A Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF APPLIED SCIENCE

in the Department of Electrical and Computer Engineering

Localization Algorithms for Passive Sensor Networks

by

Darya Ismailova
B.Eng., University of Astrakhan, 2010

Supervisory Committee

Dr. Wu-Sheng Lu, Supervisor
(Department of Electrical and Computer Engineering)

Dr. Pan Agathoklis, Departmental Member
(Department of Electrical and Computer Engineering)

Supervisory Committee

Dr. Wu-Sheng Lu, Supervisor
(Department of Electrical and Computer Engineering)

Dr. Pan Agathoklis, Departmental Member
(Department of Electrical and Computer Engineering)

ABSTRACT

In this thesis, we consider solving the problem of localizing a single radiating source based on range and range-difference measurements.

Contents

Supervisory Committee **ii**

Abstract **iii**

Table of Contents **iv**

List of Abbreviations **vii**

List of Tables **ix**

List of Figures **x**

1 Introduction **1**

1.1 The Localization Problem 1

1.2 Basic Localization Systems 3

1.2.1 Time of Arrival Method 3

1.2.2 Time Difference of Arrival Method 5

1.2.3 Angle of Arrival Method 5

1.2.4 Methods Based on Received Signal Strength 6

1.3 Contributions and Organization of the Thesis 7

1.3.1 Contributions of the Thesis 7

1.3.2 Organization of the Thesis 7

2 Iterative Re-Weighting Least-Squares Methods for Source Localization **9**

2.1 Source Localization Using Range Measurements 10

2.1.1 Problem Statement 10

2.1.2 LS Formulations and Review of Related Work 11

2.1.3 An Iterative Re-Weighting Approach 15

2.1.4 Numerical Results 21

2.2 Source Localization Using Range-Difference Measurements 25

2.2.1 Problem Formulation 25

2.2.2 Improved Solution Using Iterative Re-weighting 28

2.2.3 Numerical Results 31

2.3 Extensions 35

2.3.1 Acoustic Energy Attenuation Model and Problem Statement 36

2.3.2 Reformulation 37

3 Penalty Convex-Concave Procedure for Source Localization 39

3.1 Problem Statement and Review of Related Work 39

3.2 Fitting the Localization Problem into a CCP Framework 43

3.2.1 Basic Convex-Concave Procedure 43

3.2.2 Problem Reformulation 47

3.2.3 Imposing Error Bounds and Penalty Terms 49

3.2.4 The Algorithm 51

3.3 Numerical Results 53

4 Least Squares Localization by Sequential Convex Relaxation 58

4.1 Range-Difference Localization 58

4.1.1 Problem Formulation 58

4.1.2 Sequential Convex Relaxation 60

4.1.3 The Algorithm 64

4.1.4 Numerical Results 66

4.2 Range-based localization 68

4.2.1 Sequential Relaxation 68

4.2.2 Numerical Results 71

5 Conclusion 73

5.1 Conclusion 73

5.2 Future Work 73

A Matlab Code Listings 74

A.1 Iterative Re-Weighting LS for Source Localization Using Range Measurements . . . 74
A.1.1 IRWSR-LS Algorithm . . . 74
A.1.2 Hybrid IRWSR-LS . . . 76
A.2 Iterative Re-Weighting LS for Source Localization Using Range-Difference Measurements . . . 80
A.2.1 IRWSRD-LS Algorithm . . . 80
A.2.2 Hybrid IRWSRD-LS . . . 82
A.3 PCCP-Based LS Algorithm . . . 85
A.4 SCR-RDLS Algorithm . . . 87
A.5 SCR-RLS Algorithm . . . 88

Bibliography **90**

List of Abbreviations

AOA	Angle Of Arrival
AP	Access Point
CCP	Convex Concave Procedure
CRLB	Cramér-Rao lower bound
DC	Difference of Convex
DW-MDS	Distributed Weighted-Multi Dementional Scaling
GPS	Global Positioning System
GTRS	Generalized Trust Region Sub-problem
IRWSR-LS	Iterative Re-Weighting Squared Range-based Least Squares
IRWSRD-LS	Iterative Re-Weighting Squared Range-Difference-based Least Squares
LIDAR	Light Detection and Ranging
LOS	Line Of Sight
LP	Linear Program
LS	Least Squares
LTE	Long Term Evolution
MAC	Media Access Control
MDS	Multidimensional Scaling
ML	Maximum Likelihood
MSE	Mean Squared Error
NLLS	Non-Linear Least Squares
NLOS	Non-Line Of Sight
O-TDOA	Observed Time Difference Of Arrival
PDF	Probability Density Function
PCCP	Penalty Convex Concave Procedure
QP	Quadratic Programming
RSS	Received Signal Strength
SCP	Sequential Convex Programming
SCR-RLS	Sequential Convex Relaxation Range-based Least Squares
SCR-RDLS	Sequential Convex Relaxation Range-based Least Squares
SDP	SemiDefinite Programming

SLAM	Simultaneous Localization and Mapping
SMACOF	Scaling by MAjorizing a COmplicated Function
SOCP	Second Order Cone Programming
SPF	Standard Point Fixed
SR-LS	Squared-Range Least Squares
SRD-LS	Squared-Range-Difference Least Squares
SWLS	Sequential Weighted Least Squares
TDOA	Time Difference Of Arrival
TOA	Time Of Arrival
UWB	Ultra WideBand
WCDMA	Wide Band Code Division Multiple Access
WSR-LS	Weighted Squared Range-based Least Squares
WSRD-LS	Weighted Squared Range-Difference-based Least Squares

List of Tables

Table 2.1 MSE of position estimation for SR-LS, IRWSR-LS and *hybrid* IRWSR-LS methods 21

Table 2.2 Standard deviation of the squared estimation error for SR-LS, IRWSR-LS and *hybrid* IRWSR-LS methods 22

Table 2.3 MSE of position estimation for SRD-LS, IRWSRD-LS and *hybrid* IRWSRD-LS methods 32

Table 2.4 Standard deviation of the squared estimation error for SRD-LS, IRWSRD-LS and *hybrid* IRWSRD-LS methods 32

Table 3.1 Averaged MSE for SR-LS and PCCP methods 54

Table 4.1 MSE of position estimation for SRD-LS and SCR-RDLS methods 67

Table 4.2 Standard deviation of the squared position estimation error for SRD-LS and SCR-RDLS methods 67

Table 4.3 MSE of position estimation for SR-LS and SCR-RLS methods . 72

Table 4.4 Standard deviation of the squared position estimation error for SR-LS and SCR-RLS methods 72

List of Figures

Figure 1.1 Classical geolocation system. Range or angle information is extracted from received RF signals. Location is then estimated by lateration/angulation techniques [27]. 4

Figure 1.2 AOA positioning (angulation). The AOA estimate from 2 base stations to the mobile terminal can be used to estimate the position [27]. 6

Figure 2.1 Contours of the R-LS objective function over the region $\Re = \{\boldsymbol{x} : -6 \leq x_1 \leq 13, -10 \leq x_2 \leq 9\}$ 12

Figure 2.2 Histograms of the errors of the SR-LS (left) and IRWSR-LS (right) solutions, with standard deviation of noise $\sigma = 10^{-3}$ . . 23

Figure 2.3 Histograms of the errors of the SR-LS (left) and IRWSR-LS (right) solutions, with standard deviation of noise $\sigma = 10^{-2}$ . . 23

Figure 2.4 Histograms of the errors of the SR-LS (left) and IRWSR-LS (right) solutions, with standard deviation of noise $\sigma = 10^{-1}$ . . 24

Figure 2.5 Histograms of the errors of the SR-LS (left) and IRWSR-LS (right) solutions, with standard deviation of noise $\sigma = 1$ 33

Figure 2.6 Histograms of the errors of the SR-LS (left) and IRWSR-LS (right) solutions, with standard deviation of noise $\sigma = 10^{-1}$ . . 33

Figure 2.7 Histograms of the errors of the SR-LS (left) and IRWSR-LS (right) solutions, with standard deviation of noise $\sigma = 10^{-2}$ . . 34

Figure 2.8 Histograms of the errors of the SR-LS (left) and IRWSR-LS (right) solutions, with standard deviation of noise $\sigma = 10^{-3}$ . . 34

Figure 2.9 Histograms of the errors of the SR-LS (left) and IRWSR-LS (right) solutions, with standard deviation of noise $\sigma = 10^{-4}$ . . 35

Figure 3.1 An example of the CCP procedure (re-generated based on [17]). 46

Figure 3.2 MSE for different methods, various number of sensor nodes m and different noise levels with (a) $\sigma = 10^{-3}$, (b) $\sigma = 10^{-2}$, (c) $\sigma = 10^{-1}$, and (d) $\sigma = 1$. 55

Figure 3.3 Iteration path of the PCCP-based LS Algorithm and contours of the R-LS objective function over the region $\Re = \{\boldsymbol{x} : -15 \leq x_1 \leq 15, -25 \leq x_2 \leq 15\}$. The red cross indicates the location of the signal source. Sensors are located at $(6,4)^T$, $(0,-10)^T$, $(5,-3)^T$, $(1,-4)^T$ and $(3,-3)^T$. Large circles denote possible source locations given the noisy range reading at a particular sensor. 56

Figure 3.4 Convergernce of the proposed PCCP-based LS for random initializations with $\sigma = 10^{-1}$ for (a) 4 sensor nodes, (b) 5 sensor nodes, (c) 7 sensor nodes, and (d) 10 sensor nodes. 57

Figure 4.1 Range-difference localization. At least three base stations are required for the planar localization. The red cross indicates the location of the signal source. Sensors are placed at $\boldsymbol{a}_j = (10,-10)^T$, and $\boldsymbol{a}_0$ is the reference sensor. The time (range) differences $r_j - r_0$ and $r_i - r_0$ form two hyperboloids with focii located at $\boldsymbol{a}_i, \boldsymbol{a}_j$ and $\boldsymbol{a}_0$. Note that the hyperboloids are actually double sheeted, but for visual clarity only the halves which are part of the solution are shown. The intersection of these hyperboloids is the estimated position. The figure depicts the locus of possible source locations as one half of a two-sheeted hyperboloids. 59

Chapter 1

Introduction

1.1 The Localization Problem

Geolocation is a rapidly growing field due to the large impact it has on daily life, the general public has become increasingly dependent on it for realtime navigation, particularly on mobile smartphone devices. Location-based services are playing an increasingly important role in other fields such as teleconferencing, wireless communications, surveillance, and geophysics [10, 11, 12, 31, 37, 53, 54, 57, 63].

Different technologies have been used to build geolocation systems. Several global navigation satellite systems that are available for both military and civilian use and include GPS(US), Galileo(EU), GLONASS(Russia), BeiDou(China). Satellites are equiped with atomic clocks that provide a high-accuracy timing signal to allow a receiver to calculate the time that it takes the signal to reach the receiver. This information is used to calculate the position of the receiver by using a computational technique known as trilateration. The location of the receiver can be calculated from the tracked positions of the satellites and the times measured, each known as the Time-of-Arrival (TOA) [27]. GPS is widely used in navigation of vehicles such as airplanes, ships, and heavy equipment. It has also been used for monitoring movements rather than navigating, for example in fitness trackers such as FitBit [24] and Garmin[26] which automatically collect data about the user's activities (distance run, speed, routes taken).

As location-based services are becoming increasingly integrated into daily life, the demand for more accurate and robust localization technologies, including in GPS-denied areas, has incresed. One of the approaches to the problem was integration of various wireless technologies with GPS. One such solution is called Assisted GPS (A-GPS), which distributes data and processing over a network of cellular towers equipped with GPS-enabled servers [51]. This can greatly reduce the search space and time to first fix on location. Other systems were built purely on cellular signals, adapting the TOA techniques originally developed for GPS. However their accuracy was limited in urban environments due to obstruction of signals by buildings, and the technique did not gain widespread use beyond the Emergency-911 system it was developed for in the United States.

With the proliferation of smartphone devices and the internet, WiFi access points are becoming increasingly widespread, providing another means of determining location. One type of WiFi-based technique is called Received Signal Strength (RSS) location fingerprinting. It uses a database of WiFi signatures (RSS values and MAC addresses) with associated GPS coordinates. This has to be built up ahead of time, for example by surveying a city with a GPS and WiFi enabled vehicle. Realtime location of the mobile device is determined using the pattern-matching technique, that compares the measured RSSs between the mobile device and access points, and the RSS values stored in the database. Skyhook Wireless successfully developed such system that delivers accuracy of tens of meters in urban areas [56].

Design of any localization system naturally includes a trade-off between the overall performance and cost requirements. Some of the systems developed so far take advantage from the knowledge of the surrounding infrastructure such as cell towers, Wi-Fi hot spots, or installed RF tags [27]. Usually, the precision of the localization in such systems depends on the infrastructure. For example, hundreds or thousands of meters for cellular survey-based techniques, tens to hundreds of meters, or less than 100 meters for cellular triangulation techniques.

Indoor environments have additional challenges such as limited coverage, multipath signal fading and NLOS conditions. At the same time, indoor applications typically require greater accuracy due to the smaller spaces involved. Robotics was one early application area [23]. Systems such as CISCO Wireless Location Application [13] have applied RSS fingerprinting to indoor positioning, which works well if it is acceptable to install the required infrastructure and only coarse positioning is required. In searching for a method that would provide higher accuracy and be more robust in NLOS conditions, researches shifted their focus to Ultra-Wideband (UWB) communications which allow much more accurate timing information to be obtained. Another advantage of UWB is that it has increased bandwidth and helps prevent multipath signal fading [48].

Combining sensor data from multiple sources to improve localization results is another active area of research. Simultaneous Localization and Mapping (SLAM) is one such technique. The algorithm generates a map of the environment using RF signal, inertial sensors, images, Light Detection and Ranging (LIDAR) and sonar. A topological map is produced which can be used for navigation.

1.2 Basic Localization Systems

Many different systems have been developed for solving the localization problem in different domains such as geolocation, indoor positioning, sonar, etc. Despite the fact that the solutions themselves are different, they share some fundamental concepts. Classical non-survey based localization systems are generally comprised of two subsystems: range/angle estimation subsystem and lateration/ angulation subsystem [27]. The range/angle estimation subsystem determines distance or direction between the mobile object of unknown location and an array of reference points with known location either pre-programmed or obtained through GPS. The reference points can be satellites in GPS technology, or base stations in cellular localization, or a set of iBeacons, etc. Lateration/angulation subsystem uses the coordinates of the reference points and the distance or angle estimates to determine the unknown position. The accuracy of the positioning depends on the accuracy of the anchor nodes position, the quality of the range/angle estimates, network geometry, and the performace of the localization algorithm. Figure 1.1 illustrates this two-step procedure.

In the rest of this section, we provide an overview of the most popular ranging localization methods, namely TOA, TDOA, AOA, and RSS.

1.2.1 Time of Arrival Method

Time of Arrival (TOA) localization is a part of class of range-based localization techniques and is often reffered to as tri-lateration. It uses the fact that knowing the propagation speed of signals between a mobile device and reference points and by measuring the time that a signal takes to be received, the distance can easily be calculated. To obtain an n dimentional position estimate at least $n+1$ range measurements are required. Note that absolute times are used in calculations, so it is critical that the clocks in base stations and the mobile device are synchronized. It is

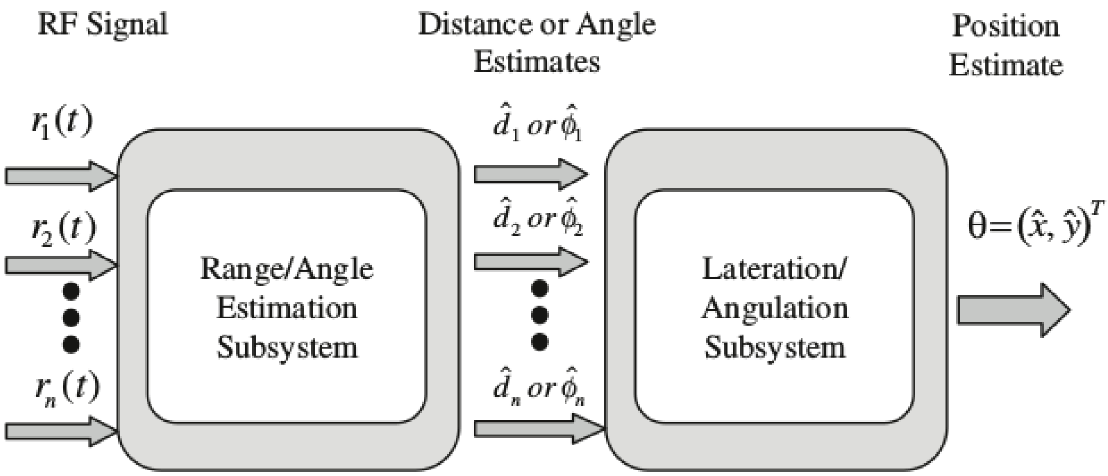


Figure 1.1: Classical geolocation system. Range or angle information is extracted from received RF signals. Location is then estimated by lateration/angulation techniques [27].

also assumed that they are all in LOS condition.

Solution techniques developed for range-based localization can be categorized as Maximum Likelihood (ML) and Least Squares (LS) approaches. Given the observed range measurements $\boldsymbol{r}_n = \boldsymbol{r} + \boldsymbol{w}$, the ML estimate $\hat{\boldsymbol{x}}$ of the unknown source location $\boldsymbol{x}$ is obtained by maximizing the conditional probability density function

$$\hat{\boldsymbol{x}} = \arg\max_{\boldsymbol{x}} P\left(\boldsymbol{r}_n | \boldsymbol{x}\right)$$

where $\boldsymbol{w}$ represents measurement noise. One of the major problems with ML approach is that it requires the knowledge of exact error-free measurements. Another difficulty is that solving for the unknown position extimate is computationally difficult. Many variants and approximations of the original ML has been developed [9], [28], [29].

In LS approaches, the source location extimate $\hat{\boldsymbol{x}}$ is found by minimizing the sum of risiduals [27]

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \left\{ \sum_{i=1}^{m} \beta_i \left(r_n^{(i)} - \|\boldsymbol{x} - \boldsymbol{a}_i\| \right)^2 \right\}$$

where $\boldsymbol{a}_i$ is a vector of known coordinates of reference points (sensors), $r_n^{(i)}$ is a noisy range measurement associated with it, β_i is a weight that can be used to emphasize the degree of confidence in the measurement, and m is the number of sensors. This problem is hard to solve in general because it is a nonconvex problem. Detailed analysis and discussion on solution methods for this problem will be given in Chapters 2, 3, and 4.

1.2.2 Time Difference of Arrival Method

A TDOA system is comprised of at least three base stations, one of which is the reference station. All stations are transmitting at precise time intervals, and the receiver measures the small differences in time that it takes the signals to arrive. From this information it is able to calculate its position. An important property of TDOA is that it requires base stations to be synchronized with each other, but it does not require the mobile receiver to be synchronized with the base stations. This technique first found application in World War II, when it was used by the British Royal Air Force for guiding airplanes at night. At that time synchronization betweeen measuring stations was not possible. Therefore TDOA techniques were feasible, while TOA were not. TDOA remains an important technique, finding modern application in 3G and LTE networks [15].

The conventional TDOA-based localization technique is a problem of solving a set of hyperbolic equations. More details can be found in Secs. 2.2 and 4.1 of the thesis.

1.2.3 Angle of Arrival Method

In the case of Angle of Arrival localization, the base stations are able to measure the angle of signal arrival with respect to an absolute reference. Knowing the location of base stations, the user position can be calculated as an intersection point of two lines that pass through the base stations [22]

$$\boldsymbol{x} = \begin{bmatrix} tan\phi_1 & -1 \\ tan\phi_2 & -1 \end{bmatrix}^{-1} \begin{bmatrix} x_1 tan\phi_1 & -y_1 \\ x_2 tan\phi_2 & -y_2 \end{bmatrix}$$

Figure 1.2 depicts the geometric principle of AOA technique. Even though AOA localization is relatively simple, it did not gain widespread usage because of poor performance in severe NLOS multipath scenarios. However, AOA has been used in hybrid positioning techniques along with TOA providing better performance than individual systems [7].

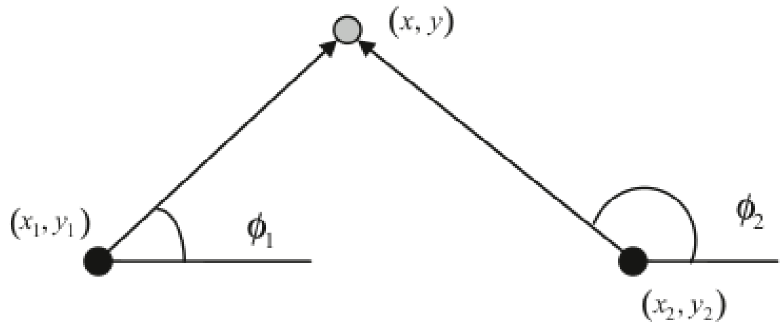


Figure 1.2: AOA positioning (angulation). The AOA estimate from 2 base stations to the mobile terminal can be used to estimate the position [27].

1.2.4 Methods Based on Received Signal Strength

The localization technique based on received signal strength is very similar to the TOA method in the sense that the estimate of the unknown position of the mobile device/radiating source is obtained from the range distance measurements between the sensors and the object. The difference is in the way a range measurement itself is obtained. In TOA method the range is obtained from the time reading through scaling, whereas in the RSS method the range is obtained from the received signal strength. The main idea that justifies this type of localization is that the strength of the RF signal attenuates with distance. The relationship between the RSS reading and the distance can be approximated by a log-normal attenuation model [38]

$$P_x(d) = P_0(d_0) - 10n_p \log_{10}\left(\frac{d_i}{d_0}\right) + X_\sigma$$

where $P_0(d_0)$ is a reference power in dB milliwatts at a reference distance d_0 away from the transmitter, n_p is the pathloss exponent governing the rate of power decay with distance, X_σ is the log-normal shadow fading component, and d_i is the distance between the mobile devices and the ith base station. Both σ and n_p are environment dependent. Given the model and model parameters, which are obtained via a priori measurements, the inter-sensor distances can be estimated from the RSS measurements. Localization algorithms can then be applied to these distance measurements to obtain estimated locations of sensors.

1.3 Contributions and Organization of the Thesis

1.3.1 Contributions of the Thesis

This thesis investigates new solution methods for single source localization problems that are potentially applicable to wireless sensor networks where range or range-difference measurements can be obtained utilizing RF signals. We remark that the methods and solution techniques proposed in the thesis are of general nature, and hence potentially useful in other scenarios that include localization problems for networks involving ultrasonic, sonic, or light sensors. In this thesis we focus on the least squares approaches for estimating the source location. Our models assume the availability of range or range-difference measurements and do not take into account the nonline-of-sight (NLOS) scenarios. Such simplifying assumptions are often made [10], [11] since mitigating the NLOS conditions is a separate research area. The main challenge facing the thesis research is that the localization problems we investigate are nonconvex problems whose global solutions are difficult to identify, although many approximate solutions [10], [37], [57], and data fusion methods [53] have been developed. The main contributions of the thesis can be summarised as follows:

- Development of a iterative re-weighting algorithm for range-based localization based on squared range-based least squares;
- Development of a iterative re-weighting algorithm for range-difference-based localization based on squared range-difference-based least squares;
- Development of a penalty convex-concave procedure (PCCP) for single source localization based on range measurements;
- Development of a sequential convex relaxation procedures for efficient computation of the LS estimate of source coordinates for range and range-difference localization.

1.3.2 Organization of the Thesis

The organization of the thesis is as follows:

Chapter 1 - Introduction

The purpose of this chapter is to introduce the systems and methods developed for the localization problems, discuss the motivations for improving the existing methods, and describe the main contributions and structure of the thesis.

Chapter 2 - Iterative Re-Weighting Least-Squares Methods for Source Localization

This chapter presents improved least squares methods that are iterative in nature. At each iteration the algorithms find a global solution to a subproblem that, as iterations proceed, approximates closely the LS solution.

Chapter 3 - Penalty Convex-Concave Procedure for Source Localization

In this chapter an algorithm is developed for finding an approximate solution to nonlinear least squares problem, for the case of range-based localization. We focus on the least squares formulation for the localization problem, where the l_2-norm of the residual errors is minimized in a setting known as difference-of-convex-functions programming. The problem at hand is then solved by applying a penalty convex-concave procedure in a successive manner.

Chapter 4 - Least Squares Localization by Sequential Convex Relaxation

This chapter adresses localization of a single radiating source based on range or range-difference measurements. In both cases the focus is on efficient computation of the least squares estimates of the source coordinates. For the case of range-difference, the central part of the procedure is a convex quadratic programming problem that needs to be solved in each iteration to provide an increment vector that updates the present iterate to next towards the solution of the localization problem at hand.

Chapter 5 - Conclusions

The chapter summarises the main results achieved in the thesis and suggests several problems for future studies in the area.

Chapter 2

Iterative Re-Weighting Least-Squares Methods for Source Localization

Locating a radiating source from range or range-difference measurements in a passive sensor network has recently attracted an increasing amount of research interest as it finds applications in a wide range of network-based wireless systems. Among the useful localization methods that have been documented over the years, least squares based algorithms constitute an important class of solution techniques as they are geometrically meaningful and often provide low complexity solution procedures with competitive estimation accuracy [4] - [57] . On the other hand, the error measure in a least squares (LS) formulation for the localization problem of interest is shown to be highly nonconvex, possessing multiple local solutions with degraded performance. There are many methods for continuous unconstrained optimization [1], however most of them are *local* methods that are sensitive to where the iteration begins, and give no guarantee to yield global solutions when applied to non-convex objective functions. In the case of source localization, this inherent feature of local methods is particular problematic because the source location is assumed to be entirely unknown and can appear practically anywhere, thus the chances to secure a good initial point for a local algorithm are next to none. For these reasons, various "global" localization techniques were investigated that are either non-iterative or insensitive to initial iterate. One representative in the class of global localization methods is the convex-relaxation based algorithm for range measurements proposed in [10], where the least squares model is relaxed to a semidefinite programming problem which is known to be convex [61], hence robust to where it starts. Another representative in this class is reference [4], where localization problems for range as well as range difference measurements are addressed by developing solution methods for *squared* range LS (SR-LS) and *squared* range difference LS (SRD-LS) problems. The methods proposed in [4] are non-iterative and the solutions obtained are proven to be the global minimizers of the respective SR-LS and SRD-LS problems, which are shown to be excellent estimates of the original LS solutions.

This chapter presents improved least squares methods that demonstrate improved localization performance when compared with some best known results from the literature. The key new ingredient of the proposed algorithms is an iterative procedure where the SR-LS (SRD-LS) algorithm is iteratively applied to a weighted sum of squared terms where the weights are carefully designed so that the iterates produced quickly converge to a solution which is found to be considerably closer to the original range-based (range-difference-based) LS solution.

2.1 Source Localization Using Range Measurements

2.1.1 Problem Statement

The source localization problem considered here involves a given array of m sensors specified by $\{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_m\}$ where $\boldsymbol{a}_i \in R^n$ contains the n coordinates of the ith sensor in space R^n. Each sensor measures its distance to a radiating source $\boldsymbol{x} \in R^n$. Throughout it is assumed that only noisy copies of the distance data are available, hence the *range measurements* obey the model

$$r_i = \|\boldsymbol{x} - \boldsymbol{a}_i\| + \varepsilon_i, i = 1, \ldots, m. \tag{2.1}$$

where ε_i denotes the unknown noise that has occurred when the ith sensor measures its distance to source $\boldsymbol{x}$. Let $\boldsymbol{r} = [r_1\ r_2 \ldots r_m]^T$ and $\boldsymbol{\varepsilon} = [\varepsilon_1\ \varepsilon_2 \ldots \varepsilon_m]^T$. The source localization problem can be stated as to estimate the exact source location $\boldsymbol{x}$ from the noisy range measurements $\boldsymbol{r}$. In the rest of this section, a least-squares (LS) formulation of the localization problem and two most relevant state-of-the-art solution methods are briefly reviewed; and a new method based on iterative re-weighting of squared range LS technique as well as a variant of the proposed method are then presented.

2.1.2 LS Formulations and Review of Related Work

Least squares approaches have proven effective for source localization problems [57] -[4]. For the localization problem at hand, the range-based least squares (R-LS) estimate refers to the solution of the problem

$$\underset{\boldsymbol{x}}{\text{minimize}}\, f(\boldsymbol{x}) \equiv \sum_{i=1}^{m} (r_i - \|\boldsymbol{x} - \boldsymbol{a}_i\|)^2 \tag{2.2}$$

The primary reason that justifies formulation (2.2) is its connection to the maximum-likelihood location estimation that determines $\boldsymbol{x}$ by examining the probabilistic model of the error vector $\boldsymbol{\varepsilon}$. Assuming the errors ε_i are independent and identically distributed (i.i.d.) Gaussian variables with zero mean and variance σ_i^2, then $\boldsymbol{\varepsilon}$ obeys a Gaussian distribution with zero mean and covariance $\boldsymbol{\Sigma} = \text{diag}(\sigma_1^2, \ldots, \sigma_m^2)$, and the maximum likelihood (ML) location estimator in this case is known to be

$$\boldsymbol{x}_{ML} = \arg \min_{\boldsymbol{x} \in R^n} (\boldsymbol{r} - \boldsymbol{g})^T \Sigma^{-1} (\boldsymbol{r} - \boldsymbol{g}) \tag{2.3}$$

where $\boldsymbol{g} = [g_1 \ g_2 \ldots \ g_m]^T$ with

$$g_i = \|\boldsymbol{x} - \boldsymbol{a}_i\| \tag{2.4}$$

It follows immediately that the ML solution in (2.3) is identical to the R-LS solution of problem (2.2) when covariance $\boldsymbol{\Sigma}$ is proportional to the identity matrix, i.e., $\sigma_1^2 = \ldots = \sigma_m^2$. In the literature this is known as the equal noise power case. For notation simplicity the method described in this chapter focuses on the equal noise power case, however the method developed below is also applicable to the unequal noise power case by working on a weighted version of the objective in (2.2) with $\{\sigma_i^{-2}, i = 1, \ldots, m\}$ as the weights.

Although many methods for unconstrained optimization are available [1], most of them are *local* methods in the sense they are sensitive to the choice of initial point where the iteration of an optimization algorithm begins. Especially when applied to a nonconvex objective function which possesses a number of local minimizers, unless a chosen local method starts at an initial point that happens to be sufficiently close to the (unknown) global minimizer, the solution obtained by the method gives no

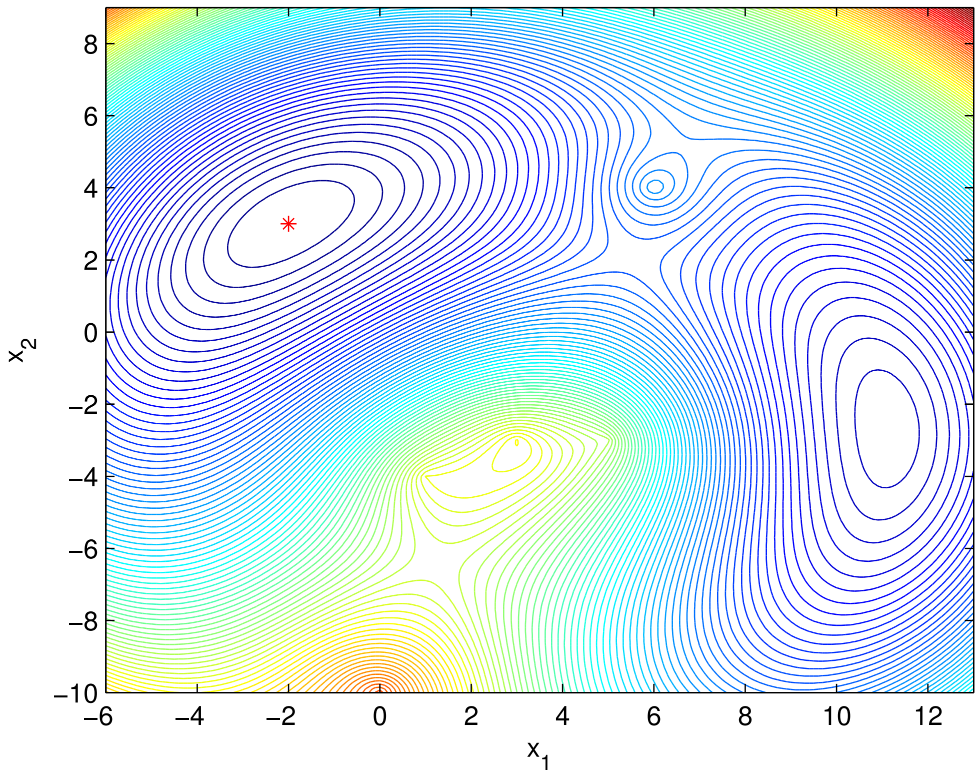


Figure 2.1: Contours of the R-LS objective function over the region $\Re = \{\boldsymbol{x} : -6 \leq x_1 \leq 13, -10 \leq x_2 \leq 9\}$

guaranty about global minimality. Unfortunately, the objective in (2.2) is highly nonconvex, possessing many local minimizers even for small-scale systems. As an example, consider an instance of the source localization problem on the plane $n = 2$ with five sensors $m = 5$ located at $(6, 4)^T, (0, -10)^T, (5, -3)^T, (1, -4)^T$ and $(3, -3)^T$ with the source emmitting the signal at $\boldsymbol{x}_s = (-2, 3)^T$. Figure 2.1 depicts a contour plot of the R-LS objective function in (2.2) over the region $\Re = \{\boldsymbol{x} : -6 \leq x_1 \leq 13, -10 \leq x_2 \leq 9\}$. It can be observed from the plot that there are two minimizers at $\tilde{\boldsymbol{x}} = (-1.9907, 3.0474)^T$ and $\widehat{\boldsymbol{x}} = (11.1152, -2.6785)^T$ with values of the objective $f(\tilde{\boldsymbol{x}}) = 0.1048$ and $f(\widehat{\boldsymbol{x}}) = 15.0083$ respectively. As expected, the global minimizer of R-LS objective offers a good approximation of the exact source location $\boldsymbol{x}_s$, but is unlikely to be precisely at point $\boldsymbol{x}_s$ because the objective $f(\boldsymbol{x})$ is defined using noisy range measurements. Note that for the exact source location $\boldsymbol{x}_s$ we have $f(\boldsymbol{x}_s) = \sum_{i=1}^{m} \varepsilon_i^2$.

Reference [10] addresses problem (2.2) by a convex relaxation technique where (2.2) is modified to a convex problem known as semidefinite programming (SDP) [61]. A key step in this procedure is to use (2.4) with g_i as new variables, which leads (2.2) to the constrained problem

$$\underset{\boldsymbol{x},\boldsymbol{g}}{\text{minimize}} \sum_{i=1}^{m} (r_i - g_i)^2 \tag{2.5a}$$

$$\text{subject to:} \quad g_i^2 = \|\boldsymbol{x} - \boldsymbol{a}_i\|^2, \quad i = 1, \ldots, m. \tag{2.5b}$$

By further defining matrix variables

$$\boldsymbol{G} = \begin{bmatrix} \boldsymbol{g} \\ 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{g}^T & 1 \end{bmatrix} \text{and } \boldsymbol{X} = \begin{bmatrix} \boldsymbol{x} \\ 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{x}^T & 1 \end{bmatrix} \tag{2.6}$$

and neglecting the rank constrains on $\boldsymbol{G}$ and $\boldsymbol{X}$, (2.5) can be reformulated in term of variables $\boldsymbol{G}$ and $\boldsymbol{X}$ as

$$\underset{\boldsymbol{X},\boldsymbol{G}}{\text{minimize}} \sum_{i=1}^{m} \left(G_{ii} - 2r_i G_{m+1,i} + r_i^2\right) \tag{2.7a}$$

$$\text{subject to:} \quad G_{ii} = Tr\left(\boldsymbol{C}_i \boldsymbol{X}\right), i = 1, \ldots, m \tag{2.7b}$$

$$\boldsymbol{G} \succeq 0, \boldsymbol{X} \succeq 0 \tag{2.7c}$$

$$G_{m+1,m+1} = G_{n+1,n+1} = 1 \tag{2.7d}$$

where

$$C_i = \begin{pmatrix} \boldsymbol{I}_{n\times n} & -\boldsymbol{a}_i \\ -\boldsymbol{a}_i^T & \|\boldsymbol{a}_i\|^2 \end{pmatrix} \quad i = 1, \ldots, m \tag{2.8}$$

which is a standard SDP problem that can be solved efficiently [61, 1]. Note that because (2.7) is a convex problem, global minimality of the solution is ensured regardless of the initial point used. On the other hand, however, because (2.7) is an approximation of the original problem in (2.2), the solution of (2.7) is only an approximate solution of problem (2.2). In what follows the solutions obtained by this SDP-relaxation based method will be referred to as SDR-LS solutions.

A rather different approach is recently proposed in [4] where the localization problem (2.2) is tackled by developing techniques that find global solution of the *squared* *range based LS* (SR-LS) problem

$$\underset{\boldsymbol{x}}{\text{minimize}} \sum_{i=1}^{m} \left(\|\boldsymbol{x}-\boldsymbol{a}_i\|^2 - r_i^2\right)^2 \tag{2.9}$$

By writing the objective in (2.9) as $\left(\alpha - 2\boldsymbol{a}_i^T\boldsymbol{x} + \|\boldsymbol{a}_i\|^2 - r_i^2\right)^2$ with $\alpha = \|\boldsymbol{x}\|^2$, it becomes a convex quadratic objective if one treats α as an additional variable and $\alpha = \|\boldsymbol{x}\|^2$ as a constraint. In this way, (2.9) is converted to the following constrained LS problem after necessary variable changes:

$$\underset{\boldsymbol{y}\in R^{n+1}}{\text{minimize}} \|\boldsymbol{A}\boldsymbol{y}-\boldsymbol{b}\|^2 \tag{2.10a}$$

$$\text{subject to:}\quad \boldsymbol{y}^T\boldsymbol{D}\boldsymbol{y} + 2\boldsymbol{f}^T\boldsymbol{y} = 0 \tag{2.10b}$$

where

$$\boldsymbol{y} = \begin{pmatrix} \boldsymbol{x} \\ \|\boldsymbol{x}\|^2 \end{pmatrix},\ \boldsymbol{A} = \begin{pmatrix} -2\boldsymbol{a}_1^T & 1 \\ \vdots & \vdots \\ -2\boldsymbol{a}_m^T & 1 \end{pmatrix},\ \boldsymbol{b} = \begin{pmatrix} r_1^2 - \|\boldsymbol{a}_1\|^2 \\ \vdots \\ r_m^2 - \|\boldsymbol{a}_m\|^2 \end{pmatrix} \tag{2.11}$$

$$\boldsymbol{D} = \begin{pmatrix} \boldsymbol{I}_{n\times n} & \boldsymbol{0}_{n\times n} \\ \boldsymbol{0}_{n\times n} & 0 \end{pmatrix},\ \boldsymbol{f} = \begin{pmatrix} \boldsymbol{0} \\ -0.5 \end{pmatrix}$$

This problem conversion, made in [4], turns out to be crucial as problem (2.10), which remains to be nonconvex because of the nonlinear equality constraint (2.10b), falls into the class of generalized trust region subproblems (GTRS) [25, 45] whose global solutions can be computed by exploring the KKT conditions which are both necessary and sufficient optimality conditions in this case [45].

We now conclude this section with a couple of remarks. First, an unconstrained version of (2.10) may be obtained by neglecting the constraint in (2.10b) as

$$\underset{\boldsymbol{y}\in R^{n+1}}{\text{minimize}} \|\boldsymbol{A}\boldsymbol{y}-\boldsymbol{b}\|^2 \tag{2.12}$$

whose solution, called *unconstrained squared-range-based LS* (USR-LS) estimate, is given by

$$\boldsymbol{y}^* = \left(\boldsymbol{A}^T\boldsymbol{A}\right)^{-1}\boldsymbol{A}^T\boldsymbol{b} \tag{2.13}$$

It is demonstrated by numerical experiments [4] that the SR-LS solution outperforms the USR-LS and, in many cases, SDR solutions. Second, the SR-LS solution, although it solves (2.9) exactly, lacks the statistical interpretation of the ML formulation. The SR-LS remains to be an approximate solution for the original problem in (2.2) and, as it was demonstrated by the numerical results in [3] and [5], provides less accurate estimates of the true source location, than the LS estimate. The method, described in detail below, tries to reduce the gap between the two solutions.

2.1.3 An Iterative Re-Weighting Approach

Iterative re-weighting least squares method is a popular technique used for solving problems involving the sums of norms. The method has found many applications, such as in robust regression [6, 47], sparse recovery [21], but the most relevand application for the current case is for solving the Fermat-Weber location problem. The Fermat–Weber problem has a long history and has been extensively studied in the field of optimization and location theory [6]. This problem can be stated as: Given m points $\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_m \in R^n$ called *anchors* and nonnegative weights $\omega_1, \omega_2, \ldots, \omega_m > 0$, find $\boldsymbol{x} \in R^n$ that minimizes the weighted sum of Euclidian distances between $\boldsymbol{x}$ and the m anchors:

$$\underset{\boldsymbol{x} \in R^n}{\text{minimize}} \sum_{i=1}^{m} \omega_i \|\boldsymbol{x} - \boldsymbol{a}_i\|.$$

Fermat–Weber problem is much easier to analyze and solve than the ML problem (2.2) because it is a well-structured nonsmooth convex minimization problem. The similarities between the Fermat–Weber problem and problem (2.2) have been noted and addressed in the literature [5] with a gradient method with a fixed step size, known as the standard fixed point (SFP) algorithm, to deal with problem (2.2). However, being a gradient method, likelihood for the SFP algorithm to converge to a local solution exists. Another method, also proposed in [5] and known as the the sequential weighted least squares algorithm (SWLS), is also an iterative method where each iteration involves solving a nonlinear least squares problem similar to (2.9). The SWLS method is found to be superior over SFP in terms of convergence rate and a wider region of convergence to the global minimum [5]. However, the possibility for SWLS to converge to a local minimum remains in certain sensor setups even if the initial point is constructed using a procedure developed specifically for SWLS. The method presented below takes an approach that is different from those described above in the sense that it does not require an initial point and the solution produced is guranteed to converge to a *global* solution.

Weighted squared range based least squares formulation

We now consider the weighted squared range based least squares (WSR-LS) problem

$$\underset{\boldsymbol{x}}{\text{minimize}} \sum_{i=1}^{m} w_i \left(\|\boldsymbol{x} - \boldsymbol{a}_i\|^2 - r_i^2 \right)^2 \tag{2.14}$$

which is obviously a weighted version of the SR-LS problem in (2.9). Following [4], it is rather straightforward to convert (2.14) into a GTRS similar to (2.10) as

$$\underset{\boldsymbol{y} \in R^{n+1}}{\text{minimize}} \, \|\boldsymbol{\Gamma} \left(\boldsymbol{A}\boldsymbol{y} - \boldsymbol{b} \right) \|^2 \tag{2.15a}$$

$$\text{subject to:} \quad \boldsymbol{y}^T \boldsymbol{D} \boldsymbol{y} + 2\boldsymbol{f}^T \boldsymbol{y} = 0 \tag{2.15b}$$

where $\boldsymbol{A}$, $\boldsymbol{b}$, $\boldsymbol{D}$, and $\boldsymbol{f}$ are defined in (2.11) and $\boldsymbol{\Gamma} = \text{diag}\left(\sqrt{w_1}, \ldots, \sqrt{w_m}\right)$. Clearly, problem (2.15) can be written as

$$\underset{\boldsymbol{y} \in R^{n+1}}{\text{minimize}} \, \|\boldsymbol{A}_w \boldsymbol{y} - \boldsymbol{b}_w\|^2 \tag{2.16a}$$

$$\text{subject to:} \quad \boldsymbol{y}^T \boldsymbol{D} \boldsymbol{y} + 2\boldsymbol{f}^T \boldsymbol{y} = 0 \tag{2.16b}$$

where $\boldsymbol{A}_w = \boldsymbol{\Gamma}\boldsymbol{A}$ and $\boldsymbol{b}_w = \boldsymbol{\Gamma}\boldsymbol{b}$. On comparing (2.16) with (2.10), if $S(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{D}, \boldsymbol{f})$ denotes a solver that produces the global solution of problem (2.10) for a given data set $\{\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{D}, \boldsymbol{f}\}$, then the same solver produces the global solution of the weighted problem (2.14) as long as it is applied to the data set $\{\boldsymbol{A}_w, \boldsymbol{b}_w, \boldsymbol{D}, \boldsymbol{f}\}$. We stress that the weights $\{w_i, i = 1, \ldots, m\}$ in (2.14) are *fixed* nonnegative constants.

Moving the SR-LS solution towards R-LS solution via iterative re-weighting

The main idea here is to use the weights $\{w_i, i = 1, \ldots, m\}$ to tune the objective in (2.14) toward the objective in (2.2) so that the solution obtained by minimizing such a WSR-LS objective is expected to be closer toward that of the problem (2.2). To substantiate the idea, we compare the ith term of the objective in (2.14) with its counterpart in (2.2), namely,

$$\underbrace{w_i \left(\|\boldsymbol{x} - \boldsymbol{a}_i\|^2 - r_i^2 \right)^2}_{\text{in (15)}} \leftrightarrow \underbrace{\left(\|\boldsymbol{x} - \boldsymbol{a}_i\| - r_i \right)^2}_{\text{in (2)}} \tag{2.17}$$

and write the term in (2.14) as

$$
\begin{aligned}
&w_i \left(\|\boldsymbol{x} - \boldsymbol{a}_i\|^2 - r_i^2\right)^2 = \\
&w_i \left(\|\boldsymbol{x} - \boldsymbol{a}_i\| + r_i\right)^2 \underbrace{\left(\|\boldsymbol{x} - \boldsymbol{a}_i\| - r_i\right)^2}_{\text{same as in (2)}}
\end{aligned}
$$

It follows that the objective in (2.14) would be the same as in (2.2) if the weights w_i were assigned to $1/\left(\|\boldsymbol{x} - \boldsymbol{a}_i\| + r_i\right)^2$. Evidently, weight assignments as such cannot be realized because w_i's must be fixed constants for (2.14) to be a globally solvable WSR-LS problem. A natural remedy to deal with this technical difficulty is to employ an iterative procedure whose kth iteration generates a global solution $\boldsymbol{x}_k$ of a WSR-LS sub-problem of the form

$$
\underset{x}{\text{minimize}} \sum_{i=1}^{m} w_i^{(k)} \left(\|\boldsymbol{x} - \boldsymbol{a}_i\|^2 - r_i^2\right)^2 \tag{2.18}
$$

where for $k \geq 2$ the weights $\{w_i^{(k)}, i = 1, \ldots, m\}$ are assigned using the previous iterate $\boldsymbol{x}_{k-1}$ as

$$
w_i^{(k)} = \frac{1}{\left(\|\boldsymbol{x}_{k-1} - \boldsymbol{a}_i\| + r_i\right)^2} \tag{2.19}
$$

while for $k = 1$ all weights $\{w_i^{(1)}, i = 1, \ldots, m\}$ are set to unity. Clearly the weights given by (2.19) are realizable. More importantly, when the iterates produced by solving (2.18) (namely $\boldsymbol{x}_k$ for $k = 1, 2, \ldots$) converge, in the kth iteration with k sufficiently large, the objective function of (2.18) in a small vicinity of its solution $\boldsymbol{x}_k$ is approximately equal to

$$
\begin{aligned}
&\sum_{i=1}^{m} w_i^{(k)} \left(\|\boldsymbol{x} - \boldsymbol{a}_i\|^2 - r_i^2\right)^2 \\
\approx &\sum_{i=1}^{m} w_i^{(k)} \left(\|\boldsymbol{x}_k - \boldsymbol{a}_i\|^2 - r_i^2\right)^2 \\
= &\sum_{i=1}^{m} w_i^{(k)} \left(\|\boldsymbol{x}_k - \boldsymbol{a}_i\| + r_i\right)^2 \left(\|\boldsymbol{x}_k - \boldsymbol{a}_i\| - r_i\right)^2 \\
\approx &\sum_{i=1}^{m} w_i^{(k)} \left(\|\boldsymbol{x}_{k-1} - \boldsymbol{a}_i\| + r_i\right)^2 \left(\|\boldsymbol{x}_k - \boldsymbol{a}_i\| - r_i\right)^2 \\
= &\sum_{i=1}^{m} \left(\|\boldsymbol{x}_k - \boldsymbol{a}_i\| - r_i\right)^2 \approx \sum_{i=1}^{m} \left(\|\boldsymbol{x} - \boldsymbol{a}_i\| - r_i\right)^2
\end{aligned}
$$

In words, with the weights from (2.19), the limiting point of the iterates produced by iteratively solving WSR-LS problem (2.18) is expected to be close to the global solution of problem (2.2).

The algorithmic steps of the proposed localization method are outlined as follows.

Algorithm 1

1) Input data: Sensor locations $\{\boldsymbol{a}_i, i = 1, \dots, m\}$, range measurements $\{r_i, i = 1, \dots, m\}$, maximum number of iterations k_{max} and convergence tolerance ζ.

2) Generate data set $\{\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{d}, \boldsymbol{f}\}$ as

$$\boldsymbol{A} = \begin{pmatrix} -2\boldsymbol{a}_1^T & 1 \\ \vdots & \vdots \\ -2\boldsymbol{a}_m^T & 1 \end{pmatrix}, \boldsymbol{b} = \begin{pmatrix} r_1^T - \|\boldsymbol{a}_1\|^T \\ \vdots \\ r_m^T - \|\boldsymbol{a}_m\|^T \end{pmatrix}$$

$$\boldsymbol{D} = \begin{pmatrix} \boldsymbol{I}_{n\times n} & \boldsymbol{0}_{n\times n} \\ \boldsymbol{0}_{n\times n} & 0 \end{pmatrix}, \boldsymbol{f} = \begin{pmatrix} \boldsymbol{0} \\ -0.5 \end{pmatrix}.$$

Set $k = 1, w_i^{(1)} = 1$ for $i = 1, \dots, m$.

3) Set $\boldsymbol{\Gamma}_k = \text{diag}\left(\sqrt{w_1^{(k)}}, \dots, \sqrt{w_m^{(k)}}\right)$, $\boldsymbol{A}_w = \boldsymbol{\Gamma}_k \boldsymbol{A}$ and $\boldsymbol{b}_w = \boldsymbol{\Gamma}_k \boldsymbol{b}$.

4) Solve the WSR-LS problem

$$\underset{x}{\text{minimize}} \sum_{i=1}^{m} w_i^{(k)} \left(\|\boldsymbol{x} - \boldsymbol{a}_i\|^2 - r_i^2\right)^2$$

by solving (2.16), i.e.

$$\begin{aligned} &\underset{\boldsymbol{y}\in R^{n+1}}{\text{minimize}} \, \|\boldsymbol{A}_w \boldsymbol{y} - \boldsymbol{b}_w\|^2 \\ &\text{subject to:} \quad \boldsymbol{y}^T \boldsymbol{D} \boldsymbol{y} + 2\boldsymbol{f}^T \boldsymbol{y} = 0 \end{aligned}$$

to obtain its global solution $\boldsymbol{x}_k$.

5) If $k = k_{max}$ or $\|\boldsymbol{x}_k - \boldsymbol{x}_{k-1}\| < \zeta$, terminate and output $\boldsymbol{x}_k$ as the solution; otherwise, set $k = k + 1$, update weights $\{w_i^{(k)}, i = 1, \dots, m\}$ using

$$w_i^{(k)} = \frac{1}{\left(\|\boldsymbol{x}_{k-1} - \boldsymbol{a}_i\| + r_i\right)^2}$$

and repeat from Step 3).

From the steps in Algorithm 1, it follows that the complexity of the algorithm is practically equal to the complexity of the WSR-LS solver involved in Step 4 times the number of iterations, k. Computer simulations have indicated that the algorithm converges with a small number of iterations, typically a $k \leq 6$ suffices. For simplicity, the solutions obtained from Algorithm 1 are called IRWSR-LS solutions. Technical details on how to solve (2.16) can be found in Appendix A.1.

A variant of Algorithm 1

As argued above, the IRWSR-LS solution from Algorithm 1 is expected to be an improved approximation of the global solution of R-LS problem in (2.2). However a small gap between the two solutions is inevitable owing to the approximate nature of the re-weighting strategy. In what follows we present a variant of Algorithm 1 that closes this gap by taking the IRWSR-LS solution as an initial point to run a good local unconstrained optimization algorithm for problem (2.2). The rationale behind this two-step approach is that the initial point produced in the first step by Algorithm 1 is highly likely within a sufficiently small vicinity of the exact global solution of problem (2.2), hence a good local method will lead it to the exact solution in a small number of iterations. We remark that such a "hybrid" approach is also expected to work with other "global" methods including the SDR-LS and SR-LS methods, but with a difference that employing an IRWSR-LS solution in the first step improves the closeness of the initial point, hence increases the likelihood of securing the exact global solution of problem (2.2) by a local method in the second step.

For the localization problem in question, the well-known Newton algorithm [1] is chosen as a local method because of its fast convergence and low complexity. We note that, unlike in a general scenario where the Newton algorithm is often considered numerically expensive because it requires to compute the inverse of the Hessian matrix, computing such an inverse is not costly in the present case because the dimension of the unknown vector $\boldsymbol{x}$ is extremely low: $n = 2$ or 3. Moreover, the Hessian matrix involved can be efficiently evaluated by a closed-form formula, as shown below.

To evaluate the Hessian of the objective $f(\boldsymbol{x})$ in (2.2), we assume $\boldsymbol{x} \neq \boldsymbol{a}_i$ for $i = 1, \ldots, m$, so that $f(\boldsymbol{x})$ is a smooth function of $\boldsymbol{x}$. The assumption simply means that the radiating source is away from the sensor at least by a certain distance, which appears to be reasonable for a practical localization problem. Under this circumstance, the gradient and Hessian of $f(\boldsymbol{x})$ are found to be

$$\boldsymbol{g}(\boldsymbol{x}) = \sum_{i=1}^{m} \left(1 - \frac{r_i}{\|\boldsymbol{x} - \boldsymbol{a}_i\|})\right) (\boldsymbol{x} - \boldsymbol{a}_i) \tag{2.20a}$$

and

$$\boldsymbol{H}(\boldsymbol{x}) = 2\left(\tau \boldsymbol{I} + \boldsymbol{H}_1(\boldsymbol{x})\right) \tag{2.20b}$$

respectively, where

$$\tau = m - \sum_{i=1}^{m} \frac{r_i}{\|\boldsymbol{x} - \boldsymbol{a}_i\|}$$

and

$$\boldsymbol{H}_1(\boldsymbol{x}) = \sum_{i=1}^{m} \frac{r_i(\boldsymbol{x} - \boldsymbol{a}_i)(\boldsymbol{x} - \boldsymbol{a}_i)^T}{\|\boldsymbol{x} - \boldsymbol{a}_i\|^3}.$$

To ensure a descent Newton step, the positive definiteness of the Hessian $\boldsymbol{H}(\boldsymbol{x})$ needs to be examined and, in case $\boldsymbol{H}(\boldsymbol{x})$ is not positive definite, to be modified to guatrantee its positive definiteness. To this end, the eigen-decomposition of $\boldsymbol{H}(\boldsymbol{x})$, namely,

$$\boldsymbol{H}(\boldsymbol{x}) = \boldsymbol{U} \boldsymbol{\Lambda} \boldsymbol{U}^T$$

may be performed, where $\boldsymbol{U}$ is orthogonal and $\boldsymbol{\Lambda} = \text{diag}\,(\tau + \ \lambda_1, \ldots, \tau + \lambda_n)$ with $\{\lambda_i, i = 1, \ldots, n\}$ being eigenvalues of $\boldsymbol{H}_1(\boldsymbol{x})$. Let l_{min} be the smallest eigenvalue of $\boldsymbol{H}(\boldsymbol{x})$, namely $l_{min} = \min\,(\tau + \lambda_1, \ldots, \tau + \lambda_n)$. If $l_{min} > 0$, then $\boldsymbol{H}(\boldsymbol{x})$ is positive definite and the Newton algorithm is carried out without modification; if $l_{min} \leq 0$, then the algorithm uses a slightly modified Hessian given by

$$\tilde{\boldsymbol{H}}(\boldsymbol{x}) = \boldsymbol{U} \tilde{\boldsymbol{\Lambda}} \boldsymbol{U}^T$$

where $\tilde{\boldsymbol{\Lambda}} = \text{diag}\left(\tilde{\lambda}_1, \ldots, \tilde{\lambda}_n\right)$

$$\tilde{\lambda}_i = \begin{cases} \tau + \lambda_i & \text{if } \tau + \lambda_i > 0 \\ \delta & \text{if } \tau + \lambda_i \leq 0 \quad i = 1, \ldots, m \end{cases}$$

and δ a small positive constant. Obviously, $\tilde{\boldsymbol{H}}(\boldsymbol{x})$ is guaranteed to be positive definite. The search direction in the kth iteration of the modified Newton algorithm is given by

$$\boldsymbol{d}_k = -\boldsymbol{U} \tilde{\boldsymbol{\Lambda}}^{-1} \boldsymbol{U}^T g(\boldsymbol{x}_k)$$

where $g(\boldsymbol{x}_k)$ is given by (2.20). In what follows, solutions obtained by the proposed two-step method are called *hybrid* IRWSR-LS solutions.

2.1.4 Numerical Results

Performance of the proposed algorithms was evaluated and compared with existing state-of-the-art methods by Monte Carlo simulations with a set-up similar to that of [4]. SR-LS solutions were used as performance benchmark for Algorithm 1 and its variant. Our simulation studies of Algorithm 1 and its variant considered a scenario that consists of $m = 5$ sensors $\{\boldsymbol{a}_i,\ i = 1, 2, \ldots, m\}$ randomly placed in the planar region in $[-15; 15] \times [-15; 15]$, and a radiating source $\boldsymbol{x}_s$, located randomly in the region $[-10; 10] \times [-10; 10]$. Coordinates of the source and sensors were generated for each dimension following a uniform distribution. The range measurements $\{r_i,\ i = 1, 2, \ldots, m\}$ were calculated using (2.1) and Step 4 of Algorithm 1 was implemented using the SR-LS algorithm proposed in [4]. Measurement noise $\{\varepsilon_i,\ i = 1, \ldots, m\}$ was modelled as i.i.d. Gaussian random variables with zero mean and variance σ^2, with σ being one of three possible levels $\{10^{-3}, 10^{-2}, 10^{-1}\}$. Accuracy of source location estimation was evaluated as the average of the squared position error $\|\boldsymbol{x}^* - \boldsymbol{x}_s\|^2$ where $\boldsymbol{x}_s$ denotes the exact source location and $\boldsymbol{x}^*$ is its estimation obtained by SR-LS, IRWSR-LS and hybrid-IRWSR-LS methods, respectively. Table 2.1 provides comparisons of these methods with SR-LS, where each table entry is a MSE averaged over 1,000 Monte Carlo runs of a given method for a given noise level. For the columns representing performance of the IRWSR-LS and *hybrid* IRWSR-LS methods each table entry lists their MSE and relative improvement over SR-LS solutions in percentage, in the format of MSE (% Improvement).

Table 2.1: MSE of position estimation for SR-LS, IRWSR-LS and *hybrid* IRWSR-LS methods

σ	SR - LS	IRWSR-LS (Im.,%)	*hybrid* IRWSR-LS (Im.,%)
1e-03	2.03251062e-06	1.19962894e-06 (41)	1.19949340e-06 (41)
1e-02	1.83717590e-04	1.24797437e-04 (32)	1.24812091e-04 (32)
1e-01	1.83611315e-02	1.22233840e-02 (33)	1.22139427e-02 (33)

It is observed that IRWSR-LS solutions offer considerable improvement over SR-LS solutions, and, as expected, in most cases hybrid IRWSR-LS solutions provide further but only incremental improvement. This is not surprising because the IRWSR-LS solutions themselves are already fairly close to the solutions of problem (2.2). It should also be noted again that for the exact source location $\boldsymbol{x}_s$ we have $f(\boldsymbol{x}_s) = \sum_{i=1}^{m} \varepsilon_i^2$. One might argue that the SR-LS solution already provides a rather good approximation for R-LS in the sense that SR-LS and IRWSR-LS (hybrid IRWSR-LS) have the same order of magnitude of the mean squared error. However, further analysis of the data that was used to generate Table 2.1 illustrates the advantage of the IRWSR-LS (hybrid IRWSR-LS) solution over the SR-LS.

Each entry in Table 2.2 is a standard deviation of the squared estimation errors aggregated over the same 1,000 Monte Carlo runs described above in Table 2.1 (where the MSE of the position estimation are shown). The results summarised in Table 2.2 demonstrate again that IRWSR-LS and hybrid IRWSR-LS outperform SR-LS. Figures 2.2 to 2.4 depict the histograms of the location estimation errors $\|\boldsymbol{x}^* - \boldsymbol{x}_s\|$ of the SR-LS solution (left images) and IRWSR-LS (right images) for all three noise levels with σ being one of $\{10^{-3}, 10^{-2}, 10^{-1}\}$, where $\boldsymbol{x}^*$ denotes the estimated location and $\boldsymbol{x}_s$ is the exact location of the source. Note that the histograms that correspond to the results obtained by IRWSR-LS are shifted closer towards 0 than those obtained by SR-LS and have smaller variance, and the solutions obtained by running IRWSR-LS have fewer outliers.

Table 2.2: Standard deviation of the squared estimation error for SR-LS, IRWSR-LS and *hybrid* IRWSR-LS methods

σ	SR - LS	IRWSR-LS	*hybrid* IRWSR-LS
1e-03	6.3438e-06	2.0843e-06	2.0864e-06
1e-02	3.2575e-04	2.0530e-04	2.0530e-04
1e-01	4.6998e-02	2.1377e-02	2.1377e-02

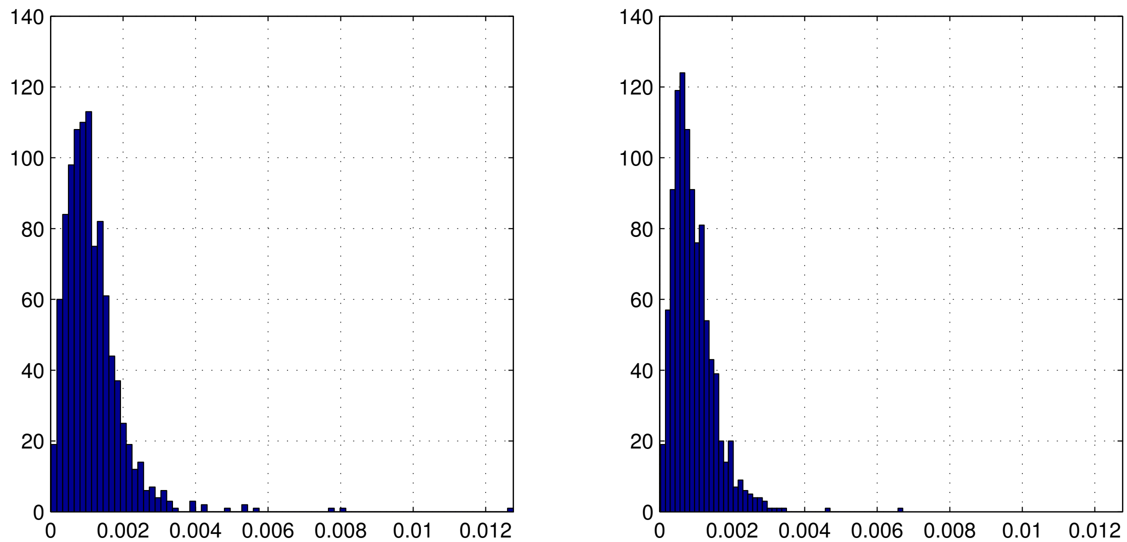


Figure 2.2: Histograms of the errors of the SR-LS (left) and IRWSR-LS (right) solutions, with standard deviation of noise $\sigma = 10^{-3}$

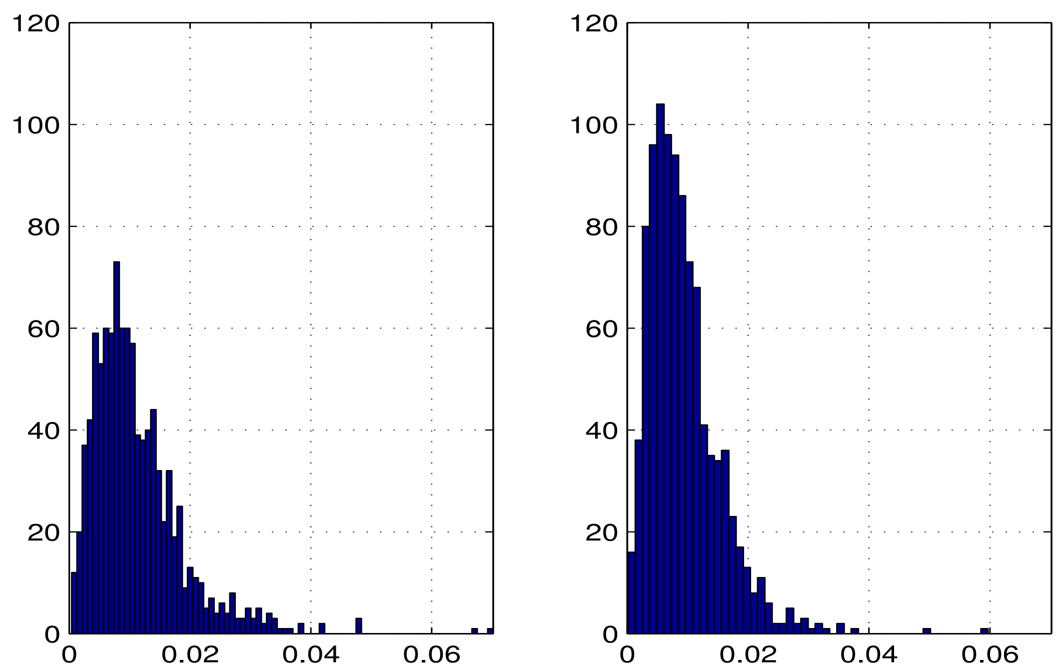


Figure 2.3: Histograms of the errors of the SR-LS (left) and IRWSR-LS (right) solutions, with standard deviation of noise $\sigma = 10^{-2}$

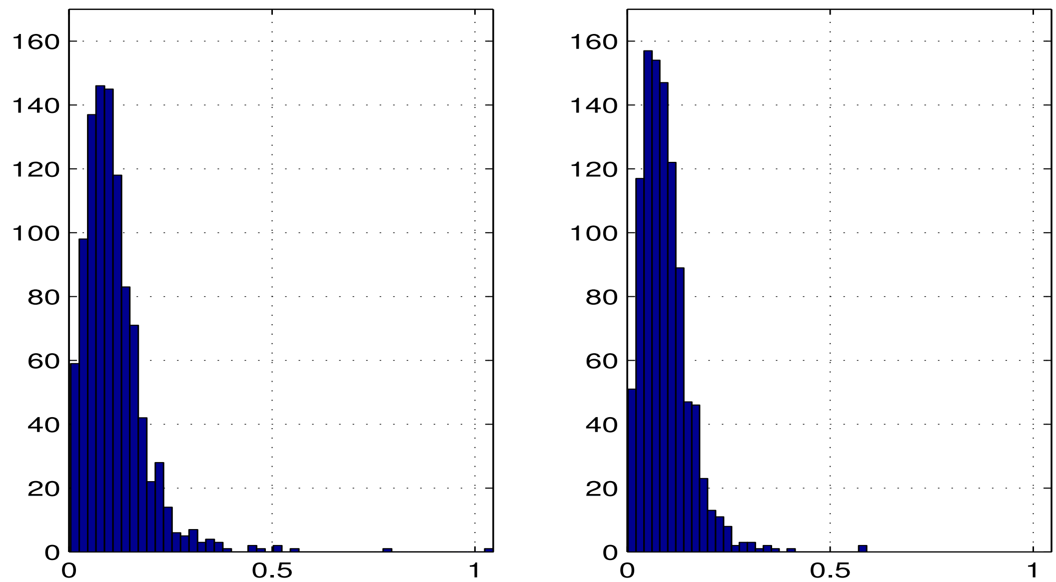


Figure 2.4: Histograms of the errors of the SR-LS (left) and IRWSR-LS (right) solutions, with standard deviation of noise $\sigma = 10^{-1}$

2.2 Source Localization Using Range-Difference Measurements

2.2.1 Problem Formulation

Another type of source localization problem that has attracted considerable attention is that of localizing a radiating source using range-difference measurements [60, 4]. In practice, range-difference measurements may be obtained from the time differences of arrival measured by an array of passive sensors. Time difference of arrival (TDOA) is a time-based positioning method based on the idea that the location of an active mobile unit (source of the signal) can be determined by examining the difference in time at which the signal arrives at multiple reference points. Adopting this technique is useful in practical scenarios where synchronization between mobile units is not available [27]. A typical example can be found in 3G (WCDMA) and LTE networks where the observed time difference of arrival (O-TDOA) technique is used to estimate the location of mobile units. We remark that in WCDMA networks, only the base stations are synchronized with each other, but mobile units are unsynchronized with base stations.

Each TDOA measurement constrains the location of the signal source to be on a hyperboloid with a constant range-difference between the two reference points. Specifically a TDOA measurement between base stations BS_i and BS_0 is given by

$$t_{i0} = (t_i - t_x) - (t_0 - t_x) = t_i - t_0$$

where t_x is the clock time of the mobile unit, t_i and t_0 are the time of arrival between the mobile unit and stations BS_i and BS_0 respectively. The above equation can be written in terms of distance (range-difference) through scaling

$$r_i = (t_i - t_0)c = r_i - r_0 = \|\boldsymbol{a}_i - \boldsymbol{x}\| - \|\boldsymbol{a}_0 - \boldsymbol{x}\|, i = 1, \ldots, m$$

where c is the speed of signal propagation, r_i is the distance from station $\boldsymbol{a}_i$ to source $\boldsymbol{x}$, r_0 is the distance from station $\boldsymbol{a}_0$ to source $\boldsymbol{x}$, and $\boldsymbol{a}_i \in R^n$ with $n = 2$ or 3, contains coordinates of the ith base station. Without loss of generality, the latter equation is valid with the assumption that the station BS_0 is placed at the origin of the coordinate system, i.e. $\boldsymbol{a}_0 = \boldsymbol{0}$ and used as a *reference* station [27].

The localization problem here is to estimate the location of a radiating source $\boldsymbol{x}$

given the locations of the $m+1$ sensors $\{\boldsymbol{a}_i, i=0,1,\ldots,m\}$ and noise-contaminated range-difference measurements $\{d_i, i=1,2,\ldots,m\}$ where

$$d_i = r_i + \varepsilon_i = \|\boldsymbol{a}_i - \boldsymbol{x}\| - \|\boldsymbol{x}\| + \varepsilon_i, \text{ for } i=1,2,\ldots,m \tag{2.21}$$

Therefore, the standard range-difference LS (RD-LS) problem is formulated as

$$\underset{\boldsymbol{x}\in R^n}{\text{minimize}}\, F(\boldsymbol{x}) = \sum_{i=1}^{m} \left(d_i + \|\boldsymbol{x}\| - \|\boldsymbol{x} - \boldsymbol{a}_i\|\right)^2 \tag{2.22}$$

Unfortunately, problem (2.22) is nonconvex and finding the global solution of (2.22) turns out to be a very hard problem. Nonlinear least squares (NLLS) algorithm [35] is widely used in TDOA localization systems for its performance. If the range measurement errors can be modeled as an additive white Gaussian noise, the accuracy of NLLS solution approaches the Cramér-Rao lower bound (CRLB). However, NLLS is not guaranteed to converge [8], [35] if the initial point is chosen far away from the actual source location. This becomes a more serious problem when the system coverage area is large since it becomes more difficult to secure an initial point that is close enough to the *unknown* source location. The Scaling by MAjorizing a COmplicated Function (SMACOF) strategy proposed in [49] can also be applied for position estimation. Compared with NLLS, it is not sensitive to the choice of the initial point and the mean-square error is guaranteed to decrease at each iteration, however the algorithm converges significantly slower. Reference [4] proposes a squared range-difference least squares (SRD-LS) approach to address this problem, which is summarized below.

By writing (2.21) as $d_i + \|\boldsymbol{x}\| = \|\boldsymbol{x} - \boldsymbol{a}_i\|$ and squaring both sides, we obtain

$$\left(d_i + \|\boldsymbol{x}\|\right)^2 = \|\boldsymbol{x} - \boldsymbol{a}_i\|^2 \tag{2.23}$$

which can be simplified to

$$-2d_i\|\boldsymbol{x}\| - 2\boldsymbol{a}_i^T\boldsymbol{x} = g_i, i=1,\ldots,m \tag{2.24}$$

where $g_i = d_i^2 - \|\boldsymbol{a}_i\|^2$. In practice (2.24) does not hold exactly due to measurement noise that contaminates the data d_i's. In other words, if d_i's in (2.24) are taken to be real-world data, then we only have

$$-2d_i\|\boldsymbol{x}\| - 2\boldsymbol{a}_i^T\boldsymbol{x} - g_i \approx 0, i=1,\ldots,m \tag{2.25}$$

Reference [4] proposes a LS solution for the problem at hand by minimizing the sum of squared residues on the left side of (2.25), namely,

$$\underset{\boldsymbol{x}\in R^n}{\text{minimize}} \sum_{i=1}^{m} \left(-2\boldsymbol{a}_i^T\boldsymbol{x} - 2d_i\|\boldsymbol{x}\| - g_i\right)^2 \tag{2.26}$$

By introducing new variable $\boldsymbol{y} = [\boldsymbol{x}^T \|\boldsymbol{x}\|]^T$ and noticing nonnegativity of the component y_{n+1} problem (2.26) is converted to

$$\underset{\boldsymbol{y}\in R^{n+1}}{\text{minimize}} \|\boldsymbol{B}\boldsymbol{y} - \boldsymbol{g}\|^2 \tag{2.27a}$$

$$\text{subject to: } \boldsymbol{y}^T\boldsymbol{C}\boldsymbol{y} = 0 \tag{2.27b}$$

$$y_{n+1} \geq 0 \tag{2.27c}$$

where

$$\boldsymbol{g} = \begin{pmatrix} g_1 \\ \vdots \\ g_m \end{pmatrix}, \ \boldsymbol{B} = \begin{pmatrix} -2\boldsymbol{a}_1^T & -2d_1 \\ \vdots & \vdots \\ -2\boldsymbol{a}_m^T & -2d_m \end{pmatrix}, \ \boldsymbol{C} = \begin{pmatrix} \boldsymbol{I}_n & \boldsymbol{0}_{n\times 1} \\ \boldsymbol{0}_{1\times n} & -1 \end{pmatrix} \tag{2.28}$$

Because of the presence of the nonnegativity constraint in (2.27c), (2.27) is no longer a GTRS problem hence the technique used for the case of range measurements does not apply. Nevertheless reference [4] presents a rigorous argument which shows that the optimal solution of (2.27) either assumes the form of

$$\tilde{\boldsymbol{y}}(\lambda) = \left(\boldsymbol{B}^T\boldsymbol{B} + \lambda\boldsymbol{C}\right)^{-1}\boldsymbol{B}^T\boldsymbol{g}$$

where λ solves

$$\tilde{\boldsymbol{y}}(\lambda)^T\boldsymbol{C}\tilde{\boldsymbol{y}}(\lambda) = 0 \tag{2.29}$$

and makes $\boldsymbol{B}^T\boldsymbol{B} + \lambda\boldsymbol{C}$ positive definite, or is the vector among $\{\boldsymbol{0}, \tilde{\boldsymbol{y}}(\lambda_1), \ldots, \tilde{\boldsymbol{y}}(\lambda_p)\}$ that gives the smallest objective function in (2.27a), where $\{\lambda_i, i = 1, \ldots, p\}$ are all roots of (2.29) such that the $(n+1)$'th component of $\tilde{\boldsymbol{y}}(\lambda_i)$ is nonnegative and $\boldsymbol{B}^T\boldsymbol{B} + \lambda\boldsymbol{C}$ has exactly one negative and n positive eigenvalues. We shall refer the global solution of (2.27) to as the SRD-LS solution.

2.2.2 Improved Solution Using Iterative Re-weighting

The Algorithm

We now present a method for improved solutions over SRD-LS solutions. The method incorporates an iterative re-weighting procedure into the SRD-LS approach, hence it is in spirit similar to the IRWRS-LS approach described in Sec. 2.1.2. We begin by considering the weighted squared range-difference least squares (WSRD-LS) problem

$$\underset{\boldsymbol{x}\in R^n}{\text{minimize}} \sum_{i=1}^{m} w_i \left(-2\boldsymbol{a}_i^T\boldsymbol{x} - 2d_i\|\boldsymbol{x}\| - g_i\right)^2 \tag{2.30}$$

where weights w_i for $i = 1, \ldots, m$ are fixed nonnegative constants. The counterpart of (2.27) for the problem (2.30) is given by

$$\underset{\boldsymbol{y}\in R^{n+1}}{\text{minimize}} \|\boldsymbol{B}_w\boldsymbol{y} - \boldsymbol{g}_w\| \tag{2.31a}$$

$$\text{subject to: } \boldsymbol{y}^T\boldsymbol{C}\boldsymbol{y} = 0 \tag{2.31b}$$

$$y_{n+1} \geq 0 \tag{2.31c}$$

where $\boldsymbol{g}_w = \boldsymbol{\Gamma}\boldsymbol{g}$, $\boldsymbol{B}_w = \boldsymbol{\Gamma}\boldsymbol{B}$, $\boldsymbol{\Gamma} = \text{diag}\{\sqrt{w_1}, \ldots, \sqrt{w_m}\}$, and $\boldsymbol{g}$, $\boldsymbol{B}$ are defined in (2.28). On comparing (2.31) with (2.27), it follows immediately that the global solver for problem (2.27) characterized by data set $\{\boldsymbol{B}, \boldsymbol{g}, \boldsymbol{C}\}$ can also be suited for solving problem (2.31) be used applying it to data set $\{\boldsymbol{B}_w, \boldsymbol{g}_w, \boldsymbol{C}\}$.

Concerning the assignment of weights $\{w_i, i = 1, \ldots, w_m\}$, we recall (2.23), (2.24) and observe that the ith term of the objective function in (2.30) can be written as

$$\begin{aligned} & w_i \left(-2d_i\|\boldsymbol{x}\| - 2\boldsymbol{a}_i^T\boldsymbol{x} - g_i\right)^2 \\ =& w_i \left((d_i + \|\boldsymbol{x}\|)^2 - \|\boldsymbol{x} - \boldsymbol{a}_i\|^2\right)^2 \\ =& w_i \left(d_i + \|\boldsymbol{x}\| + \|\boldsymbol{x} - \boldsymbol{a}_i\|\right)\left(d_i + \|\boldsymbol{x}\| - \|\boldsymbol{x} - \boldsymbol{a}_i\|\right) \end{aligned}$$

Clearly, the last expression above would become the ith term of the objective function in the RD-LS problem (2.22) if weights w_i were set to

$$\frac{1}{(d_i + \|\boldsymbol{x}\| + \|\boldsymbol{x} - \boldsymbol{a}_i\|)^2}$$

so that the first two factors are cancelled out. This suggests that a realizable weight assignment for performing practically the same cancellation can be made by means of iterative re-weighting for problems (2.30) and (2.31) where the weights in the kth iteration are assigned to

$$w_i^{(k)} = \frac{1}{(d_i + \|\boldsymbol{x}_{k-1}\| + \|\boldsymbol{x}_{k-1} - \boldsymbol{a}_i\|)^2}, i = 1, \ldots, m \tag{2.32}$$

Based on the analysis above, a localization algorithm for range-difference measurements can be outlined as follows.

Algorithm 2

1) Input data: Sensor locations $\{\boldsymbol{a}_i, i = 0, 1, \ldots, m\}$ with $\boldsymbol{a}_0 = \boldsymbol{0}$, range-difference measurements $\{d_i, i = 1, \ldots, m\}$, maximum number of iterations k_{max} and convergence tolerance ξ.

2) Generate data set $\{\boldsymbol{B}, \boldsymbol{g}, \boldsymbol{C}\}$ as

$$\boldsymbol{g} = \begin{pmatrix} d_1^2 - \|\boldsymbol{a}_1\|^2 \\ \vdots \\ d_m^2 - \|\boldsymbol{a}_m\|^2 \end{pmatrix}, \boldsymbol{B} = \begin{pmatrix} -2\boldsymbol{a}_1^T & -2d_1 \\ \vdots & \vdots \\ -2\boldsymbol{a}_m^T & -2d_m \end{pmatrix}, \boldsymbol{C} = \begin{pmatrix} \boldsymbol{I}_n & \boldsymbol{0}_{n\times 1} \\ \boldsymbol{0}_{1\times n} & -1 \end{pmatrix}.$$

Set $k = 1$, $w_i^{(1)} = 1$ for $i = 1, \ldots, m$.

3) Set $\boldsymbol{\Gamma}_k = \text{diag}\left(\sqrt{w_1^{(k)}}, \ldots, \sqrt{w_m^{(k)}}\right)$, $\boldsymbol{B}_w = \boldsymbol{\Gamma}_k \boldsymbol{B}$ and $\boldsymbol{g}_w = \boldsymbol{\Gamma}_k \boldsymbol{g}$.

4) Solve WSRD-LS problem

$$\begin{aligned} &\underset{\boldsymbol{y}\in R^{n+1}}{\text{minimize}} \; \|\boldsymbol{B}_w \boldsymbol{y} - \boldsymbol{g}_w\| \\ &\text{subject to: } \boldsymbol{y}^T \boldsymbol{C} \boldsymbol{y} = 0 \\ &\qquad\qquad\quad y_{n+1} \geq 0 \end{aligned}$$

to obtain its global solution $\boldsymbol{x}_k$.

5) If $k = k_{max}$ or $\|\boldsymbol{x}_k - \boldsymbol{x}_{k-1}\| < \xi$, terminate and output $\boldsymbol{x}_k$ as the solution; otherwise, set $k = k + 1$, update weights $\{w_i^{(k)}, i = 1, \ldots, m\}$ as

$$w_i^{(k)} = \frac{1}{(d_i + \|\boldsymbol{x}_{k-1}\| + \|\boldsymbol{x}_{k-1} - \boldsymbol{a}_i\|)^2}$$

and repeat from Step 3).

It is evident that the complexity of the algorithm is practically equal to the complexity of the WSRD-LS solver involved in Step 4 times the number of iterations, k, which is typically in the range of 3 to 6. We shall call the solutions obtained from Algorithm 2 IRWSRD-LS solutions. Technical details with regard to solving problem (2.31) can be found in Appendix 1.

A variant of Algorithm 2

Like the case of range measurements, once the IRWSRD-LS solution is obtained by applying Algorithm 2, which is expected to be within a small vicinity of the true global solution of the RD-LS problem (2.22), the gap can be closed by running a good local method that takes the IRWSRD-LS solution as an initial point. Again, the Newton method is chosen for fast convergence, low complexity due to the extremely low dimension n, and the availability of closed-form formulas to compute the gradient and Hessian of $F(\boldsymbol{x})$ in (2.22).

Assuming $\boldsymbol{x} \neq \boldsymbol{a}_i$ for $i = 0, 1, \ldots, m$, the gradient and Hessian of $F(\boldsymbol{x})$ is found to be

$$\boldsymbol{g}(\boldsymbol{x}) = \sum_{i=1}^{m} c_i \left(\boldsymbol{q}_i - \tilde{(}\boldsymbol{x}) \right)$$

and

$$\boldsymbol{H}(\boldsymbol{x}) = \sum_{i=1}^{m} \left[(\boldsymbol{q}_i - \tilde{(}\boldsymbol{x}))(\boldsymbol{q}_i - \tilde{(}\boldsymbol{x}))^T + c_i(\boldsymbol{Q}_{1i} + \boldsymbol{Q}_2) \right]$$

respectively, where

$$c_i = \|\boldsymbol{x} - \boldsymbol{a}_i\| - \|\boldsymbol{x}\|, \boldsymbol{q}_i = \frac{\boldsymbol{x} - \boldsymbol{a}_i}{\|\boldsymbol{x} - \boldsymbol{a}_i\|}, \tilde{\boldsymbol{x}} = \frac{\boldsymbol{x}}{\|\boldsymbol{x}\|}$$

and

$$\boldsymbol{Q}_{1i} = \frac{1}{\|\boldsymbol{x} - \boldsymbol{a}_i\|} \left(\boldsymbol{I} - \boldsymbol{q}_i \boldsymbol{q}^T \right), \boldsymbol{Q}_2 = \frac{1}{\|\boldsymbol{x}\|} \left(\boldsymbol{I} - \tilde{\boldsymbol{x}} \tilde{\boldsymbol{x}}^T \right)$$

To ensure the positive definiteness of Hessian, eigen-decomposition of $\boldsymbol{H}(\boldsymbol{x})$, namely,

$$\boldsymbol{H}(\boldsymbol{x}) = \boldsymbol{U} \boldsymbol{\Lambda} \boldsymbol{U}^T$$

is performed, where $\boldsymbol{U}$ is orthogonal and $\boldsymbol{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_n)$ with $\{\lambda_i, i = 1, \ldots, n\}$ being the eigenvalues of $\boldsymbol{H}(\boldsymbol{x})$. Let λ_{min} be the smallest eigenvalue of $\boldsymbol{H}(\boldsymbol{x})$. If λ_{min}, then $\boldsymbol{H}(\boldsymbol{x})$ is positive definite and the Newton algorithm is carried out without modification; if $\lambda_{min} \leq 0$, then the algorithm uses a slightly modified Hessian given by

$$\tilde{\boldsymbol{H}}(\boldsymbol{x}) = \boldsymbol{U}\tilde{\boldsymbol{\Lambda}}\boldsymbol{U}^T$$

where $\tilde{\boldsymbol{\Lambda}} = \operatorname{diag}\left(\tilde{\lambda}_1, \ldots, \tilde{\lambda}_n\right)$ with

$$\tilde{\lambda}_i = \begin{cases} \lambda_i & \text{if } \lambda_i > 0 \\ \delta & \text{if } \lambda_i \leq 0 \quad i = 1, \ldots, m \end{cases}$$

and δ a small positive constant. Obviously, $\tilde{\boldsymbol{H}}(\boldsymbol{x})$ is guaranteed to be positive definite. In what follows, solutions obtained by the proposed two-step method are called *hybrid* IRWSRD-LS solutions.

2.2.3 Numerical Results

Performance of the proposed algorithms was evaluated and compared with the method of [4] by Monte Carlo simulations with a set-up similar to that of [4]. SRD-LS solutions were used as performance benchmarks for Algorithm 2 and its variant. In both cases the system consisted of $m = 11$ sensors $\{\boldsymbol{a}_i,\ i = 1, 2, \ldots, 10\}$ with $\boldsymbol{a}_0 = \boldsymbol{0}$ and other ten sensors placed randomly placed in the planar region $[-15; 15] \times [-15; 15]$, and a radiating source $\boldsymbol{x}_s$, located randomly in the region $[-10; 10] \times [-10; 10]$. Coordinates of the source and sensors were generated for each dimension following a uniform distribution. The range-difference measurements used to form matrix $\boldsymbol{B}$ in Step 2 of the Algorithm 2 were calculated as noise-contaminated range-difference measurements d_i in (2.21). Measurement noise $\{\varepsilon_i,\ i = 1, \ldots, m\}$ was modelled as i.i.d. random variables with zero mean and variance σ^2, with σ being one of five possible levels $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1\}$. Step 4 of Algorithm 2 was carried out using the SRD-LS algorithm in [4]. Accuracy of source location estimation was evaluated in terms of mean squared error in the form $\text{MSE} = E\{\|\boldsymbol{x}^* - \boldsymbol{x}_s\|\}$ where $\boldsymbol{x}_s$ denotes the exact source location and $\boldsymbol{x}^*$ is its estimation obtained by SRD-LS, IRWSRD-LS and *hybrid* IRWSRD-LS methods, respectively. Table 2.3 provides comparisons of these methods with SRD-LS, where each entry was averaged MSE over 1,000 Monte Carlo runs of the method. For the columns representing performance of the IRWSDR-LS and *hybrid* IRWSDR-LS methods each table entry lists their MSE and relative improvement over SRD-LS solutions in percentage, in the format of MSE (% Improvement). Again, the IRWSRD-LS solutions offer considerable improvement over SRD-LS solutions. Further analysis of the data that was used to generate Table 2.3

illustrates the advantage of the IRWSR-LS (hybrid IRWSR-LS) solution over the SR-LS. Each entry in Table 2.4 is a standard deviation of the squared estimation errors aggregated over the same 1,000 Monte Carlo runs described above in Table 2.3 (where the MSE of the position estimation are shown). The results summarised in Table 2.4 suggest that, again, IRWSR-LS and hybrid IRWSR-LS outperform SR-LS.

Table 2.3: MSE of position estimation for SRD-LS, IRWSRD-LS and *hybrid* IRWSRD-LS methods

σ	SRD - LS	IRWSRD-LS (Im.,%)	*hybrid* IRWSRD-LS (Im.,%)
1e-04	1.38301598e-08	8.22705918e-09 (40)	8.22705918e-09 (40)
1e-03	1.60398717e-06	1.03880406e-06 (35)	1.038804061e-06 (35)
1e-02	1.11632818e-04	6.67785604e-05 (40)	6.67265316e-05 (40)
1e-01	1.20947651e-02	7.20891487e-03 (40)	7.07178346e-03 (41)
1e+0	1.57050323e+00	9.70756420e-01 (40)	7.86289933e-01 (48)

Table 2.4: Standard deviation of the squared estimation error for SRD-LS, IRWSRD-LS and *hybrid* IRWSRD-LS methods

σ	SRD - LS	IRWSRD-LS	*hybrid* IRWSRD-LS
1e-04	4.5624e-08	2.2446e-08	2.2446e-08
1e-03	3.9506e-06	3.1610e-06	3.1610e-06
1e-02	2.2710e-04	1.2812e-04	1.2812e-04
1e-01	3.0108e-02	1.8891e-02	1.8891e-02
1e+0	4.5781e+00	3.0597e+00	3.0597e+00

Figures 2.5 to 2.9 represent the histograms of the location estimation errors $\|\boldsymbol{x}^* - \boldsymbol{x}_s\|$ of the SR-LS solution (left images) and IRWSR-LS (right images) for all four noise levels with σ being one of $\{10^{-3}, 10^{-2}, 10^{-1}, 1\}$, where $\boldsymbol{x}^*$ denotes the estimated location and $\boldsymbol{x}_s$ is the exact location of the source. Histograms that correspond to the results obtained by IRWSRD-LS are shifted closer towards 0 than those obtained by SR-LS, have smaller variance, and in most cases solutions obtained by IRWSRD-LS have fewer outliers.

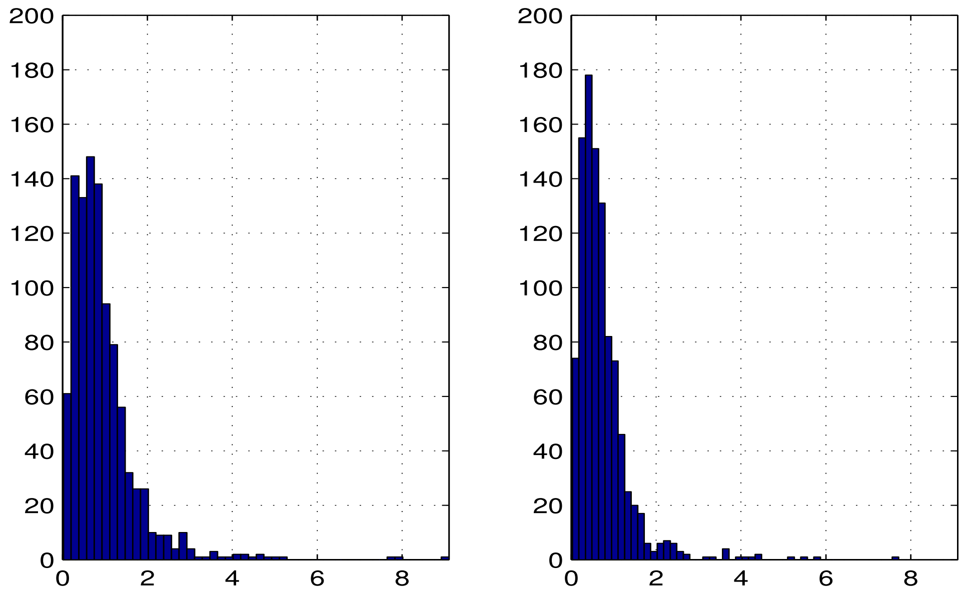


Figure 2.5: Histograms of the errors of the SR-LS (left) and IRWSR-LS (right) solutions, with standard deviation of noise $\sigma = 1$

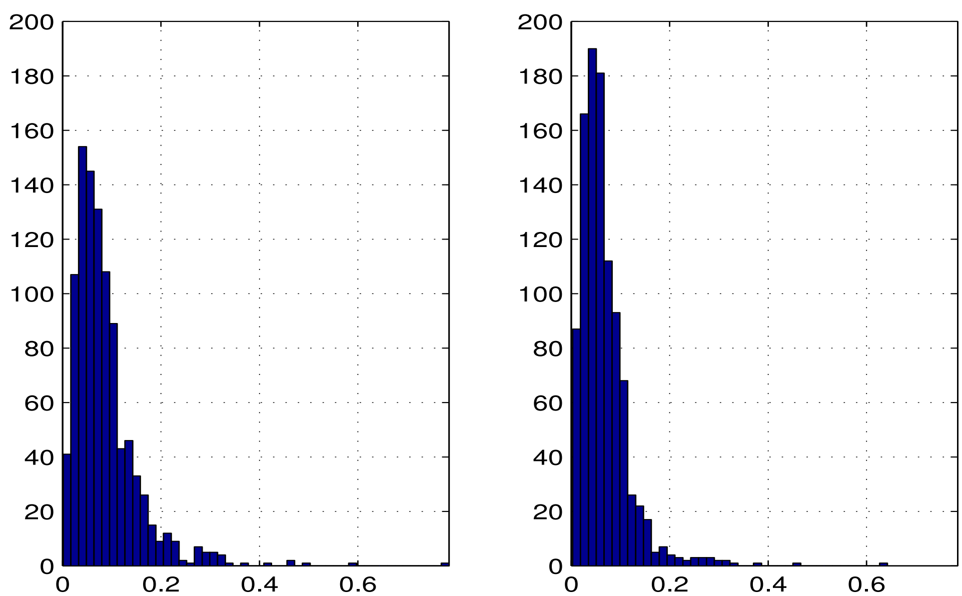


Figure 2.6: Histograms of the errors of the SR-LS (left) and IRWSR-LS (right) solutions, with standard deviation of noise $\sigma = 10^{-1}$

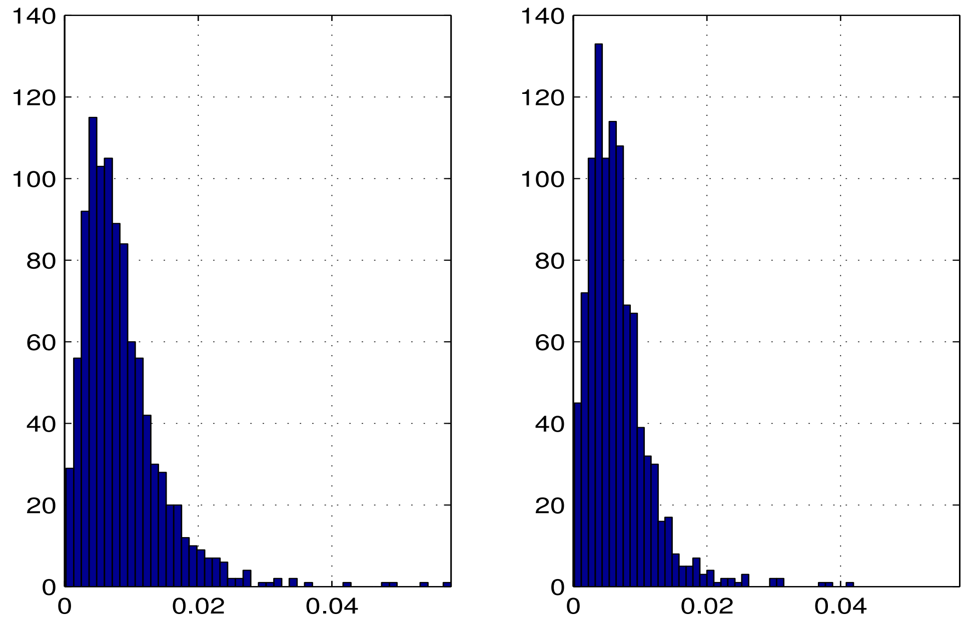


Figure 2.7: Histograms of the errors of the SR-LS (left) and IRWSR-LS (right) solutions, with standard deviation of noise $\sigma = 10^{-2}$

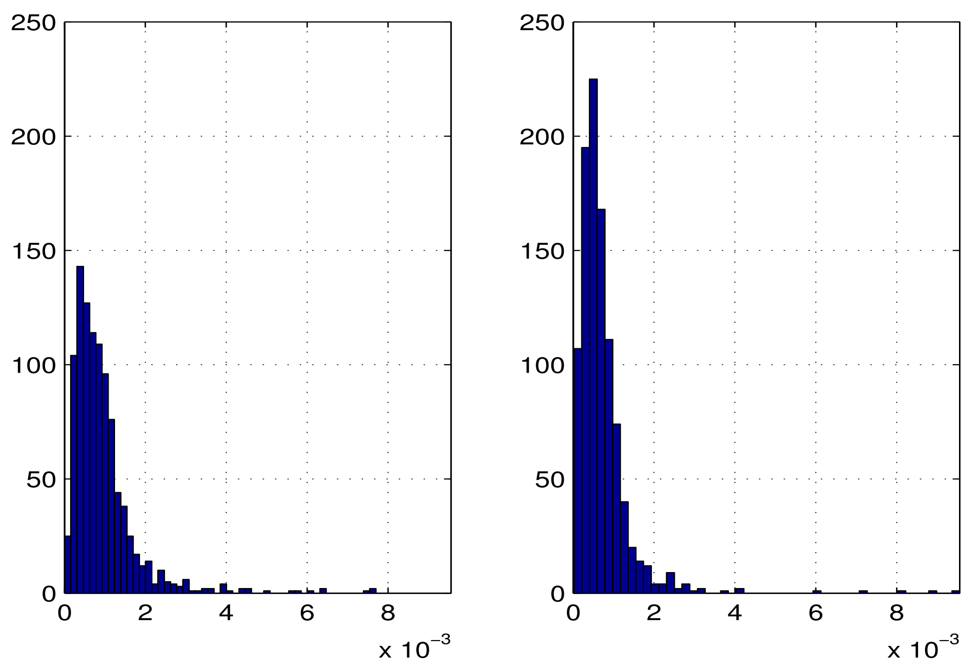


Figure 2.8: Histograms of the errors of the SR-LS (left) and IRWSR-LS (right) solutions, with standard deviation of noise $\sigma = 10^{-3}$

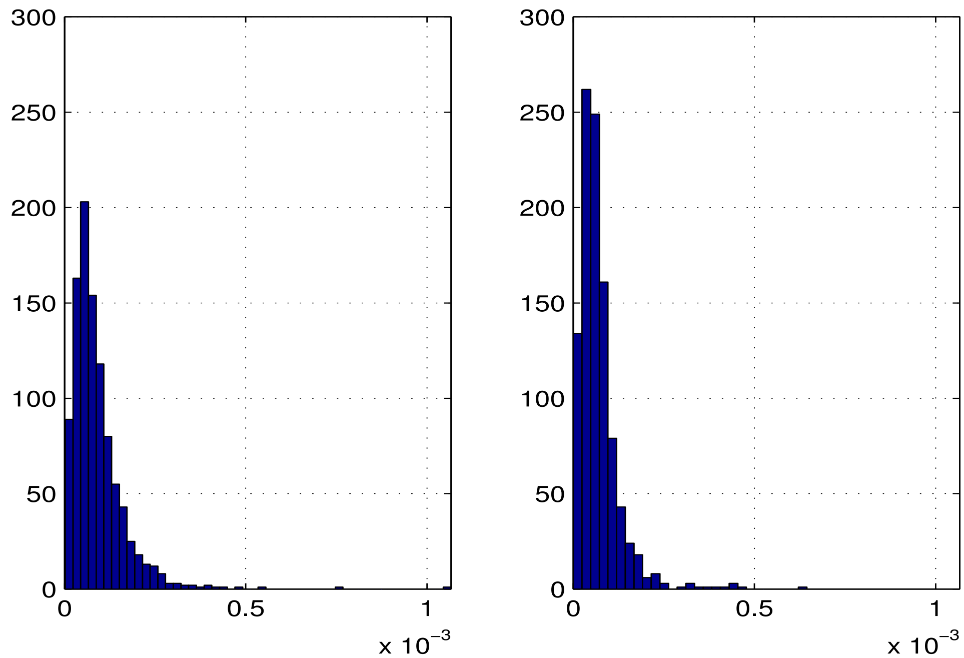


Figure 2.9: Histograms of the errors of the SR-LS (left) and IRWSR-LS (right) solutions, with standard deviation of noise $\sigma = 10^{-4}$

2.3 Extensions

Methods developed in this chapter for localization based on range measurements can also be adopted to solve the problem of single sourse localization using energy measurements [60]. Energy-based source localization, advocated in [37], [52], is motivated by the simple observation that the sound intensity received by a listener decreases when the distance between the sound source and the listener increases. By modeling the relation between sound intensity (energy) and distance from the sound source, one may estimate the source location using multiple energy readings at different known sensor locations. It is known that when the sound is propagating through the air, the acoustic energy emitted omnidirectionally from a sound source will attenuate at a rate that is inversely proportional to the square of the distance [37]. Using this fact and some simple manipulations, it is possible to obtain an equation in the vector of unknow source location $\boldsymbol{x}$ that is somewhat similar to (2.9). The rest of the section provides technical details of this reformulation.

2.3.1 Acoustic Energy Attenuation Model and Problem Statement

Let m be a number of acoustic sensors involved. For consistency of notation, let $\boldsymbol{a}_i$ denote the known location of the sensor i in space R^n with $n = 2$ or 3. Each sensor measures the acoustic intensity radiated by a source $\boldsymbol{x} \in R^n$ over a time period $T = \frac{M}{f_s}$, where M is the number of sample points used for estimating the acoustic enery and f_s is the sampling frequency. Acoustic energy received by sensor i over a time period T can be represented as

$$r_i = g_i \frac{S}{\|\boldsymbol{x} - \boldsymbol{a}_i\|^\alpha} + \varepsilon_i \tag{2.33}$$

where $\|\boldsymbol{x} - \boldsymbol{a}_i\|$ is the Euclidean distance between the ith sensor and the source, and g_i is a scaling factor that takes into account ith sensor gain. It is asssumed that the gain of individual sensors is either known, i.e. obtained at the sensor callibration stage, or is same for all sensors. In (2.33) S is the unknown acoustic energy measured 1 unit distance away from the source, α is the energy decay factor and is usually assumed to have a value 2 [37], and ε_i denotes the square of the background noise affecting the measurement of sensor i. If the number of sample points M used for estimating the acoustic energy is large, the error ε_i can be approximated well as a normal distribution with positive mean μ_i and variance σ_i^2 [37], [55]. More details on derivation and assumptions of this model can be found in [37] - [39], [52] and the references therein.

References [37, 55] argue that the maximum likelihood estimation of unknown parameters $\boldsymbol{\theta} = [\boldsymbol{x}^T S]^T$ can be obtained by solving the minimization problem

$$\underset{\theta}{\text{minimize}} \quad \ell(\boldsymbol{\theta}) = \|\boldsymbol{Z} - S\boldsymbol{H}\| \tag{2.34}$$

where

$$\boldsymbol{H} = \begin{pmatrix} \frac{g_1}{\sigma_1\|\boldsymbol{x}-\boldsymbol{a}_1\|^2} \\ \frac{g_2}{\sigma_2\|\boldsymbol{x}-\boldsymbol{a}_2\|^2} \\ \vdots \\ \frac{g_m}{\sigma_m\|\boldsymbol{x}-\boldsymbol{a}_m\|^2} \end{pmatrix}, \ \boldsymbol{Z} = \begin{pmatrix} \frac{y_1-\mu_1}{\sigma_1} \\ \frac{y_2-\mu_2}{\sigma_2} \\ \vdots \\ \frac{y_m-\mu_m}{\sigma_m} \end{pmatrix}$$

and $\boldsymbol{Z}$ are (estimated) normalized energy measurements for the case of the single radiating source. We remark that μ_i and σ_i in (2.34) are assumed to be known.

2.3.2 Reformulation

There are two things to note about (2.34). First, (2.34) involves a *nonlinear* least square objective function because the vector $\boldsymbol{H}$ is a nonlinear function of the n unknown source coordinates, where n is the dimention of the location coordinates. Second, there are m sensors reporting the acousting readings and there are a total of $n+1$ unknowns with $n+1 \leq m$, including the unknown acoustic energy S radiated from the source. To eliminate the unknown source energy S from formultion, reference [55] propose first to compute the ratio k_{ij} of the calibrated energy readings from ith and jth sensor as

$$k_{ij} = \left(\frac{z_i/g_i}{z_j/g_j}\right)^{-1/\alpha} = \frac{\|\boldsymbol{x} - \boldsymbol{a}_i\|}{\|\boldsymbol{x} - \boldsymbol{a}_j\|} \tag{2.35}$$

for $i = 1, 2, \ldots, m-1$, and $j = i+1, \ldots, m$. For the case $0 < k_{ij} \neq 1$ all possible source coordinates $\boldsymbol{x}$ that form a solution to (2.35) reside on an n-dimensional hyper-sphere described by the equation:

$$\|\boldsymbol{x} - \boldsymbol{c}\|^2 = \rho_{ij}^2 \tag{2.36}$$

where the center $\boldsymbol{c}_{ij}$ and the radius ρ_{ij} of the hyper-sphere associated with the sensors i and j are given by

$$\boldsymbol{c}_{ij} = \frac{\boldsymbol{a}_i - k_{ij}^2 \cdot \boldsymbol{a}_j}{1 - k_{ij}^2}, \ \rho_{ij} = \frac{k_{ij}\|\boldsymbol{a}_i - \boldsymbol{a}_j\|}{1 - k_{ij}^2} \tag{2.37}$$

For the case when $k_{ij} \to 1$, the possible source locations $\boldsymbol{x}$ reside on the hyperplane between sensors $\boldsymbol{a}_i$ and $\boldsymbol{a}_j$, i.e.:

$$\boldsymbol{x}^T \boldsymbol{\gamma}_{ij} = \tau_{ij}$$

where $\boldsymbol{\gamma}_{ij} = \boldsymbol{a}_i - \boldsymbol{a}_j$ and $\tau_{ij} = (\|\boldsymbol{a}_i\|^2 - \|\boldsymbol{a}_j\|^2)/2$.

Let I_1 and I_2 be two index sets such that $0 < k_{ij} \neq 1$ for all $\{i, j\} \in I_1$ and $k_{ij} = 1$ for all $\{i, j\} \in I_2$ with $1 \leq i \leq m-1, i+1 \leq j \leq m$ and $I_1 \cap I_2 = \emptyset$. Let L_1 and L_2 denote the number of elements in sets I_1 and I_2 respectively (number of hyperspheres and hyperplanes) and $L_1 + L_2 = m(m-1)/2$. Then the unknown location of the source can be found via minimization of the following criterion which is equivalent to (2.34) [55]

$$\underset{\boldsymbol{x}}{\text{minimize}} \sum_{l_1=1}^{L_1} \left(\|\boldsymbol{x} - \boldsymbol{c}_{l_1}\|^2 - \rho_{l_1}^2 \right)^2 + \sum_{l_2=1}^{L_2} \left(\boldsymbol{x}^T \boldsymbol{\gamma}_{l_2} - \tau_{l_2} \right)^2 \tag{2.38}$$

For the brevity of notation, the double indexes ij were replaced by single indexes l_1

and l_2. After some simple manupilations and necessary variable changes, (2.38) can be converted to the following constrained problem which is similar to (2.10)

$$\underset{\boldsymbol{y}\in R^{n+1}}{\text{minimize}}\ \|\boldsymbol{A}\boldsymbol{y}-\boldsymbol{b}\|^2 \tag{2.39a}$$

$$\text{subject to:}\quad \boldsymbol{y}^T\boldsymbol{D}\boldsymbol{y}+2\boldsymbol{f}^T\boldsymbol{y}=0 \tag{2.39b}$$

where

$$\boldsymbol{y}=\begin{pmatrix}\boldsymbol{x}\\ \|\boldsymbol{x}\|^2\end{pmatrix},\ \boldsymbol{A}=\begin{pmatrix}\boldsymbol{A}_1\\ \boldsymbol{A}_2\end{pmatrix},\ \boldsymbol{b}=\begin{pmatrix}\boldsymbol{b}_1\\ \boldsymbol{b}_2\end{pmatrix} \tag{2.40}$$

$$\boldsymbol{D}=\begin{pmatrix}\boldsymbol{I}_{n\times n} & \boldsymbol{0}_{n\times 1}\\ \boldsymbol{0}_{1\times n} & 0\end{pmatrix},\ \boldsymbol{f}=\begin{pmatrix}\boldsymbol{0}\\ -0.5\end{pmatrix}$$

and submatrices of $\boldsymbol{A}$ and elements of $\boldsymbol{b}$ are formed as follows:

$$\boldsymbol{A}_1=\begin{pmatrix}-2\boldsymbol{c}_1^T & 1\\ \vdots & \vdots\\ -2\boldsymbol{a}_{L_1}^T & 1\end{pmatrix},\ \boldsymbol{b}_1=\begin{pmatrix}\rho_1^2-\|\boldsymbol{c}_1\|^2\\ \vdots\\ \rho_{L_1}^2-\|\boldsymbol{c}_{L_1}\|^2\end{pmatrix} \tag{2.41}$$

$$\boldsymbol{A}_2=\begin{pmatrix}\gamma_1^T & 0\\ \vdots & \vdots\\ \gamma_{L_2}^T & 0\end{pmatrix},\ \boldsymbol{b}_2=\begin{pmatrix}\tau_1\\ \vdots\\ \tau_{L_2}\end{pmatrix}$$

The unconstrained problem in (2.38), due to mathematical analogy with (2.9), can be solved using Algorithm 1 developed in Section 3.1.

Chapter 3

Penalty Convex-Concave Procedure for Source Localization

In connection with the problem of localizing a single radiating source based on range measurements, in this chapter we explore special structure of the cost function of an unconstrained least squares (LS) formulation and show that it is well suited in a setting known as difference-of-convex-functions (DC) programming. In the literature, the DC programming is sometimes referred to as convex-concave procedure. Our focus in this chapter will be placed on the localization problem based on range measurements. We present an algorithm for solving the LS problem at hand based on a penalty convex-concave procedure (PCCP) [40] that accommodates infeasible initial points in solving a fairly large class of *nonconvex* constrained problems. Algorithmic details are provided to show that the PCCP-based formulation is tailored to the localization problem at hand. These include additional constraints that enforce the algorithms iteration path towards the LS solution, and several strategies to secure good initial points. Numerical results are presented to demonstrate that the proposed algorithm offers substantial performance improvement relative to some best known results from the literature.

3.1 Problem Statement and Review of Related Work

Typically non-survey based localization techniques compute the location estimates in two steps: range/angle estimation and tri-lateration/angulation [27]. In general, the range estimates can be based on different types of measurements, e.g. received signal strength (RSS), or time of arrival (TOA). This chapter will focus on the problem of range-based localization given the TOA information. In the TOA method, the one-way propagation time of the signal traveling between radiating source and the sensor node is measured. Each TOA measurement then provides a circle centered at the sensor node on which the source of the signal must lie. With three or more sensor nodes the measurements are converted into a set of circular equations that, with knowledge of the geometry of the sensor network, allow to determine the unknown source position [27]. The accuracy of the positioning depends on the quality of the range measurements, network geometry, and the performace of the localization algorithm. In real-world situations, multipath and non line-of-sight (NLOS) propagation are two major sources of error, which can introduce large biases in the TOA measurements and result in unreliable position esitmation [11]. In fact, mitigation of the impairments due to multipath and/or NLOS is another key research topic in wireless location and recent works in this area have reported some promising results [20]. Ultra-wideband (UWB) technology has the potential to deliver very accurate range measurements, thus enabling accurate positioning [18, 19, 20]. As a result, we assume that the multipath and NLOS errors in the TOA measurements have been successfully mitigated.

The source localization problem discussed in this chapter involves a given array of m sensors placed in the $n = 2$ or 3 dimentional space with coordinates specified by $\{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_m, \boldsymbol{a}_i \in R^n\}$. Each sensor measures its distance to a radiating source $\boldsymbol{x} \in R^n$. Throughout it is assumed that only noisy copies of the distance data are available, hence the *range measurements* obey the model

$$r_i = \|\boldsymbol{x} - \boldsymbol{a}_i\| + \varepsilon_i, \quad i = 1, \ldots, m. \tag{3.1}$$

where ε_i denotes the unknown noise that has occurred when the ith sensor measures its distance to source $\boldsymbol{x}$. Let $\boldsymbol{r} = [r_1\ r_2 \ldots r_m]^T$ and $\boldsymbol{\varepsilon} = [\varepsilon_1\ \varepsilon_2 \ldots \varepsilon_m]^T$. The source localization problem can be stated as to estimate the exact source location $\boldsymbol{x}$ from noisy range measurements $\boldsymbol{r}$.

The nonlinear least squares (NLLS) estimate refers to the solution of the problem

$$\underset{\boldsymbol{x}}{\text{minimize}} \quad F(\boldsymbol{x}) = \sum_{i=1}^{m} (r_i - \|\boldsymbol{x} - \boldsymbol{a}_i\|)^2 \tag{3.2}$$

If ranging errors ε_i are i.i.d. random variables that follow Gaussian distribution with zero mean and covariance matrix proportional to the identity matrix, then the NLLS estimate becomes identical to the maximum likelihood estimate. The NLLS formulation is also geometrically meaningful and has been often used as a benchmark to compare new algorithms [4, 19].

In Sec.2.1 of the thesis, it is demonstrated that these problems are hard to solve globally. Many relaxation and approximation methods were developed that offer either lower computational complexity, robustness against positive bias in the distance estimates due to non line-of-site situations, or better performance compared to standard unconstrain optimization methods applied to the NLLS problem. Convex relaxation of a nonconvex problem in (3.2) to an SDP problem and solution methods for *squared* range LS problems are discussed in detail in [32] and Sec.2.1 and of the thesis. Another localization approach that has received considerable interest applies classical multidimensional scaling (MDS) algorithm or its variants to the problem at hand [11, 50, 14, 58].

Multidimensional scaling is a field of study concerning the search of points in a low dimensional space that represent the objects of interest and the pairwise distances between the points (objects) (as measure of dissimilarities) match a set of given values. As such MDS has been an attractive technique for analyzing experimental data in physical, biological, and behavioral science [11]. The classical MDS is a subset of MDS techniques where the relative coordinates of points are determined given only their pairwise Euclidean distances.

When applied to the localization problem at hand, classical MDS starts with constructing a multidimentional similarity matrix. Let $\boldsymbol{X}$ denote an m x n distance matrix

$$\boldsymbol{X} = \begin{pmatrix} (\boldsymbol{x} - \boldsymbol{a}_1)^T \\ (\boldsymbol{x} - \boldsymbol{a}_2)^T \\ \vdots \\ (\boldsymbol{x} - \boldsymbol{a}_m)^T \end{pmatrix}$$

The multidimentional similarity matrix is then defined by $\boldsymbol{D} = \boldsymbol{X}\boldsymbol{X}^T$ which can also be expressed in terms of pairwise distances between sensor nodes and error-free range measurements. Since the exact distances between the sourse $\boldsymbol{x}$ and sensors are not available, the noisy range measurements r_i are used to construct an approximation of $\boldsymbol{D}$ as

$$\hat{\boldsymbol{D}}=\frac{1}{2}\begin{pmatrix} 2r_1^2 & r_1^2+r_2^2-\|\boldsymbol{a}_1-\boldsymbol{a}_2\|^2 & \dots & r_1^2+r_m^2-\|\boldsymbol{a}_1-\boldsymbol{a}_m\|^2 \\ r_1^2+r_2^2-\|\boldsymbol{a}_1-\boldsymbol{a}_2\|^2 & 2r_2^2 & \dots & r_2^2+r_m^2-\|\boldsymbol{a}_2-\boldsymbol{a}_m\|^2 \\ \vdots & \vdots & \ddots & \vdots \\ r_1^2+r_m^2-\|\boldsymbol{a}_1-\boldsymbol{a}_m\|^2 & r_2^2+r_m^2-\|\boldsymbol{a}_1-\boldsymbol{a}_m\|^2 & \dots & 2r_m^2 \end{pmatrix}$$

Because matrix $\hat{\boldsymbol{D}}$ is symmetric, it admits the orthogonal eigen-decomposition

$$\hat{\boldsymbol{D}} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T$$

where $\boldsymbol{\Lambda} = \operatorname{diag}(\lambda_1, \lambda_2, \dots, \lambda_m)$ is the diagonal matrix of eigenvalues of $\hat{\boldsymbol{D}}$ with $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_m \geq 0$, and $\boldsymbol{U} = [\boldsymbol{u}_1 \quad \boldsymbol{u}_2 \dots \boldsymbol{u}_m]$ is an orthonormatl matrix whose columns are the corresponding eigenvectors. Since the rank of the ideal $\boldsymbol{D}$ is 2, an LS estimate of $\boldsymbol{X}$, denoted by $\boldsymbol{X}_r$, can be computed up to an arbitrary rotation as a solution to the following problem [11]

$$\boldsymbol{X}_r = \arg\min_{\tilde{\boldsymbol{X}}} \|\hat{\boldsymbol{D}} - \tilde{\boldsymbol{X}}\tilde{\boldsymbol{X}}^T\|_F^2 = \boldsymbol{U}_s\boldsymbol{\Lambda}_s^{(1/2)}$$

where $\tilde{\boldsymbol{X}}$ is the variable matrix for $\boldsymbol{X}$, $\|.\|_F$ represents the Frobenius norm, $\boldsymbol{U}_s = [\boldsymbol{u}_1 \quad \boldsymbol{u}_2]$ corresponds to the signal subspase, and $\boldsymbol{\Lambda}^{(1/2)} = \operatorname{diag}(\lambda_1^{(1/2)}, \lambda_2^{(1/2)})$. In practical situations of nonzero range errors, the relationship between $\boldsymbol{X}_r$ and $\boldsymbol{X}$ is then

$$\boldsymbol{X} \approx \boldsymbol{X}_r\boldsymbol{\Omega}$$

where $\boldsymbol{\Omega}$ is an unknown rotation matrix to be determined. The estimate of the unknown rotation matrix $\boldsymbol{\Omega}$ and source location $\boldsymbol{x}$ can be obtained by solving an overdetermined system of linear equations [11]. In the absence of noise the symmetric $\hat{\boldsymbol{D}}$ is identical to $\boldsymbol{D}$, is positive semi-definite, and has a rank of 2. In the practical situations of nonzero range errors, $\hat{\boldsymbol{D}}$ will have a full rank.

Other methods based on MDS include a generalized subspace approach by So and Chan [58], that performs position estimation based on the noise subspace. A subspase-based weighting Lagrangian multiplier estimator [50] reduces computational complexity by avoiding the process of eigendecomposition or inverse computation, but it requires some a priori knowledge about noise statistic to construct the weighting matrix. On the other hand, the distributed weighted-multidimentional scaling

(DW-MDS) [14] adds a penalty term to the standard MDS objective function which accounts for prior knowledge about node locations. Although these methods work well in general and can be efficient in terms of complexity, they are found to produce poor performance in certain sensor deployments [19].

In this chapter, we focus on the least squares formulation for the localization problem, where the l_2-norm of the residual errors is minimized in a setting known as difference-of-convex-functions programming. The problem at hand is then solved by applying a penalty convex-concave procedure (PCCP) in a succesive manner [33].

3.2 Fitting the Localization Problem into a CCP Framework

3.2.1 Basic Convex-Concave Procedure

The CCP refers to an effective heuristic method to deal with a class of difference of convex (DC) programming problems, which assume the form

$$\begin{aligned} \underset{\boldsymbol{x}}{\text{minimize}} \quad & f(\boldsymbol{x}) - g(\boldsymbol{x}) && \text{(3.3a)} \\ \text{subject to:} \quad & f_i(\boldsymbol{x}) \leq g_i(\boldsymbol{x}) \quad \text{for: } i = 1, 2, \ldots, m && \text{(3.3b)} \end{aligned}$$

where $f(\boldsymbol{x}), g(\boldsymbol{x}), f_i(\boldsymbol{x}), g_i(\boldsymbol{x})$ for $i = 1, 2, \ldots, m$ are convex. A DC program is not convex unless the functions g and g_i are affine, and therefore is hard to solve in general. The class of DC functions is very broad. For example, any C^2 function can be expressed as a difference of convex functions [30]. Classes of problems that can be expressed as difference of convex programming among others include Boolean linear program, circle packing, circuit layout, and multi-matrix proncipal component analysis, examples of which can be found in [40] and references therein.

The basic CCP algorithm is an iterative procedure including two key steps (in the k-th iteration where iterate $\boldsymbol{x}_k$ is known):

(i) Convexification of the objective function and constraints by replacing $g(\boldsymbol{x})$ and $g_i(\boldsymbol{x})$, respectively, with their affine approximations

$$\hat{g}(\boldsymbol{x}, \boldsymbol{x}_k) = g(\boldsymbol{x}_k) + \triangledown g(\boldsymbol{x}_k)^T (\boldsymbol{x} - \boldsymbol{x}_k) \tag{3.4a}$$

and

$$\hat{g}_i(\boldsymbol{x}, \boldsymbol{x}_k) = g_i(\boldsymbol{x}_k) + \bigtriangledown g_i(\boldsymbol{x}_k)^T(\boldsymbol{x} - \boldsymbol{x}_k) \\ \text{for: } i = 1, 2, \ldots, m \tag{3.4b}$$

(ii) Solving the convex problem

$$\underset{\boldsymbol{x}}{\text{minimize}} \quad f(\boldsymbol{x}) - \hat{g}(\boldsymbol{x}, \boldsymbol{x}_k) \tag{3.5a}$$

$$\text{subject to:} \quad f_i(\boldsymbol{x}) - \hat{g}_i(\boldsymbol{x}, \boldsymbol{x}_k) \leq 0 \tag{3.5b}$$

$$\text{for: } i = 1, 2, \ldots, m$$

Because of the convexity of all the functions involved, it can be shown that the basic CCP is a descent algorithm and the iterates $\boldsymbol{x}_k$ converge to the critical point of the original problem [40]. In fact, the global convergence analysis for CCP has also been studied [36, 34]. Note that functions $g(\boldsymbol{x})$ and $g_i(\boldsymbol{x})$ for $i = 1, 2, \ldots, m$ are required to be convex but not necessarily differentiable. If any of $g(\boldsymbol{x})$ or $g_i(\boldsymbol{x})$ are not differentiable at some point $\tilde{\boldsymbol{x}}$ then the corresponding term $\bigtriangledown g(\tilde{\boldsymbol{x}})$ (or $\bigtriangledown g_i(\tilde{\boldsymbol{x}})$) is replaced by a subgradient of $g(\boldsymbol{x})$ (or $g_i(\boldsymbol{x})$) at point $\tilde{\boldsymbol{x}}$.

Let $\mathcal{D}$ be a nonempty set in R^n. A vector $\boldsymbol{h} \in R^n$ is said to be a subgradient of a convex function $f : \mathcal{D} \to R$ at $\boldsymbol{x} \in \mathcal{D}$ if

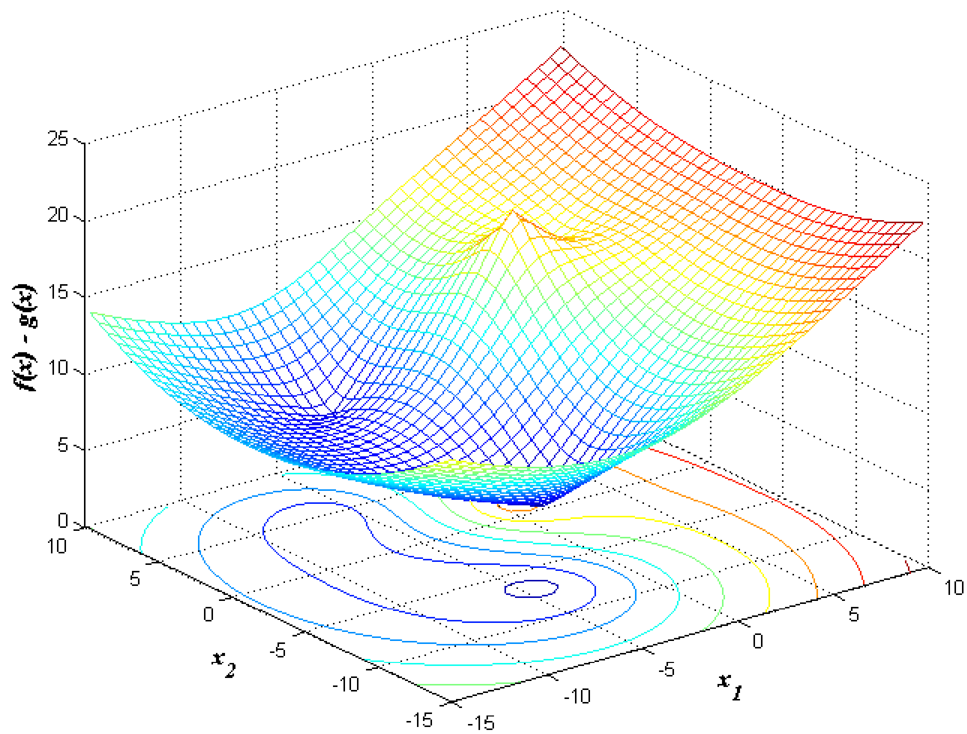
$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \boldsymbol{h}^T(\boldsymbol{y} - \boldsymbol{x}) \text{ for all } \boldsymbol{y} \in \mathcal{D}$$

Geometrically, the subgradients at a point $\boldsymbol{x}$ for the case where the convex function $f(x)$ is not differentiable correspond to different tangent lines at $\boldsymbol{x}$ [1].

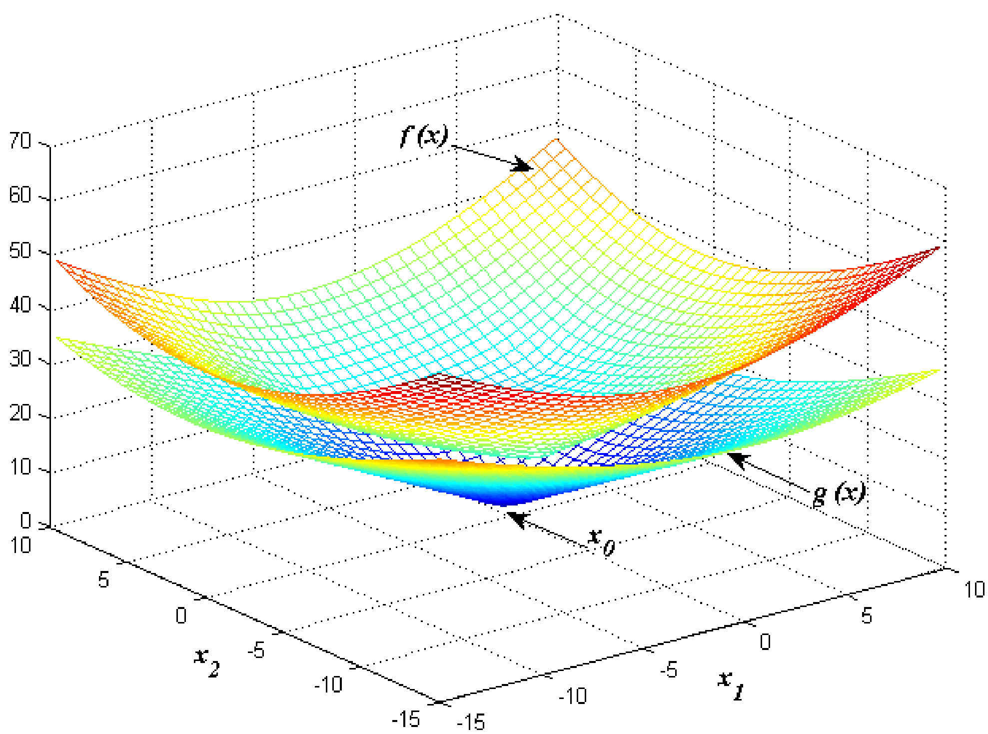
Figure 3.1 shows an example of the CCP approach for an unconstrained DC problem in the form of (3.3a), where $f(\boldsymbol{x})$ and $g(\boldsymbol{x})$ are given by:

$$\begin{aligned} f(\boldsymbol{x}) &= \|\boldsymbol{x} - \boldsymbol{a}_1\| + \|\boldsymbol{x} - \boldsymbol{a}_2\| + \|\boldsymbol{x} - \boldsymbol{a}_3\| \\ g(\boldsymbol{x}) &= \|2\boldsymbol{x} - \boldsymbol{a}_4\| \end{aligned}$$

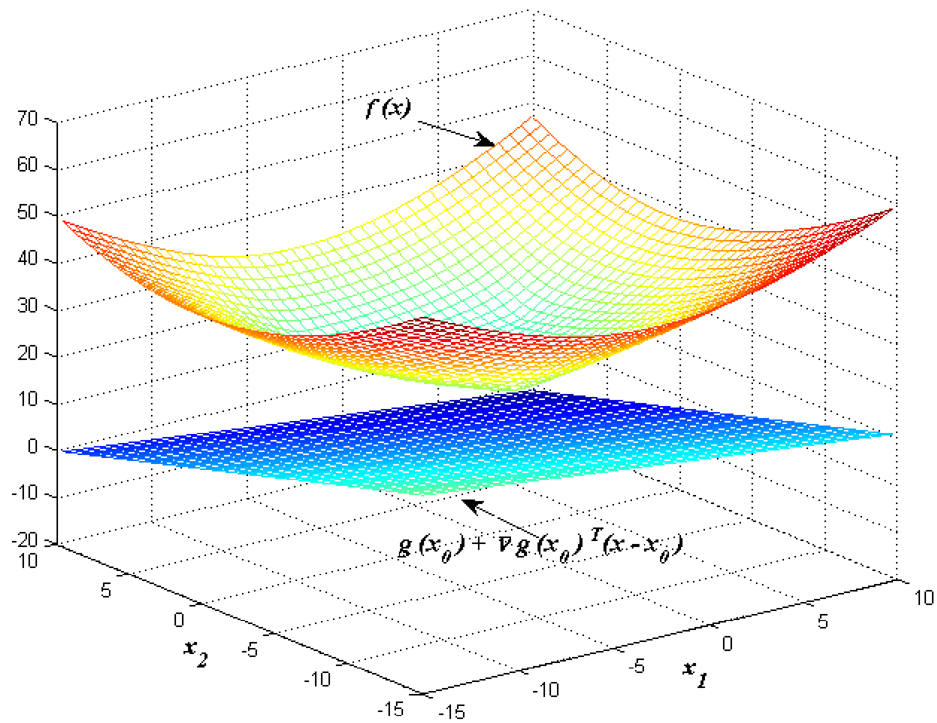
with $\boldsymbol{a}_1 = [3\ 2]^T$, $\boldsymbol{a}_2 = [6\ 5]^T$, $\boldsymbol{a}_3 = [4\ 7]^T$, and $\boldsymbol{a}_4 = [1\ 2]^T$. In this figure, the original nonconvex problem is transferred to a convex problem by replacing a nonconvex part $(-g(\boldsymbol{x}))$ by its affine approximation around the point $\boldsymbol{x}_0 = [0\ 0]^T$.



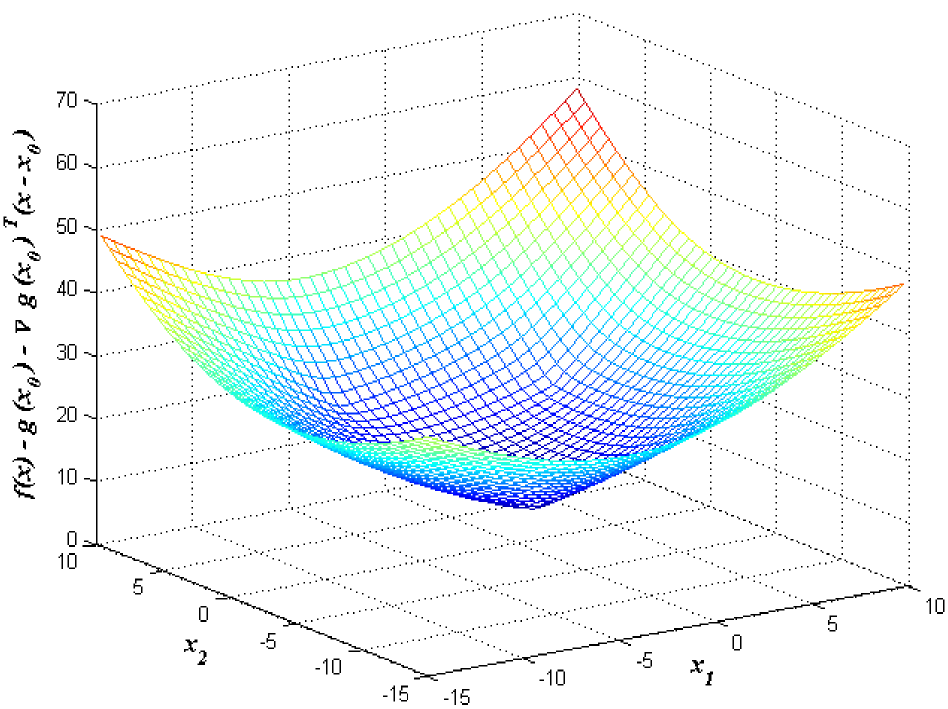
(a) A nonconvex function in the form of the difference of two convex functions and its contour plot.



(b) Separation of the nonconvex function into two convex functions $f(\boldsymbol{x})$ and $g(\boldsymbol{x})$.



(c) First order approximation of $g(\boldsymbol{x})$.



(d) A convex approximation of the original nonconvex function at $\boldsymbol{x}_0 = [0\ 0]^T$.

Figure 3.1: An example of the CCP procedure (re-generated based on [17]).

The basic CCP requires a *feasible* initial point $\boldsymbol{x}_0$ (in the sense that $\boldsymbol{x}_0$ satisfies (3.5b) for $i = 1, 2, \ldots, m$) to start the procedure. By introducing additional slack variables, a penalty CCP has been adopted to accept infeasible initial points. In what follows, we reformulate our localization problem to fit it into the basic CCP framework. Bounds on squared measurement errors as well as penalty terms are then imposed, and a PCCP-based algorithm is developed for solving the problem.

3.2.2 Problem Reformulation

We begin by re-writing the NLLS objective function in (3.2) up to a constant as

$$F(\boldsymbol{x}) = m\boldsymbol{x}^T\boldsymbol{x} - 2\boldsymbol{x}^T\sum_{i=1}^{m}\boldsymbol{a}_i - 2\sum_{i=1}^{m} r_i\|\boldsymbol{x} - \boldsymbol{a}_i\| \tag{3.6}$$

The objective in (3.6) is not convex. This is because, for points $\boldsymbol{x}$ that are not coincided with $\boldsymbol{a}_i$ for $1 \le i \le m$, the Hessian of $F(\boldsymbol{x})$ is given by

$$\bigtriangledown^2 F(\boldsymbol{x}) = 2m\boldsymbol{I} + 2\sum_{i=1}^{m}\frac{r_i}{\|\boldsymbol{x} - \boldsymbol{a}_i\|^3}\cdot\Big((\boldsymbol{x} - \boldsymbol{a}_i)(\boldsymbol{x} - \boldsymbol{a}_i)^T - \|\boldsymbol{x} - \boldsymbol{a}_i\|^2\boldsymbol{I}\Big)$$

which is obviously not always positive semidefinite. On the other hand, by defining

$$\begin{aligned} f(\boldsymbol{x}) &= m\boldsymbol{x}^T\boldsymbol{x} - 2\boldsymbol{x}^T\sum_{i=1}^{m}\boldsymbol{a}_i \\ g(\boldsymbol{x}) &= 2\sum_{i=1}^{m} r_i\|\boldsymbol{x} - \boldsymbol{a}_i\| \end{aligned} \tag{3.7}$$

the objective in (3.6) can be expressed as

$$F(\boldsymbol{x}) = f(\boldsymbol{x}) - g(\boldsymbol{x})$$

with both $f(\boldsymbol{x})$ and $g(\boldsymbol{x})$ convex, hence it fits naturally into (3.3). Note that $g(\boldsymbol{x})$ in (3.7) is not differentiable at the point where $\boldsymbol{x} = \boldsymbol{a}_i$ for some $1 \le i \le m$, thus we replace the term $\bigtriangledown g(\boldsymbol{x}_k)$ in (3.4b) by a subgradient [46] of $g(\boldsymbol{x})$ at $\boldsymbol{x}_k$, denoted by $\partial g(\boldsymbol{x}_k)$, as

$$\partial g(\boldsymbol{x}_k) = 2\sum_{i=1}^{m} r_i\partial\|\boldsymbol{x}_k - \boldsymbol{a}_i\|$$

where

$$\partial\|\boldsymbol{x}_k - \boldsymbol{a}_i\| = \begin{cases} \dfrac{\boldsymbol{x}_k - \boldsymbol{a}_i}{\|\boldsymbol{x}_k - \boldsymbol{a}_i\|}, & \text{if } \boldsymbol{x}_k \neq \boldsymbol{a}_i \\ \boldsymbol{0}, & \text{otherwise} \end{cases}$$

Hence $\hat{g}(\boldsymbol{x}, \boldsymbol{x}_k)$ in (3.4b) is given by

$$\begin{aligned} \hat{g}(\boldsymbol{x}, \boldsymbol{x}_k) &= 2\sum_{i=1}^{m} r_i \|\boldsymbol{x}_k - \boldsymbol{a}_i\| + 2\left(\boldsymbol{x} - \boldsymbol{x}_k\right)^T \sum_{i=1}^{m} r_i \partial\|\boldsymbol{x}_k - \boldsymbol{a}_i\| \\ &= 2\boldsymbol{x}^T \sum_{i=1}^{m} r_i \partial\|\boldsymbol{x}_k - \boldsymbol{a}_i\| + c \end{aligned}$$

where c is a constant given by

$$\begin{aligned} c &= 2\sum_{i=1}^{m} r_i \|\boldsymbol{x}_k - \boldsymbol{a}_i\| - 2\boldsymbol{x}_k^T \sum_{i=1}^{m} r_i \partial\|\boldsymbol{x}_k - \boldsymbol{a}_i\| \\ &= 2\sum_{i=1}^{m} r_i \|\boldsymbol{x}_k - \boldsymbol{a}_i\| - 2\sum_{i=1}^{m} r_i \boldsymbol{x}_k^T \partial\|\boldsymbol{x}_k - \boldsymbol{a}_i\| \\ &= 2\sum_{i=1}^{m} r_i \|\boldsymbol{x}_k - \boldsymbol{a}_i\| - 2\sum_{i=1}^{m} r_i \left(\boldsymbol{x}_k^T - \boldsymbol{a}_i + \boldsymbol{a}_i\right)^T \partial\|\boldsymbol{x}_k - \boldsymbol{a}_i\| \\ &= 2\sum_{i=1}^{m} r_i \|\boldsymbol{x}_k - \boldsymbol{a}_i\| - 2\sum_{i=1}^{m} r_i \left(\boldsymbol{x}_k^T - \boldsymbol{a}_i\right)^T \partial\|\boldsymbol{x}_k - \boldsymbol{a}_i\| - 2\sum_{i=1}^{m} r_i \boldsymbol{a}_i^T \partial\|\boldsymbol{x}_k - \boldsymbol{a}_i\| \\ &= 2\sum_{i=1}^{m} r_i \|\boldsymbol{x}_k - \boldsymbol{a}_i\| - 2\sum_{i=1}^{m} r_i \|\boldsymbol{x}_k - \boldsymbol{a}_i\| - 2\sum_{i=1}^{m} r_i \boldsymbol{a}_i^T \partial\|\boldsymbol{x}_k - \boldsymbol{a}_i\| \\ &= -2\sum_{i=1}^{m} r_i \boldsymbol{a}_i^T \partial\|\boldsymbol{x}_k - \boldsymbol{a}_i\|. \end{aligned}$$

The convex approximation of the objective in (3.6) can now be derived as

$$\begin{aligned} \hat{F}(\boldsymbol{x}) &= f(\boldsymbol{x}) - \hat{g}(\boldsymbol{x}, \boldsymbol{x}_k) \\ &= m\boldsymbol{x}^T\boldsymbol{x} - 2\boldsymbol{x}^T \sum_{i=1}^{m} \boldsymbol{a}_i - 2\boldsymbol{x}^T \sum_{i=1}^{m} r_i \partial\|\boldsymbol{x}_k - \boldsymbol{a}_i\| + c \end{aligned}$$

It follows that, up to a multiplicative factor $1/m$ and an additive constant term, the convex objective function in (3.5b) can be written as

$$\underset{\boldsymbol{x}}{\text{minimize}} \quad \hat{F}(\boldsymbol{x}) = \boldsymbol{x}^T\boldsymbol{x} - 2\boldsymbol{x}^T\boldsymbol{v}_k \tag{3.8}$$

where

$$\boldsymbol{v}_k = \bar{\boldsymbol{a}} + \frac{1}{m}\sum_{i=1}^{m} r_i \partial \|\boldsymbol{x}_k - \boldsymbol{a}_i\|, \quad \bar{\boldsymbol{a}} = \frac{1}{m}\sum_{i=1}^{m} \boldsymbol{a}_i \tag{3.9}$$

It is rather straightforward to see that given $\boldsymbol{x}_k$ (in the k-th iteration) the solution of the quadratic problem (3.8) can be obtained as

$$\boldsymbol{x}_{k+1} = \bar{\boldsymbol{a}} + \frac{1}{m}\sum_{i=1}^{m} r_i \partial \|\boldsymbol{x}_k - \boldsymbol{a}_i\|. \tag{3.10}$$

3.2.3 Imposing Error Bounds and Penalty Terms

The algorithm being developed can be enhanced by imposing a bound on each squared measurement error, namely

$$(\|\boldsymbol{x} - \boldsymbol{a}_i\| - r_i)^2 \leq \delta_i^2 \tag{3.11}$$

which leads to

$$\|\boldsymbol{x} - \boldsymbol{a}_i\| - r_i - \delta_i \leq 0 \tag{3.12a}$$

$$r_i - \delta_i \leq \|\boldsymbol{x} - \boldsymbol{a}_i\| \tag{3.12b}$$

for $1 \leq i \leq m$. Placing such bounds means that as iterations proceed the new iterates (coordinates of possible source locations) are restricted to lie within a physically meaningfull region determined by parameters δ_i.

Note that the constraints in (3.12a) are convex and fit into the form of basic CCP in (3.5b) with $f_i(\boldsymbol{x}) = \|\boldsymbol{x} - \boldsymbol{a}_i\| - r_i - \delta_i$ and $g_i(\boldsymbol{x}) = 0$, while those in (3.12b) are in the form of (3.3) with $f_i(\boldsymbol{x}) = r_i - \delta_i$ and $g_i(\boldsymbol{x}) = \|\boldsymbol{x} - \boldsymbol{a}_i\|$. Following CCP (see (3.4b)), $g_i(\boldsymbol{x}) = \|\boldsymbol{x} - \boldsymbol{a}_i\|$ is linearized around iterate $\boldsymbol{x}_k$ to

$$\hat{g}_i(\boldsymbol{x}, \boldsymbol{x}_k) = \|\boldsymbol{x}_k - \boldsymbol{a}_i\| + \partial \|\boldsymbol{x}_k - \boldsymbol{a}_i\|^T (\boldsymbol{x} - \boldsymbol{x}_k)$$

and (3.12b) is convexified as

$$r_i - \delta_i \leq \|\boldsymbol{x}_k - \boldsymbol{a}_i\| + \partial \|\boldsymbol{x}_k - \boldsymbol{a}_i\|^T (\boldsymbol{x} - \boldsymbol{x}_k)$$

which now fits into (3.5b), or equivalently

$$-\|\boldsymbol{x}_k - \boldsymbol{a}_i\| - \partial\|\boldsymbol{x}_k - \boldsymbol{a}_i\|^T(\boldsymbol{x} - \boldsymbol{x}_k) + r_i - \delta_i \leq 0 \tag{3.13}$$

We remark that constraint (3.13) is not only convex but also tighter than (3.12b). As a matter of fact, the convexity of the norm $\|\boldsymbol{x} - \boldsymbol{a}_i\|$ implies that it obeys the property

$$\|\boldsymbol{x} - \boldsymbol{a}_i\| \geq \|\boldsymbol{x}_k - \boldsymbol{a}_i\| + \partial\|\boldsymbol{x}_k - \boldsymbol{a}_i\|^T(\boldsymbol{x} - \boldsymbol{x}_k)$$

Therefore, a point $\boldsymbol{x}$ satisfying (3.13) automatically satisfies (3.12b). Summarizing, the convexified problem in the k-th iteration can be stated as

$$\begin{aligned} &\underset{\boldsymbol{x}}{\text{minimize}} && \boldsymbol{x}^T\boldsymbol{x} - 2\boldsymbol{x}^T\boldsymbol{v}_k && \text{(3.14a)} \\ &\text{subject to:} && \|\boldsymbol{x} - \boldsymbol{a}_i\| - r_i - \delta_i \leq 0 && \text{(3.14b)} \end{aligned}$$

$$-\|\boldsymbol{x}_k - \boldsymbol{a}_i\| - \partial\|\boldsymbol{x}_k - \boldsymbol{a}_i\|^T(\boldsymbol{x} - \boldsymbol{x}_k) + r_i - \delta_i \leq 0 \tag{3.14c}$$

A technical problem making the formulation in (3.14) difficult to implement is that it requires a feasible initial point $\boldsymbol{x}_0$. The problem can be overcome by introducing nonnegative slack variables $s_i \geq 0, \hat{s}_i \geq 0$, for $i = 1, \ldots, m$ into the constraints in (3.14b) and (3.14c) to replace their right-hand sides (which are zeros) by relaxed upper bounds (as these new bounds themselves are nonnegative variables). This leads to a *penalty* CCP (PCCP) based formulation as follows:

$$\begin{aligned} &\underset{\boldsymbol{x},\boldsymbol{s},\hat{\boldsymbol{s}}}{\text{minimize}} && \boldsymbol{x}^T\boldsymbol{x} - 2\boldsymbol{x}^T\boldsymbol{v}_k + \tau_k\sum_{i=1}^{m}(s_i + \hat{s}_i) && \text{(3.15a)} \\ &\text{subject to:} && \|\boldsymbol{x} - \boldsymbol{a}_i\| - r_i - \delta_i \leq s_i && \text{(3.15b)} \end{aligned}$$

$$\begin{aligned} -\|\boldsymbol{x}_k - \boldsymbol{a}_i\| - \frac{(\boldsymbol{x}_k - \boldsymbol{a}_i)^T}{\|\boldsymbol{x}_k - \boldsymbol{a}_i\|}(\boldsymbol{x} - \boldsymbol{x}_k) + r_i - \delta_i \leq \hat{s}_i && \text{(3.15c)} \\ s_i \geq 0, \hat{s}_i \geq 0, \text{ for: } i = 1, 2, \ldots, m && \text{(3.15d)} \end{aligned}$$

where the weight $\tau_k \geq 0$ increases as iterations proceed until it reaches an upper limit τ_{max}. By using a monotonically increasing τ_k for the penalty term in (3.15a), the algorithm reduces the slack variables s_i and $\hat{s}_i$ very quickly. As a result, new iterates quickly become feasible as s_i and $\hat{s}_i$ vanish. The upper limit τ_{max} is imposed to avoid numerical difficulties that may occur if τ_k becomes too large and to ensure convergence if a feasible region is not found [9]. Consequently, while formulation (3.15) accepts *infeasible* initial points, the iterates obtained by solving (16) are practically identical to those obtained by solving (3.14).

3.2.4 The Algorithm

The input parameters for the algorithm include the bounds δ_i on the measurement error. Setting δ_i to a lower value leads to a "tighter" solution. On the other hand, a larger δ_i would make the algorithm less sensitive to outliers. However, some a priori knowledge about noise statistic, if available, or sensor geometry can be used to derive reasonable values for δ_i. For example, if measurement noise $\boldsymbol{\varepsilon}$ obeys a Gaussian distribution with zero mean and known covariance $\boldsymbol{\Sigma} = \text{diag}(\sigma_1^2, \ldots, \sigma_m^2)$, then δ_i can be expressed as $\delta_i = \gamma\sigma_i$, where γ is a parameter that determines the width of the confidence interval. For example, for $\gamma = 3$ we have the probability $Pr\{|\varepsilon_i| \leq 3\sigma_i\} \approx 0.99$. Other input parameters are initial point $\boldsymbol{x}_0$, maximum number of iterations K_{max}, initial weight τ_0, and upper limit of weight τ_{max} (to avoid numerical problems that may occur if τ_i becomes too large).

As mentioned in Sec. 3.2 of the thesis, the original LS objective is highly nonconvex with many local minimums even for small-scale systems. Consequently, it is of critical importance to select a good initial point for the proposed PCCP-based algorithm because PCCP is essentially a local procedure. Several techniques are available, these include:

(i) Select the initial point uniformly randomly over the same region as the unknown radiating source;

(ii) Set the initial point to the origin;

(iii) Run the algorithm from a set of candidate initial points and identify the solution as the one with lowest LS error. Typically, comparing the results from n distinct initial points shall suffice. For the planar case ($n = 2$), for example, it is sufficient to compare the two intersection points of the two circles that are associated with the two smallest distance readings as the target is very likely to be in the vicinity of these sensors;

(iv) Apply a global localization algorithm such as those in [4] to generate an approximate LS solution, then take it as the initial point to run the proposed algorithm.

The algorithm can be now outlined as follows.

Algorithm 3. PCCP-based LS Algorithm for Source Localization

1) Input data: Sensor locations $\{\boldsymbol{a}_i, i = 1, \ldots, m\}$, range measurements $\{r_i, i = 1, \ldots, m\}$, initial point $\boldsymbol{x}_0$, maximum number of iterations K_{max}, initial weight τ_0 and upper limit of weight τ_{max}, weight increment $\mu > 0$, error bounds δ_i. Set iteration count to $k = 0$.

2) Form $\boldsymbol{v}_k$ as

$$\boldsymbol{v}_k = \bar{\boldsymbol{a}} + \frac{1}{m}\sum_{i=1}^{m} r_i \partial \|\boldsymbol{x}_k - \boldsymbol{a}_i\|, \quad \bar{\boldsymbol{a}} = \frac{1}{m}\sum_{i=1}^{m} \boldsymbol{a}_i$$

and solve

$$\begin{aligned} \underset{\boldsymbol{x},\boldsymbol{s},\hat{\boldsymbol{s}}}{\text{minimize}} \quad & \boldsymbol{x}^T\boldsymbol{x} - 2\boldsymbol{x}^T\boldsymbol{v}_k + \tau_k \sum_{i=1}^{m}(s_i + \hat{s}_i) \\ \text{subject to:} \quad & \|\boldsymbol{x} - \boldsymbol{a}_i\| - r_i - \delta_i \leq s_i \end{aligned}$$

$$-\|\boldsymbol{x}_k - \boldsymbol{a}_i\| - \frac{(\boldsymbol{x}_k - \boldsymbol{a}_i)^T}{\|\boldsymbol{x}_k - \boldsymbol{a}_i\|}(\boldsymbol{x} - \boldsymbol{x}_k) + r_i - \delta_i \leq \hat{s}_i$$
$$s_i \geq 0, \hat{s}_i \geq 0, \text{ for: } i = 1, 2, \ldots, m$$

Denote the solution as $(\boldsymbol{s}^*, \hat{\boldsymbol{s}}^*, \boldsymbol{x}^*)$.

3) Update $\tau_{k+1} = \min\,(\mu\tau_k, \tau_{max})$, set $k = k + 1$.

4) If $k = K_{max}$, terminate and output $\boldsymbol{x}^*$ as the solution; otherwise, set $\boldsymbol{x}_k = \boldsymbol{x}^*$ and repeat from Step 2.

3.3 Numerical Results

For illustration purposes, the proposed algorithm was applied to a network with five sensors, and its performance was evaluated and compared with existing state-of-the-art methods by Monte Carlo simulations with a set-up similar to that of [4]. SR-LS solutions were used as performance benchmarks for the PCCP-based LS Algorithm. The system consisted of 5 sensors $\{\boldsymbol{a}_i, i = 1, 2, \ldots, 5\}$ randomly placed in the planar region in $[-15; 15] \times [-15; 15]$, and a radiating source $\boldsymbol{x}_s$, located randomly in the region $\{\boldsymbol{x} = [x_1; x_2], -10 \leq x_1, x_2 \leq 10\}$. The coordinates of the source and sensors were generated for each dimension following a uniform distribution. Measurement noise $\{\varepsilon_i, i = 1, \ldots, m\}$ was modelled as independent and identically distributed (i.i.d) random variables with zero mean and variance σ^2, with σ being one of four possible levels $\{10^{-3}, 10^{-2}, 10^{-1}, 1\}$. The range measurements $\{r_i, i = 1, 2, \ldots, 5\}$ were calculated using (3.1). Accuracy of source location estimation was evaluated in terms of average of the squared position error error in the form $\|\boldsymbol{x}^* - \boldsymbol{x}_s\|^2$, where $\boldsymbol{x}_s$ denotes the exact source location and $\boldsymbol{x}^*$ is its estimation obtained by SR-LS and PCCP methods, respectively. In our simulations parameter γ was set to 3 and the number of iterations was set to 20. The proposed method was implemented by using CVX [16] and implementation of SR-LS followed [4]. The PCCP algorithm was initialized with intersection points of the two circles that are associated with the two smallest distance readings. A candidate solution point with lowest LS error in (3.2) was chosen as a PCCP solution. In cases when the circles did not intersect due to high noise level, the initial point was set as a midpoint between the centers of the two circles.

Table 3.1 provides comparisons of the PCCP with SR-LS and MLE, where each entry is averaged squared error over 1,000 Monte Carlo runs of the method. The MLE was implemented using Matlab function *lsqnonlin* [43], initialized with the same point as PCCP. It is observed that, comparing with SR-LS, the estimates produced by the proposed algorithm are found to be closer to the true source locations in MSE sense. The last column of the table represents relative improvement of the proposed method over SR-LS solutions in percentage.

Figure 3.2 illustrates the location estimation errors for the SR-LS solution and the proposed algorithm for various numbers of sensors. Each entry is a squared error of the given method averaged over 50 random initializations of the system consisting of m sensors, with m being one of $\{4, 5, 7, 10\}$. The MLE was implemented using Matlab

Table 3.1: Averaged MSE for SR-LS and PCCP methods

σ	MLE	SR - LS	PCCP	R.I.
1e-03	6.0159e-01	1.3394e-06	**9.5243e-07**	29%
1e-02	3.5077e-01	1.4516e-04	**9.5831e-05**	34%
1e-01	3.7866e-01	1.2058e-02	**8.7107e-03**	28%
1e+0	1.4470e+00	1.3662e+00	**1.2346e+00**	10%

function *lsqnonlin*, initialized with the same point as PCCP-based LS. It is observed that at all noise levels the proposed approach performes well compared to SR-LS, especially for a low number of reference nodes. From the figure, it is noted that for the high standard deviation of the noise, as the number of sensor nodes increases, the PCCP-based LS and SR-LS show similar performance.

Finally, we study the convergence of the PCCP-based LS Algorithm. As an example, consider an instance of the source localization problem on the plane ($n = 2$) as discussed in Sec. 2.1 of the thesis. The system consists of five sensors ($m = 5$) located at $(6, 4)^T, (0, -10)^T, (5, -3)^T, (1, -4)^T$ and $(3, -3)^T$ with the source emmitting the signal at $\boldsymbol{x}_s = (-2, 3)^T$. Figure 3.3 shows the iteration path the PCCP-based LS follows when initialized with a randomly selected point $\boldsymbol{x}_0 = (-20, -10)^T$. It took 8 iterations for the PCCP-based LS to converge to a point $\boldsymbol{x}^* = (-1.9914, 3.1632)^T$ which is a good estimate of the R-LS approximation of the exact source location $\boldsymbol{x}_s$.

Figure 3.4 depicts the convergence speed of the proposed algorithm for 50 random initializations of the system with a fixed number of sensors (m). In the simulations, we set $\sigma = 10^{-1}$ and consider systems with 4, 5, 7 and 10 sensors randomly placed in the planar region in $[-15; 15] \times [-15; 15]$. For every estimate of the PCCP-based LS, we compute the value of the objective in (3.6). It is observed that PCCP-based LS converges fairly fast, in most cases after 10 iterations. For systems with large numbers of sensor nodes ($m = 10$) PCCP-based LS converges approximately in three sequential updates.

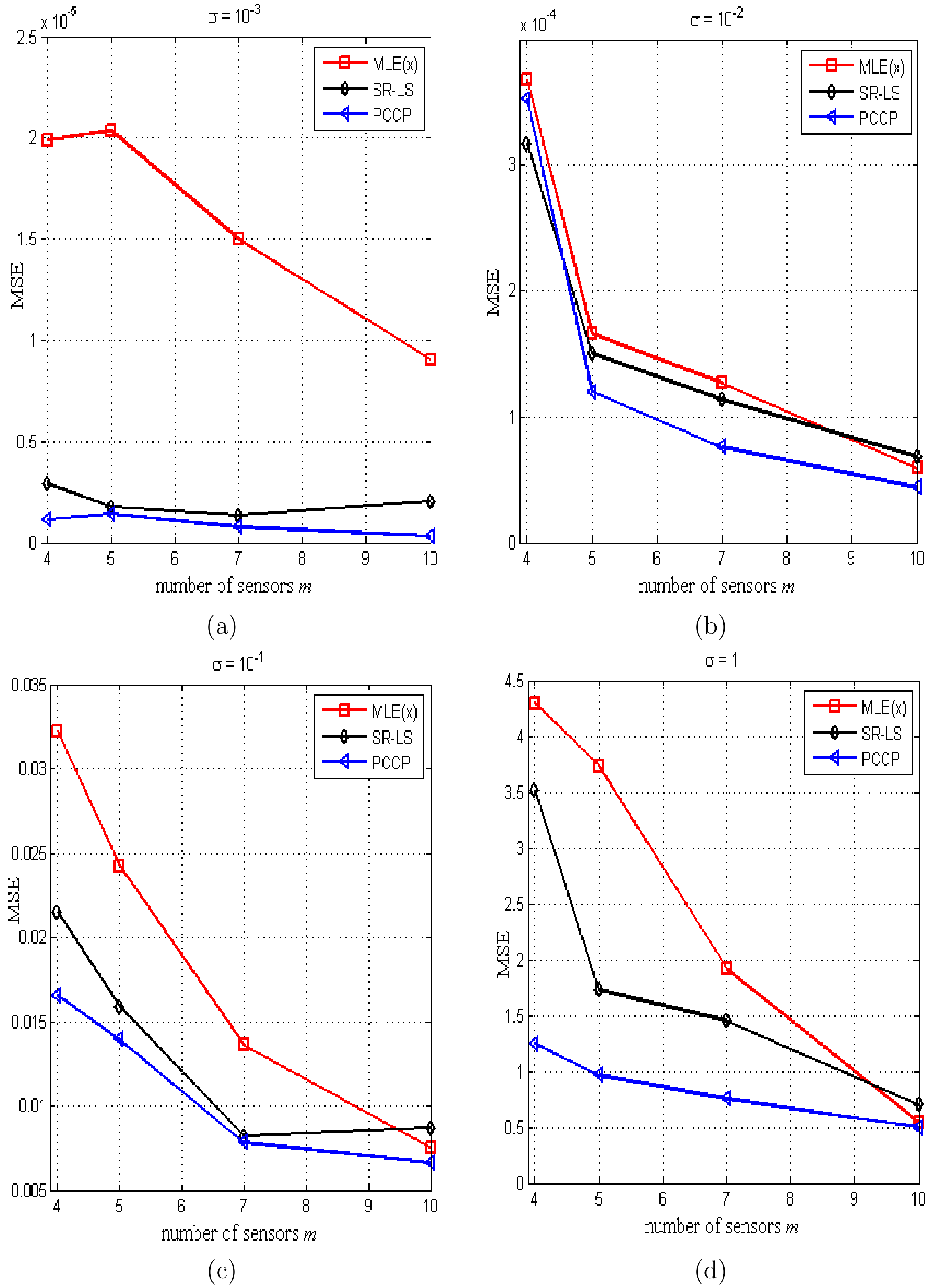


Figure 3.2: MSE for different methods, various number of sensor nodes m and different noise levels with (a) $\sigma = 10^{-3}$, (b) $\sigma = 10^{-2}$, (c) $\sigma = 10^{-1}$, and (d) $\sigma = 1$.

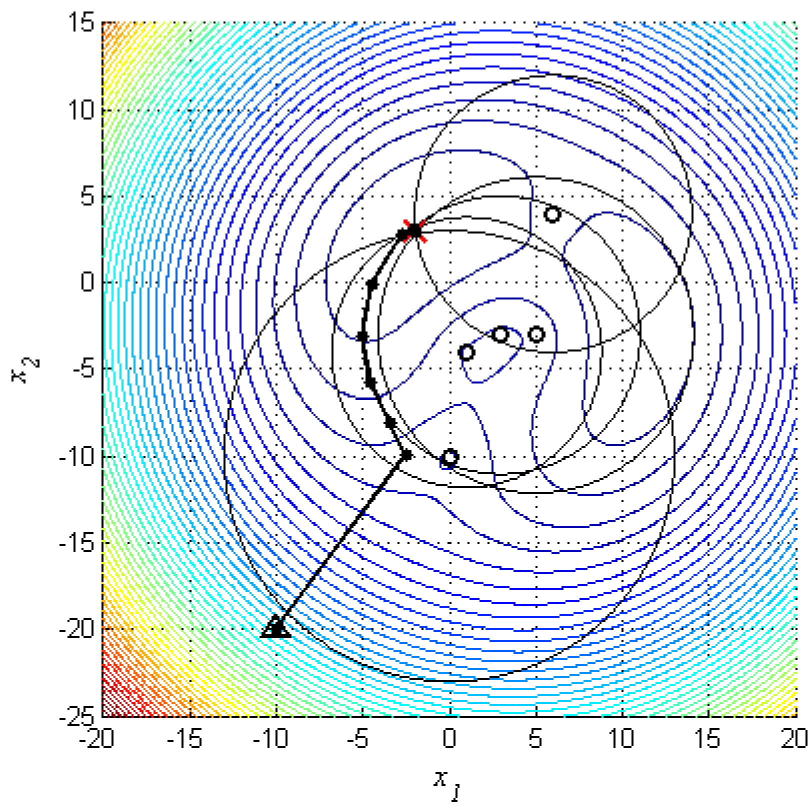


Figure 3.3: Iteration path of the PCCP-based LS Algorithm and contours of the R-LS objective function over the region $\Re = \{\boldsymbol{x} : -15 \leq x_1 \leq 15, -25 \leq x_2 \leq 15\}$. The red cross indicates the location of the signal source. Sensors are located at $(6, 4)^T$, $(0, -10)^T$, $(5, -3)^T$, $(1, -4)^T$ and $(3, -3)^T$. Large circles denote possible source locations given the noisy range reading at a particular sensor.

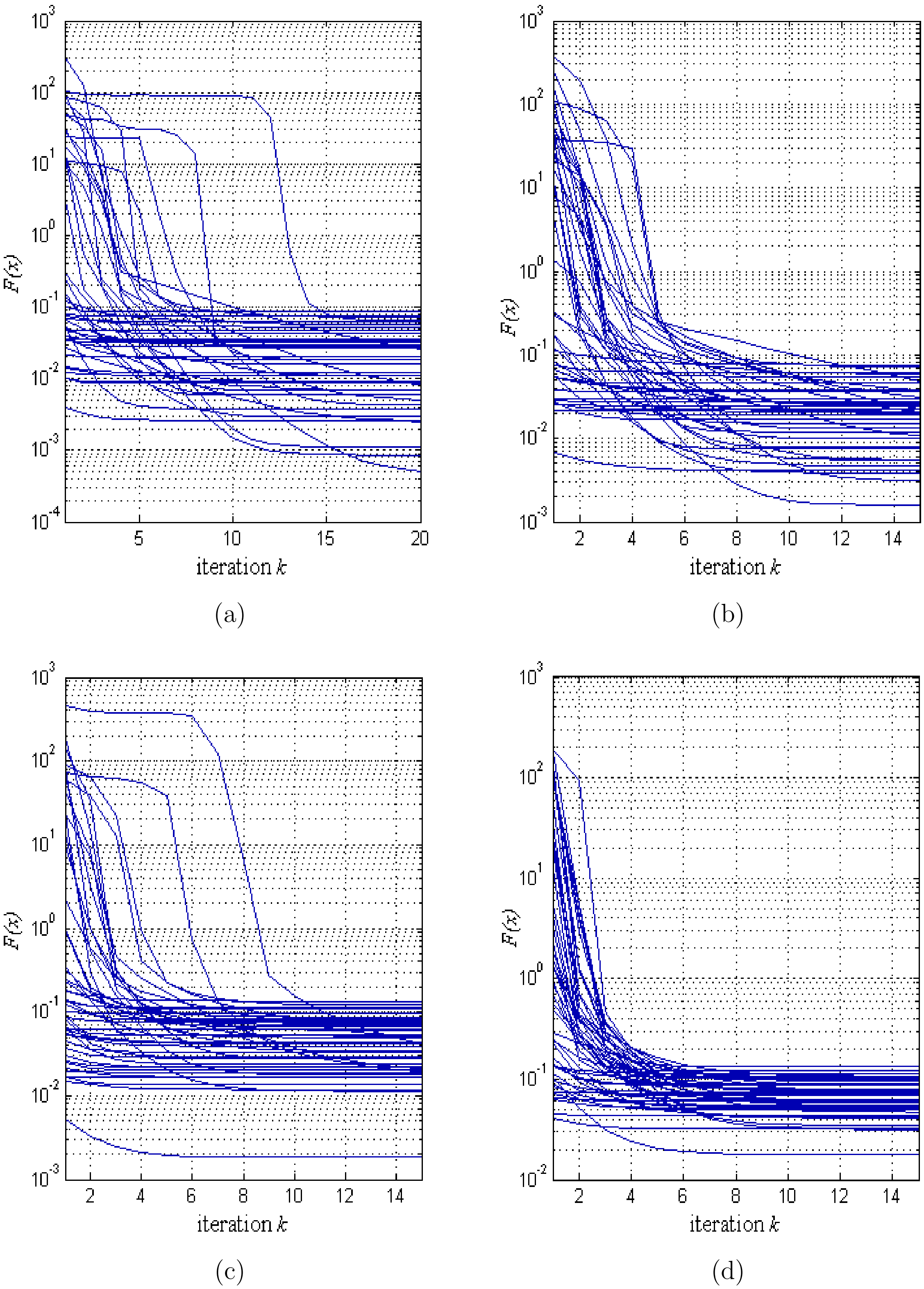


Figure 3.4: Convergernce of the proposed PCCP-based LS for random initializations with $\sigma = 10^{-1}$ for (a) 4 sensor nodes, (b) 5 sensor nodes, (c) 7 sensor nodes, and (d) 10 sensor nodes.

Chapter 4

Least Squares Localization by Sequential Convex Relaxation

In this chaper we adress the problems of localizing a single radiating source based on range or range-difference measurements. In both cases we focus on the efficient computation of the least squares estimates of the source coordinates. We start with converting the unconstrained LS problem to constrained problems. We then develop solution methods for these nonconvex problems that are ispired by the sequential convex programming based on exact penalty formulation and Taylor expansion. In Sec. 4.1 we describe the algorithm for the range-difference based localization. Then in Sec. 4.2 we consider the problem of range-based localization. Numerical results, that conclude each section, are presented to demonstrate that the proposed algorithms offer substantial performance improvement relative to some best known results from the literature.

4.1 Range-Difference Localization

4.1.1 Problem Formulation

In this section we focus on the problem of range-difference based localization given the time-difference of arrival information. TDOA localization, also known as multilateration, or hyperbolic positioning, is a method where the position of the mobile unit (signal source) can be determined using the differences in the TOAs from different base stations. By using this method the clock biases between the mobile units and base stations are automatically removed, since only the pairwise differences between the TOAs from base stations are considered [44]. A hyperbola is the basis for solving multilateration problems. In particular, the set of possible positions of a mobile unit that has a range difference of d_i from two given base stations BS_i and BS_0, placed at $\boldsymbol{a}_i$ and $\boldsymbol{a}_0$ respectively, is a hyperbola with vertex separation of d_i and focii located at $\boldsymbol{a}_i$ and $\boldsymbol{a}_0$. BS_0 is placed at the origin of the coordinate system, i.e. $\boldsymbol{a}_0 = \mathbf{0}_n$, and used as a reference station. Consider now a third base station BS_j at a third location. This would provide one extra independent measurement between BS_j and BS_0 and the source is located on the curve determined by the two intersecting hyperboloids. Figure 4.1 illustrates an example of the range-difference localization based on TDOA measurements.

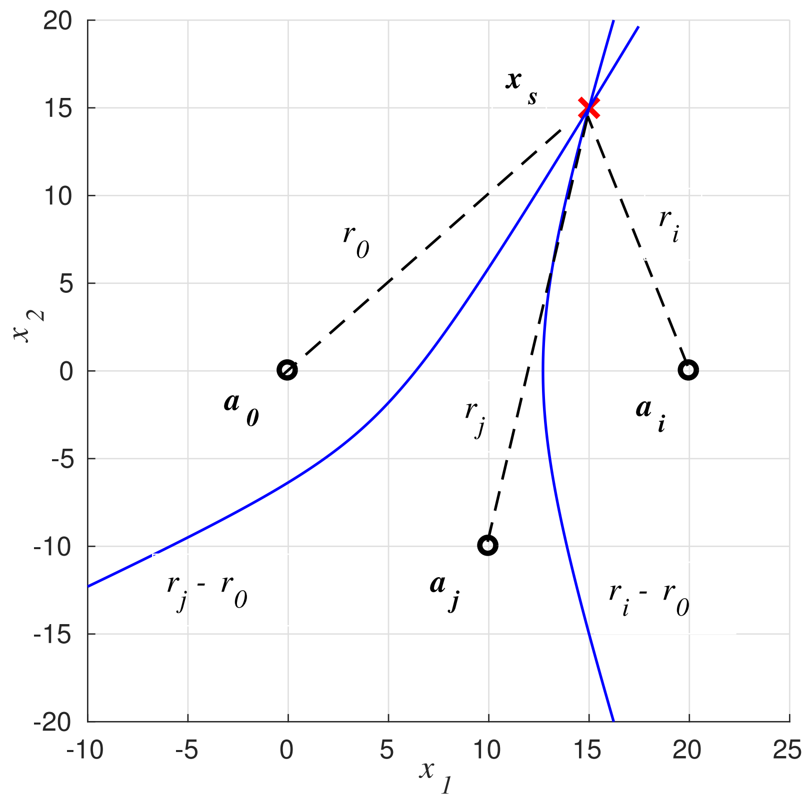


Figure 4.1: Range-difference localization. At least three base stations are required for the planar localization. The red cross indicates the location of the signal source. Sensors are placed at $\boldsymbol{a}_j = (10, -10)^T$, and $\boldsymbol{a}_0$ is the reference sensor. The time (range) differences $r_j - r_0$ and $r_i - r_0$ form two hyperboloids with focii located at $\boldsymbol{a}_i, \boldsymbol{a}_j$ and $\boldsymbol{a}_0$. Note that the hyperboloids are actually double sheeted, but for visual clarity only the halves which are part of the solution are shown. The intersection of these hyperboloids is the estimated position. The figure depicts the locus of possible source locations as one half of a two-sheeted hyperboloids.

The localization problem discussed in this section involves a given array of $m+1$ sensors placed in the $n = 2$ or 3 dimentional space with coordinates specified by $\{\boldsymbol{a}_1, \boldsymbol{a}_2, ..., \boldsymbol{a}_m, a_i \in R^n\}$ and $\boldsymbol{a}_0 = \mathbf{0}_n$ placed at the origin and used as a refference sensor. The localization problem here is to estimate the location of a radiating source $\boldsymbol{x}$ given the locations of the $m+1$ sensors and noise-contaminated range-difference measurements $\{d_i, i = 1, 2, \ldots, m\}$ where

$$d_i = \|\boldsymbol{x} - \boldsymbol{a}_i\| - \|\boldsymbol{x}\| + \varepsilon_i, \text{ for } i = 1, 2, \ldots, m \tag{4.1}$$

Therefore, the standard range-difference LS (RD-LS) problem is formulated as

$$\text{minimize} \sum_{i=1}^{m} (\|\boldsymbol{x} - \boldsymbol{a}_i\| - \|\boldsymbol{x}\| - d_i)^2 \tag{4.2}$$

As described in Sec.2.2 of the thesis, finding the solution to (4.2) is a non-trivial problem and many approaches have been developed to address it. In the following we propose a new iterative procedure to tackle the RD-LS problem (4.2), with the goal of achieving a more accurate and robust solution. The central part of the procedure is a convex quadratic programming (QP) problem that needs to be solved in each iteration to provide an increment vector that updates the present iterate to next towards the solution of the localization problem at hand.

4.1.2 Sequential Convex Relaxation

We begin by re-writing the unconstrained problem in (4.2) as a constrained problem with second-order cone constraints

$$\underset{\boldsymbol{x}, y, \boldsymbol{z}}{\text{minimize}} \quad \sum_{i=1}^{m} (z_i - y - d_i)^2 \tag{4.3a}$$

$$\text{subject to:} \quad \|\boldsymbol{x} - \boldsymbol{a}_i\| = z_i, \quad i = 1, 2, \ldots m \tag{4.3b}$$

$$\|\boldsymbol{x}\| = y \tag{4.3c}$$

Assume we are in the kth iteration and we are to update the kth iterate $\{\boldsymbol{x}_k, y_k, \boldsymbol{z}_k\}$. Let the next iterate be

$$\begin{aligned}
\boldsymbol{x}^{k+1} &= \boldsymbol{x}^k + \boldsymbol{\delta}_x && \text{(4.4a)}\\
y^{k+1} &= y^k + \delta_y && \text{(4.4b)}\\
\boldsymbol{z}^{k+1} &= \boldsymbol{z}^k + \boldsymbol{\delta}_z && \text{(4.4c)}
\end{aligned}$$

where $\{\boldsymbol{\delta}_x, \delta_y, \boldsymbol{\delta}_z\}$ are such that the objective function in (4.4a) is reduced and, at the same time, the constraints in (4.3b) and (4.3c) are better approximated at $\{\boldsymbol{x}_{k+1}, y_{k+1}, \boldsymbol{z}_{k+1}\}$ in the sense that

$$\begin{aligned}
\|\boldsymbol{x}_{k+1} - \boldsymbol{a}_i\| &\approx z_i^{k+1}, \quad i = 1, 2, \ldots, m\\
\|\boldsymbol{x}_{k+1}\| &\approx y_{k+1}
\end{aligned}$$

namely,

$$\begin{aligned}
\|\boldsymbol{x}_k + \boldsymbol{\delta}_x - \boldsymbol{a}_i\| &\approx z_i^k + \boldsymbol{\delta}_{z_i}, \quad i = 1, 2, \ldots, m\\
\|\boldsymbol{x}_k + \boldsymbol{\delta}_x\| &\approx y_k + \boldsymbol{\delta}_y
\end{aligned}$$

By replacing the left-hand sides of the above equations with their first-order Taylor approximations, we obtain

$$\begin{aligned}
\|\boldsymbol{x}_k - \boldsymbol{a}_i\| + \partial_x^T \|\boldsymbol{x}_k - \boldsymbol{a}_i\| \boldsymbol{\delta}_x &\approx z_i^k + \boldsymbol{\delta}_{z_i}, \quad i = 1, 2, \ldots, m\\
\|\boldsymbol{x}_k\| + \partial_x^T \|\boldsymbol{x}_k\| \boldsymbol{\delta}_x &\approx y_k + \boldsymbol{\delta}_y
\end{aligned}$$

where ∂_x is the subdifferential operator with respect to variable $\boldsymbol{x}$. Assuming $\boldsymbol{x}_k \neq \boldsymbol{a}_i$ and $\boldsymbol{x}_k$ is nonzero, then

$$\partial_x \|\boldsymbol{x}_k - \boldsymbol{a}_i\| = \frac{\boldsymbol{x}_k - \boldsymbol{a}_i}{\|\boldsymbol{x}_k - \boldsymbol{a}_i\|} \text{ and } \partial_x \|\boldsymbol{x}_k\| = \frac{\boldsymbol{x}_k}{\|\boldsymbol{x}_k\|}$$

Hence

$$\begin{aligned}
\|\boldsymbol{x}_k - \boldsymbol{a}_i\| + \frac{(\boldsymbol{x}_k - \boldsymbol{a}_i)^T \boldsymbol{\delta}_x}{\|\boldsymbol{x}_k - \boldsymbol{a}_i\|} &\approx z_i^{(k)} + \delta_{z_j}, \quad i = 1, 2, ..., m && \text{(4.5a)}\\
\|\boldsymbol{x}_k\| + \frac{\boldsymbol{x}_k^T \boldsymbol{\delta}_x}{\|\boldsymbol{x}_k\|} &\approx y_k + \delta_y && \text{(4.5b)}
\end{aligned}$$

The objective in (4.3a) can be written as

$$\begin{aligned} F(\boldsymbol{x}_{k+1}) &= \sum_{i=1}^{m} \left(z_i^{(k)} + \delta_{z_i} - (y_k + \delta_y) - d_i \right)^2 \\ &= \sum_{i=1}^{m} \left(-\delta_y + \delta_{z_i} - \tilde{d}_i^{(k)} \right)^2 \end{aligned}$$

where

$$\tilde{d}_i^{(k)} = d_i - y_k - z_i^{(k)}$$

are grouped known constant terms. Based on the above, the problem to be solved in the kth iteration is formulated as

$$\underset{\tilde{\boldsymbol{\delta}}}{\text{minimize}} \quad f(\tilde{\boldsymbol{\delta}}) = \sum_{i=1}^{m} \left(-\delta_y + \delta_{z_j} - \tilde{d}_i^{(k)} \right)^2 \tag{4.6a}$$

$$\text{subject to:} \quad \|\boldsymbol{x}_k - \boldsymbol{a}_i\| + \frac{(\boldsymbol{x}_k - \boldsymbol{a}_i)^T \boldsymbol{\delta}_x}{\|\boldsymbol{x}_k - \boldsymbol{a}_i\|} = z_i^{(k)} + \delta_{z_j}, \quad i = 1, 2, \ldots m \tag{4.6b}$$

$$\|\boldsymbol{x}_k\| + \frac{\boldsymbol{x}_k^T \boldsymbol{\delta}_x}{\|\boldsymbol{x}_k\|} = y_k + \delta_y \tag{4.6c}$$

$$\begin{bmatrix} -\beta \mathbf{1}_2 \\ -\beta \\ -\beta \mathbf{1}_m \end{bmatrix} \le \begin{bmatrix} \boldsymbol{\delta}_x \\ \delta_y \\ \boldsymbol{\delta}_z \end{bmatrix} \le \begin{bmatrix} \beta \mathbf{1}_2 \\ \beta \\ \beta \mathbf{1}_m \end{bmatrix} \tag{4.6d}$$

The constraints in (4.6d) assure that the magnitude of each component in $\{\boldsymbol{\delta}_x, \delta_y, \boldsymbol{\delta}_z\}$ is no greater than β. Obviously, the problem in (4.6) is a convex QP problem. One technical difficulty that may occur in solving problem (4.6) is that the feasible region defined by (4.6b), (4.6c), and (4.6d) may be empty. In such a case the constraints in problem (4.6) must be adequately relaxed in order for the problem to be solvable. To this end we express the problem in (4.6) in a more compact form as

$$\underset{\tilde{\boldsymbol{\delta}}}{\text{minimize}} \quad f\left(\tilde{\boldsymbol{\delta}}\right) \tag{4.7a}$$

$$\text{subject to} \quad \boldsymbol{A}_k \tilde{\boldsymbol{\delta}} = \boldsymbol{b}_k \tag{4.7b}$$

$$\boldsymbol{C} \tilde{\boldsymbol{\delta}} \le \boldsymbol{q} \tag{4.7c}$$

where the objective function is

$$f\left(\tilde{\boldsymbol{\delta}}\right) = \|\boldsymbol{S}\tilde{\boldsymbol{\delta}} - \tilde{\boldsymbol{d}}_k\|_2$$

and

$$\boldsymbol{S} = \begin{bmatrix} \boldsymbol{0}_{m\times 1} & -\boldsymbol{1}_{m\times 1} & -\boldsymbol{I}_m \end{bmatrix}, \tilde{\boldsymbol{\delta}} = \begin{bmatrix} \boldsymbol{\delta}_x \\ \delta_y \\ \boldsymbol{\delta}_z \end{bmatrix}, \boldsymbol{A}_k = \begin{bmatrix} \boldsymbol{A}_1^{(k)} & \boldsymbol{0}_{m\times 1} & -\boldsymbol{I}_m \\ \boldsymbol{x}_k & -1 & \boldsymbol{0}_{1\times m} \end{bmatrix} \tag{4.8a}$$

$$\boldsymbol{A}_1^{(k)} = \begin{bmatrix} \frac{(\boldsymbol{x}_k - \boldsymbol{a}_1)^T}{\|\boldsymbol{x}_k - \boldsymbol{a}_1\|} \\ \frac{(\boldsymbol{x}_k - \boldsymbol{a}_2)^T}{\|\boldsymbol{x}_k - \boldsymbol{a}_2\|} \\ \vdots \\ \frac{(\boldsymbol{x}_k - \boldsymbol{a}_m)^T}{\|\boldsymbol{x}_k - \boldsymbol{a}_m\|} \end{bmatrix}, \quad \boldsymbol{b}_k = \begin{bmatrix} z_1^{(k)} - \|\boldsymbol{x}_k - \boldsymbol{a}_1\| \\ z_2^{(k)} - \|\boldsymbol{x}_k - \boldsymbol{a}_2\| \\ \vdots \\ z_m^{(k)} - \|\boldsymbol{x}_k - \boldsymbol{a}_m\| \\ y_k - \|\boldsymbol{x}_k\| \end{bmatrix} \tag{4.8b}$$

$$\tilde{\boldsymbol{d}}_k = \begin{bmatrix} d_1 + y^k - z_1^k \\ d_2 + y^k - z_2^k \\ \vdots \\ d_m + y^k - z_m^k \end{bmatrix}, \boldsymbol{C} = \begin{bmatrix} \boldsymbol{I}_{m+3} \\ -\boldsymbol{I}_{m+3} \end{bmatrix}, \boldsymbol{q} = \beta \boldsymbol{e} \tag{4.8c}$$

where $\boldsymbol{e} = \boldsymbol{1}_{2(m+3)}$ is the all-one vector of dimention $2(m+3)$. By introducing nonnegative slack variables $\boldsymbol{u}, \boldsymbol{v}$, and w, we relax the problem in (4.7) to

$$\begin{aligned} \underset{\boldsymbol{\delta}}{\text{minimize}} \quad & f\left(\tilde{\boldsymbol{\delta}}\right) + \tau \sum_{i=1}^{m+1} (u_i + v_i) + \tau w & (4.9\text{a}) \\ \text{subject to} \quad & \boldsymbol{A}_k \tilde{\boldsymbol{\delta}} - \boldsymbol{b}_k = \boldsymbol{u} - \boldsymbol{v} & (4.9\text{b}) \\ & \boldsymbol{C}\tilde{\boldsymbol{\delta}} - \boldsymbol{q} \le w\boldsymbol{e} & (4.9\text{c}) \\ & \boldsymbol{u} \ge \boldsymbol{0}, \boldsymbol{v} \ge \boldsymbol{0}, w \ge 0 & (4.9\text{d}) \end{aligned}$$

where $\tau > 0$ is a sufficiently large scalar. It is easy to verify that the feasible region defined by (4.9b) - (4.9d) is always nonempty. For example, if we fix $\tilde{\boldsymbol{\delta}} = \tilde{\boldsymbol{\delta}}_0$ arbitrarily, then obviously the point $\{\tilde{\boldsymbol{\delta}}_0, \boldsymbol{u}_0, \boldsymbol{v}_0, w_0\}$ with

$$\boldsymbol{u}_0 = \max\{\boldsymbol{0}, \boldsymbol{A}_0\tilde{\boldsymbol{\delta}}_0 - \boldsymbol{b}_0\}, \quad \boldsymbol{v}_0 = \max\{\boldsymbol{0}, -(\boldsymbol{A}_0\tilde{\boldsymbol{\delta}}_0 - \boldsymbol{b}_0)\}, \text{and } w_0 = \max\{0, \boldsymbol{C}\tilde{\boldsymbol{\delta}}_0 - \boldsymbol{q}\}$$

is a feasible point for problem (4.9). Penalty terms $\tau \sum_{i=1}^{m+1} (u_i + v_i) + \tau w$ are included in the objective function in (4.9a) in order to reduce the magnitudes of the slack variables while minimizing the original objective function. If the solution slack variables turn out to be all zero, then the solution of (4.9) also solves problem (4.7). Otherwise, we conclude that problem (4.7) in not solvable and the solution obtained by solving (4.9) is a reasonable candidate for the kth iteration to update $\{\boldsymbol{x}_k, y_k, \boldsymbol{z}_k\}$.

4.1.3 The Algorithm

The input parameters for the algorithm include the bound β on the increment vector $\tilde{\boldsymbol{\delta}} = (\boldsymbol{\delta}_x, \delta_y, \boldsymbol{\delta}_z)$. It controls the size of a "trust region" over which a convex subproblem is carried out and is responsible for the performance of the algorithm. If parameter β is chosen too large, then approximation $\boldsymbol{x}_{k+1}$ will perform poorly because in this case the convex model is less accurate [41]. If β is too small the progress of the algorithm will be too slow. In our simulations a β between 3 and 6 was found to work very well. Other input parameters are initial point $\boldsymbol{x}_0$, initial weight for penalty terms τ_0, and upper limit of the weight τ_{max} (to avoid numerical problems that may occur if τ_k becomes too large).

We remark that the method proposed in this section is *local*, therefore is not guaranteed to converge to the global minimum when applied to the nonconvex problem. As discussed in Sec. 2.2 of the thesis, the original LS objective in (4.2) is highly nonconvex with many local minimums even for small-scale systems. Consequently, the choise of a good initial point is critical to ensure good performance of the method. Several techniques are available, these include: i) set the initial point to the origin; ii) select the initial point uniformly randomly over the same region as the unknown radiating source; iii) run the algorithm from a set of candidate initial points and identify the solution as the one with lowest LS error. The candidate initial points can be selected either randomly over the region or based on some a priori information, for example, the geometry of the sensor network; iv) solve an approximating problem or apply a global localization algorithm such as those in [4] to generate an approximate LS solution, then take it as the initial point to run the proposed algorithm.

Based on the analysis above, the localization algorithm for range-difference measurements can be outlined as follows. For the ease of notation we will refer to the proposed algorithm as Sequential Convex Relaxation for Range-Difference-based Least Squares (SCR-RDLS).

Algorithm 4. Sequential Convex Relaxation for Range-Difference Localization

1) Input data: Sensor locations $\{\boldsymbol{a}_i, i = 1, \ldots, m\}$, range-difference measurements $\{d_i, i = 1, \ldots, m\}$, initial point $\boldsymbol{x}_0$, initial weight τ_0 and upper limit of weight τ_{max}, increment bound β and convergence tolerance ϵ. Set iteration count to $k = 0$. Form $\boldsymbol{S}, \boldsymbol{C}$ and $\boldsymbol{q}$ as

$$\boldsymbol{S} = \begin{bmatrix} \boldsymbol{0}_{m\times 1} & -\boldsymbol{1}_{m\times 1} & -\boldsymbol{I}_m \end{bmatrix}, \boldsymbol{C} = \begin{bmatrix} \boldsymbol{I}_{m+3} \\ -\boldsymbol{I}_{m+3} \end{bmatrix}, \boldsymbol{q} = \beta \boldsymbol{e}$$

2) Form y_k and $\boldsymbol{z}_k$ as

$$y_k = \|\boldsymbol{x}_k\|, \boldsymbol{z}_k = \begin{bmatrix} \|\boldsymbol{x}_k - \boldsymbol{a}_1\| \\ \vdots \\ \|\boldsymbol{x}_k - \boldsymbol{a}_m\| \end{bmatrix}$$

form $\boldsymbol{A}_k, \tilde{\boldsymbol{d}}_k, \boldsymbol{b}_k$ and $\boldsymbol{C}_k$ as in (4.8) and solve

$$\begin{aligned} \underset{\boldsymbol{\delta}}{\text{minimize}} \quad & f\left(\tilde{\boldsymbol{\delta}}\right) + \tau_k \sum_{i=1}^{m+1} (u_i + v_i) + \tau_k w \\ \text{subject to} \quad & \boldsymbol{A}_k \tilde{\boldsymbol{\delta}} - \boldsymbol{b}_k = \boldsymbol{u} - \boldsymbol{v} \\ & \boldsymbol{C}\tilde{\boldsymbol{\delta}} - \boldsymbol{q} \leq w\boldsymbol{e} \\ & \boldsymbol{u} \geq \boldsymbol{0}, \boldsymbol{v} \geq \boldsymbol{0}, w \geq 0 \end{aligned}$$

Denote the solution as $\tilde{\boldsymbol{\delta}}_k = (\boldsymbol{\delta}_x^*, \delta_y^*, \boldsymbol{\delta}_z^*)$.
3) Update $\tau_{k+1} = \min\,(1.5\tau_k, \tau_{max})$, set $k = k + 1$. Update $\tilde{\boldsymbol{x}}^*$ to

$$\begin{aligned} \boldsymbol{x}^* &= \boldsymbol{x}^k + \boldsymbol{\delta}_x^* \\ y^* &= y^k + \delta_y^* \\ \boldsymbol{z}^* &= \boldsymbol{z}^k + \boldsymbol{\delta}_z^* \end{aligned}$$

4) If $\|\tilde{\boldsymbol{\delta}}_k\| \leq \epsilon$, terminate and output $\boldsymbol{x}^*$ as the solution; otherwise, set $\tilde{\boldsymbol{x}}_k = \boldsymbol{x}^*$ and repeat from Step 2.

4.1.4 Numerical Results

Performance of the proposed algorithm was evaluated and compared with existing state-of-the-art methods by Monte Carlo simulations with a set-up similar to that of [4]. SRD-LS solutions were used as performance benchmarks for the Algorithm 4. The sensor network consisted of 11 sensors with $\boldsymbol{a}_0 = \mathbf{0}$ placed at the origin and other ten sensors $\{\boldsymbol{a}_i, i = 1, 2, \ldots, 10\}$ randomly placed in the planar region in $[-15; 15] \times [-15; 15]$. A radiating source $\boldsymbol{x}_s$ was located randomly in the region $\{\boldsymbol{x} = [x_1; x_2], -10 \leq x_1, x_2 \leq 10\}$. The coordinates of the source and sensors were generated for each dimension following a uniform distribution. Measurement noise $\{\varepsilon_i, i = 1, \ldots, m\}$ was modelled as independent and identically distributed (i.i.d) random variables with zero mean and variance σ^2, with σ being one of four possible levels $\{10^{-3}, 10^{-2}, 10^{-1}, 1\}$. The range-difference measurements $\{d_i, i = 1, 2, \ldots, 5\}$ were calculated using (4.1). Accuracy of source location estimation was evaluated in terms of average of the squared position error error in the form $\|\boldsymbol{x}^* - \boldsymbol{x}_s\|^2$, where $\boldsymbol{x}_s$ denotes the exact source location and $\boldsymbol{x}^*$ is its estimation obtained by SR-LS and proposed methods, respectively. In our simulations parameter β was set to 3, the initial penalty term τ_0, and upper limit of the penalty term τ_{max} were set to 10 and 10 000 respectively. The proposed method was implemented by using CVX [16] and implementation of SRD-LS followed [4]. In our simulations we set the inital point $\boldsymbol{x}_0$ to the solution of the following subproblem

$$\begin{aligned} \underset{\boldsymbol{x}, y, \boldsymbol{z}}{\text{minimize}} \quad & \sum_{i=1}^{m} (z_i - y - d_i)^2 \\ \text{subject to:} \quad & \|\boldsymbol{x} - \boldsymbol{a}_i\| \leq z_i, \quad i = 1, 2, \ldots m \\ & \|\boldsymbol{x}\| \leq y \end{aligned}$$

The iterations proceeded until the magnitude of the increment vector $\|\tilde{\boldsymbol{\delta}}_k\|$ falls below a convergence tolerance ϵ which was set to $\epsilon = 10^{-6}$.

Table 4.1 provides comparisons of the SCR-RDLS with SRD-LS, where each entry is averaged squared error over 1,000 Monte Carlo runs of the method. It is observed that, comparing with SR-LS, the estimates produced by the proposed algorithm are found to be closer to the true source locations in MSE sense. The last column of the table represents relative improvement (R.I.) of the proposed method over SRD-LS solutions in percentage. Further analysis of the data that was used to generate Table 4.1 illustrates the advantage of the SCR-RDLS solution over the SR-LS. Each entry in Table 4.2 is a standard deviation of the squared estimation errors aggregated over the same 1,000 Monte Carlo runs described above in Table 4.1 where the MSEs of the position estimation are shown. The results summarised in Table 4.2 indicate that the SCR-RDLS algorithm provides considerable performance improvement relative to the SRD-LS algorithm.

Table 4.1: MSE of position estimation for SRD-LS and SCR-RDLS methods

σ	SRD - LS	SCR-RDLS	(R.I.,%)
1e-03	1.2655e-06	8.4626e-07	33
1e-02	1.4492e-04	6.8385e-05	52
1e-01	1.3329e-02	7.1676e-03	46
1e+0	1.6077e+00	9.5371e-01	40

Table 4.2: Standard deviation of the squared position estimation error for SRD-LS and SCR-RDLS methods

σ	SRD - LS	SCR-RDLS
1e-03	3.3990e-06	3.2917e-06
1e-02	5.9818e-04	1.3673e-04
1e-01	3.6154e-02	2.4337e-02
1e+0	4.2206e+00	3.4931e+00

4.2 Range-based localization

In this section we explore a sequential convex relaxation approach to the range based localization problem that you had studied in previous chapters. The localization problem discussed in this section involves a given array of m sensors placed in the $n = 2$ or 3 dimentional space with coordinates specified by $\{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_m, \boldsymbol{a}_i \in R^n\}$. Each sensor measures its distance to a radiating source $\boldsymbol{x} \in R^n$. Throughout it is assumed that only noisy copies of the distance data are available, hence the *range measurements* obey the model

$$r_i = \|\boldsymbol{x} - \boldsymbol{a}_i\| + \varepsilon_i, \quad i = 1, \ldots, m.$$

where ε_i denotes the unknown noise that has occurred when the ith sensor measures its distance to source $\boldsymbol{x}$. The localization problem can be stated as to estimate the *unknown* location of the radiating device, $\boldsymbol{x}$, by solving the nonlinear least squares problem

$$\underset{\boldsymbol{x}}{\text{minimize}} \sum_{i}^{m} \left(\|\boldsymbol{x} - \boldsymbol{a}_i\| - r_i\right)^2 \tag{4.10}$$

4.2.1 Sequential Relaxation

The problem in (4.10) can be (equivalently) written as

$$\begin{aligned}
\underset{\boldsymbol{x},\boldsymbol{z}}{\text{minimize}} \quad & \sum_{i}^{m} (z_i - r_i)^2 && \text{(4.11a)} \\
\text{subject to:} \quad & \|\boldsymbol{x} - \boldsymbol{a}_i\| = z_i, \quad i = 1, 2, ..., m && \text{(4.12b)} \\
& \boldsymbol{z} \geq \boldsymbol{0} && \text{(4.12c)}
\end{aligned}$$

The constraints in (4.11b) are difficult to satisfy exactly. This motivates a relaxed set of constraints that lead to the following constrained problem

$$\begin{aligned}
\underset{\boldsymbol{x},\boldsymbol{z}}{\text{minimize}} \quad & \sum_{i}^{m} (z_i - r_i)^2 && \text{(4.13a)} \\
\text{subject to:} \quad & \|\boldsymbol{x} - \boldsymbol{a}_i\| \leq (1+\gamma) z_i && \text{(4.13b)} \\
& \|\boldsymbol{x} - \boldsymbol{a}_i\| \geq (1-\gamma) z_i, \quad i = 1, 2, ..., m && \text{(4.13c)}
\end{aligned}$$

where $\gamma > 0$ is a small scalar, initially set in the range $\gamma_0 \in (0, 0.5)$. The solution of the problem (4.13) may be taken as an approximate solution of the problem in (4.11), hence of problem (4.10) as well. By allowing γ to sequentially and monotonically decrease from γ_0 to zero, the solution of the problem in (4.13) converges that of the problem in (4.11), because as γ approaches 0, the feasible region of the problem in (4.13) will become equivalent to that in (4.11). As iterations proceed, the objective in (4.13) will not be monotonically decreasing in general but it will converge to a critical point.

Problem in (4.13) is nonconvex due to the nonconvexity of some of the inequality constraints. To be precise, the constraints in (4.13b), namely $\|\boldsymbol{x} - \boldsymbol{a}_i\| \leq (1+\gamma)z_i$, are convex, while those in (4.13c) are not

$$\|\boldsymbol{x} - \boldsymbol{a}_i\| \geq (1-\gamma)z_i \Longleftrightarrow \underbrace{-\|\boldsymbol{x} - \boldsymbol{a}_i\|}_{nonconvex} \leq -(1-\gamma)z_i$$

Because the norm $\|\boldsymbol{x} - \boldsymbol{a}_i\|$ is a convex function with respect to $\boldsymbol{x}$, it follows that for a *known* $\boldsymbol{x}_k$

$$\|\boldsymbol{x} - \boldsymbol{a}_i\| \geq \|\boldsymbol{x}_k - \boldsymbol{a}_i\| + \partial\|\boldsymbol{x}_k - \boldsymbol{a}_i\|^T(\boldsymbol{x} - \boldsymbol{a}_i)$$

Hence the constraint in 4.13c can be replaced with its affine approximation

$$-\|\boldsymbol{x}_k - \boldsymbol{a}_i\| - \partial\|\boldsymbol{x}_k - \boldsymbol{a}_i\|^T(\boldsymbol{x} - \boldsymbol{a}_i) \leq -(1-\gamma)z_i$$

In this way, in the kth iteration when the iterate $\boldsymbol{x}_k$ is known, the nonconvex problem in 4.13 can be converted to a convex second-order cone programming (SOCP) problem

$$\underset{\boldsymbol{x},\boldsymbol{z}}{\text{minimize}} \quad \sum_{i}^{m} (z_i - r_i)^2 \tag{4.14a}$$

$$\text{subject to:} \quad \|\boldsymbol{x} - \boldsymbol{a}_i\| \leq (1+\gamma)z_i \tag{4.14b}$$

$$-\|\boldsymbol{x}_k - \boldsymbol{a}_i\| - \partial\|\boldsymbol{x}_k - \boldsymbol{a}_i\|^T(\boldsymbol{x} - \boldsymbol{a}_i) \leq -(1-\gamma)z_i, \quad i = 1, 2, ..., m \tag{4.14c}$$

where the relaxation parameter γ controls the size of the convex hull that defines a feasibility region of the problem (4.14), and γ needs to be monotonically decreasing as the iteration proceeds. Specifically, a reasonable choice of the value for γ_0 is 0.2 or

0.3, and we update the value of γ_{k+1} linearly as

$$\gamma_{k+1} = \gamma_0 - k\frac{\gamma_0}{K_{max} - 1} \tag{4.15}$$

or quadratically as

$$\gamma_{k+1} = \gamma_0 \frac{(K_{max} - 1 - k)^2}{(K_{max} - 1)^2} \tag{4.16}$$

where K_{max} is the number of iterations to be executed.

For the ease of notation the proposed algorithm will be reffered to as Sequential Convex Relaxation for Range-based Least Squares (SCR-RLS). It is important to note that the SCR-RLS is a local method and its performance often depend on the choise of the initial point. Several available techniques described in Sec.4.1.3 of the thesis can be applied here. Summarizing the analysis above, the localization algorithm based on range measurements can be outlined as follows.

Algorithm 5. Sequential Convex Relaxation for Range-based Localization

1) Input data: Sensor locations $\{\boldsymbol{a}_i, i = 1, \ldots, m\}$, range measurements $\{r_i, i = 1, \ldots, m\}$, initial point $\boldsymbol{x}_0$, initial relaxation parameter γ_0, and the number of iterations to be executed K_{max}. Set iteration count to $k = 0$.
2) Solve

$$\begin{aligned} \underset{\boldsymbol{x},\boldsymbol{z}}{\text{minimize}} \quad & \sum_{i}^{m} (z_i - r_i)^2 \\ \text{subject to:} \quad & \|\boldsymbol{x} - \boldsymbol{a}_i\| \leq (1+\gamma) z_i \end{aligned}$$

$$-\|\boldsymbol{x}_k - \boldsymbol{a}_i\| - \partial\|\boldsymbol{x}_k - \boldsymbol{a}_i\|^T(\boldsymbol{x} - \boldsymbol{a}_i) \leq -(1-\gamma) z_i, \;\; i = 1, 2, ..., m$$

Denote the solution as $\tilde{\boldsymbol{x}}_k = (\boldsymbol{x}^*, \boldsymbol{z}^*)$.
3) Update $\gamma_{k+1} = f(\gamma_k)$ linearly, as in (4.15), or quadratically, as in (4.16). Set $k = k + 1$.
4) If $k = K_{max}$, terminate and output $\boldsymbol{x}^*$ as the solution; otherwise, set $\boldsymbol{x}_k = \boldsymbol{x}^*$ and repeat from Step 2.

4.2.2 Numerical Results

Performance of the proposed SCR-RLS algorithm was evaluated and compared with existing state-of-the-art SR-LS solutions [4] with a set-up similar to that of [4]. Our simulation studies of Algorithm 5 considered a scenario that consists of $m = 5$ sensors $\{\boldsymbol{a}_i,\ i = 1, 2, \ldots, m\}$ randomly placed in the planar region in $[-15; 15] \times [-15; 15]$, and a radiating source $\boldsymbol{x}_s$, located randomly in the region $[-10; 10] \times [-10; 10]$. Coordinates of the source and sensors were generated for each dimension following a uniform distribution. Measurement noise $\{\varepsilon_i,\ i = 1, \ldots, m\}$ was modelled as i.i.d. Gaussian random variables with zero mean and variance σ^2, with σ being one of three possible levels $\{10^{-3}, 10^{-2}, 10^{-1}\}$. The range measurements $\{r_i,\ i = 1, 2, \ldots, m\}$ were calculated using

$$r_i = \|\boldsymbol{x} - \boldsymbol{a}_i\| + \varepsilon_i, \quad i = 1, \ldots, m.$$

Accuracy of source location estimation was evaluated as the average of the squared position error $\|\boldsymbol{x}^* - \boldsymbol{x}_s\|^2$ where $\boldsymbol{x}_s$ denotes the exact source location and $\boldsymbol{x}^*$ is its estimation obtained by SR-LS and SCR-RLS methods, respectively. Implementation of SR-LS followed [4] and the method proposed in this section was implemented using CVX [16]. In our simulations, we set the value of γ_0 to 0.2 and the number of iterations K_{max} to 9. The SCR-RLS algorithm was initialized with the midpoint between the two sensors associated with the two smallest distance readings.

Table 4.3 provides comparisons of the proposed sequential relaxation method with SR-LS, where each table entry is a squared error averaged over 200 Monte Carlo runs of a given method for a given noise level. The last column of the table represents relative improvement (R.I.) of the proposed method over SR-LS solutions in percentage. Table 4.4 compares a standard deviation of the squared estimation errors aggregated over the same 200 Monte Carlo runs described above in Table 4.3 where the MSE of the position estimation are shown. The results summarised in Table 4.3 and Table 4.4 indicate that the SCR-RLS algorithm yields improved performance relative to the SR-LS algorithm.

Table 4.3: MSE of position estimation for SR-LS and SCR-RLS methods

σ	SR - LS	SCR-RLS	(R.I.,%)
1e-02	2.5360e-04	2.0596e-04	18
1e-01	1.8696e-02	1.4802e-02	21
1e+0	1.4440e+00	9.6327e-01	33

Table 4.4: Standard deviation of the squared position estimation error for SR-LS and SCR-RLS methods

σ	SR - LS	SCR-RLS
1e-02	1.1711e-03	3.7880e-04
1e-01	3.5513e-02	2.4685e-02
1e+0	1.7519e+00	9.6863e-01

Chapter 5

Conclusion

5.1 Conclusion

5.2 Future Work

Influense of sensor geometry on the accuracy of the proposed methods. For example, one of the highest impacts on accuracy degradation is Geometric Dilusion of Presicion. Optimum sensor arrangement is a separate research area.

Multi-source localization, when the range measurements were obtained from the energy-based readings.

Appendix A

Matlab Code Listings

A.1 Iterative Re-Weighting LS for Source Localization Using Range Measurements

A.1.1 IRWSR-LS Algorithm

The followng m-files implement the iterative re-weighting range-based least squares (IRWSR-LS) algorithm developed in Sec. 2.1 of the thesis.

Input parameters

Am $= [\boldsymbol{a}_1\ \boldsymbol{a}_2 \ldots \boldsymbol{a}_m]$, matrix of sensor locations, where $\boldsymbol{a}_i$ is a column vector that stores the coordinates of the ith sensor.

$\boldsymbol{r}_n = [r_1\ r_2 \ldots r_m]^T$, noisy range measurements, where r_i is a measurement obtained from the ith sensor.

Tunable parameters

epsi, convergence tolerance for the bisection routine.

K, number of iterations for re-weighting procedure.

Otput parameters

$\boldsymbol{x}_w$, estimated location of the source.

Calling sequence

Function sr_ls_irw.m calls gtrs_r.m to find the solution of the GTRS problem (2.16) in Step 3 of the algorithm.

```
function xw = sr_ls_irw(Am,rn,epsi)
[n,m] = size(Am);
```

```
A = [-2*Am' ones(m,1)];
b = zeros(m,1);
for i = 1:m,
    ai = Am(:,i);
    b(i) = rn(i)^2 - ai'*ai;
end
z = zeros(n,1);
D = [eye(n) z; z' 0];
f = [z; -0.5];
yk = gtrs_r(A,b,D,f,epsi);
xk = yk(1:n);
wk = zeros(m,1);
ep = 1e-3;
err=1;
Kmax=10;
K=1;
while err >= 1e-9 &&  K <= Kmax
    for j = 1:m,
        nj = abs(norm(xk-Am(:,j))+rn(j));
        if nj < ep,
            wk(j) = 1/ep;
        else
            wk(j) = 1/nj;
        end
    end
    Gk = diag(wk);
    Agk = Gk*A;
    bgk = Gk*b;
    yk = gtrs_r(Agk,bgk,D,f,epsi);
    err=norm(xk-yk(1:n));
    xk = yk(1:n);
    K=K+1;
end
xw = xk;

function y = gtrs_r(A,b,D,f,epsi)
% Find the left end of the interval I in (2.16)
A1 = A'*A;
atb = A'*b;
[U1,D1] = eig(A1);
d1 = diag(D1);
d1i = 1./sqrt(d1);
Si = U1*diag(d1i);
q = -1/max(eig(Si'*D*Si));
bL = q + 1e-26;
bu0 = abs(bL);
% Get an upper bound of lambda
bU = bu0;
yt = inv(A1+bU*D)*(atb-bU*f);
phit = (D*yt + 2*f)'*yt;
while phit > 1e-3,
    bU = bU + bu0;
    yt = inv(A1+bU*D)*(atb-bU*f);
    phit = (D*yt + 2*f)'*yt;
end
% Perform bisectioning
L = bL;
U = bU;
```

```
dt = U - L;
while dt > epsi,
    t = 0.5*(L + U);
    dt = 0.5*dt;
    yt = inv(A1+t*D)*(atb-t*f);
    phit = (D*yt + 2*f)'*yt;
    if phit > 0,
       L = t;
    else
       U = t;
    end
end
t = 0.5*(L + U);
% the best value of lambda
% Solution of GTRS is obtained as:
y = inv(A1+t*D)*(atb-t*f);
```

A.1.2 Hybrid IRWSR-LS

The followng m-files implement the *hybrid* iterative re-weighting range-based least squares (IRWSR-LS) algorithm developed in Sec. 2.1 of the thesis.

Input parameters

$fname$, name of the function to calculate the value of the LS objective at a point.

$gname$, name of the function to calculate the gradient of the LS objective at a point.

$hname$, name of the function to calculate the Hessian of the LS objective at a point.

$\mathbf{Am} = [\boldsymbol{a}_1\ \boldsymbol{a}_2 \ldots \boldsymbol{a}_m]$, matrix of sensor locations, where $\boldsymbol{a}_i$ is a column vector that stores the coordinates of the ith sensor.

$\boldsymbol{r}_n = [r_1\ r_2 \ldots r_m]^T$, noisy range measurements, where r_i is a measurement obtained from the ith sensor.

$\boldsymbol{x}_0$, solution of the sr_ls_irw.m as an initial point for the Newton algorithm.

Tunable parameters

dt, small positive constant to modify the Hessian $\boldsymbol{H}(\boldsymbol{x})$ of the objective in case it is not positive definite.

$epsi$, convergence tolerance.

Otput parameters

$\boldsymbol{x}_s$, estimated location of the source.

k, number of iterations it took the algorithm to converge to the solution point with the convergence tolerance $epsi$.

f_s, value of the LS objective at the estimated soluion point.

Calling sequence

Function newton.m calls inex_lsearch.m to perform the line search and Hx_R_LS.m to evaluate the Hessian of the LS objective at a point. Function inex_lsearch.m calls fx_R_LS.m and gx_R_LS.m to evaluate the objective and the gradient of the LS objective at a point respectively.

```
function [xs,fs,k] = newton(fname,gname,hname,Am, rn, x0,dt,epsi)
max_it = 1e+4;
k = 1;
[n,m] = size(Am);
xk = x0;
gk = feval(gname,Am, rn, xk);
[Hk, tauk] = feval(hname,Am, rn, xk);
[V,D] = eig(Hk);
di = diag(D)+tauk;
% % ensure that Hk is PD
dmin = min(di);
if dmin > 0
    Hki = V*diag(1./di)*V';
else
    ind = find(di<=0);
    di(ind) = dt;
    Hki = V*diag(1./di)*V';
end
dk = -Hki*gk;
ak = inex_lsearch(Am, rn, xk,dk,fname,gname);
adk = ak*dk;
er = norm(adk);
it = 0;
while (er >= epsi) && (it <= max_it)
    xk = xk + adk;
    gk = feval(gname,Am, rn, xk);
    [Hk, tauk] = feval(hname,Am, rn, xk);
    [V,D] = eig(Hk);
    di = diag(D)+tauk;
    dmin = min(di);
    dmin = min(di);
    if dmin > 0
        Hki = V*diag(1./di)*V';
    else
        for i=1:n
            di(i) = max(di(i), dt);
        end
        Hki = V*diag(1./di)*V';
    end
    dk = -Hki*gk;
    ak = inex_lsearch(Am, rn, xk,dk,fname,gname);
    adk = ak*dk;
    er = norm(adk);
    k = k + 1;
    it = it+1;
end
xs = xk + adk;
fs = feval(fname,Am, rn, xs);

function z = inex_lsearch(Am, rn, xk,s,fname,gname)
```

```
k = 0;
m = 0;
tau = 0.1;
chi = 0.75;
rho = 0.1;
sigma = 0.1;
mhat = 400;
epsilon = 1e-10;
xk = xk(:);
s = s(:);
% compute f0 and g0
    f0 = feval(fname,Am, rn, xk);
    gk = feval(gname,Am, rn, xk);
  m = m+2;
  deltaf0 = f0;
% step 2 Initialize line search
  dk = s;
  aL = 0;
  aU = 1e99;
  fL = f0;
  dfL = gk'*dk;
  if abs(dfL) > epsilon,
    a0 = -2*deltaf0/dfL;
  else
    a0 = 1;
 end
 if ((a0 <= 1e-9)|(a0 > 1)),
    a0 = 1;
 end
%step 3
 while 1,
    deltak = a0*dk;
    f0 = feval(fname, Am, rn, xk+deltak);
    m = m + 1;
%step 4
    if ((f0 > (fL + rho*(a0 - aL)*dfL)) ...
    & (abs(fL - f0) > epsilon) & (m < mhat))
      if (a0 < aU)
        aU = a0;
      end
      % compute a0hat using equation
      a0hat = aL + ((a0 - aL)^2*dfL)/(2*(fL - f0 + (a0 - aL)*dfL));
      a0Lhat = aL + tau*(aU - aL);
      if (a0hat < a0Lhat)
        a0hat = a0Lhat;
      end
      a0Uhat = aU - tau*(aU - aL);
      if (a0hat > a0Uhat)
        a0hat = a0Uhat;
      end
      a0 = a0hat;
    else
      gtemp = feval(gname,Am, rn, xk+a0*dk);
      df0 = gtemp'*dk;
      m = m + 1;
      % step 6
      if (((df0 < sigma*dfL) & (abs(fL - f0) > epsilon) ...
       & (m < mhat) & (dfL ~= df0)))
```

```
        deltaa0 = (a0 - aL)*df0/(dfL - df0);
        if (deltaa0 <= 0)
          a0hat = 2*a0;
        else
          a0hat = a0 + deltaa0;
        end
        a0Uhat = a0 + chi*(aU - a0);
        if (a0hat > a0Uhat)
          a0hat = a0Uhat;
        end
        aL = a0;
        a0 = a0hat;
        fL = f0;
        dfL = df0;
      else
        break;
      end
    end
 end % while 1
 if a0 < 1e-5,
    z = 1e-5;
 else
    z = a0;
 end

  function f = fx_R_LS(Am, rn,x0)
[n,m] = size(Am);
f = 0;
for i=1:m
    f = f + (norm(x0-Am(:,i))-rn(i))^2;
end
f = f/2;

function g = gx_R_LS(Am, rn,x0)
[n,m] = size(Am);
g = zeros(n,1);
for i =1:m
    g = g + (x0 - Am(:,i))*(1 - rn(i) / norm(x0 - Am(:,i)));
end

function [H1, tau] = Hx_R_LS(Am, rn,x0)
[n,m] = size(Am);
H1 = zeros(n,n);
tau = 0;
k = 0;
for i = 1:m
    k = x0 - Am(:,i);
    H1 = H1 + (rn(i) / norm(k)^3) * k * k';
    tau = tau + rn(i)/norm(k);
end
tau = m - tau;
```

A.2 Iterative Re-Weighting LS for Source Localization Using Range-Difference Measurements

A.2.1 IRWSRD-LS Algorithm

The followng m-files implement the iterative re-weighting range-difference based least squares (IRWSRD-LS) algorithm developed in Sec. 2.2 of the thesis.

Input parameters

Am $= [\boldsymbol{a}_1\ \boldsymbol{a}_2 \ldots \boldsymbol{a}_m]$, matrix of sensor locations, where $\boldsymbol{a}_i$ is a column vector that stores the coordinates of the ith sensor.

$\boldsymbol{d}_n = [d_1\ d_2 \ldots d_m]^T$, noisy range-difference measurements, where d_i is a measurement obtained from the pair of ith and the refference sensors.

Tunable parameters

$Kmax$, number of iterations for re-weighting procedure.

$epsi1$, convergence tolerance for iteratively-reweighted solution.

$epsi$, convergence tolerance for the bisection routine.

Otput parameters

$\boldsymbol{x}$, estimated location of the source.

```
function x = findroots_w_m(Am,dn,Kmax,epsi1,epsi)
xk = findroots(Am,dn,epsi);
C = [eye(2) zeros(2,1); zeros(1,2) -1];
m = size(Am,2);
err = 10;
ep = 1e-6;
k = 1;
while err >= epsi1 && k <= Kmax,
  nxk = norm(xk);
  for i = 1:m,
      ni = abs(norm(xk-Am(:,i))+nxk+dn(i));
      if ni >= ep,
         wk(i) = 1/ni;
      else
         wk(i) = 1/ep;
      end
  end
  wk = sqrt(m)*wk/norm(wk);
  Wk = diag(wk);
  B = Wk*[-2*Am' -2*dn];
  g = zeros(m,1);
  for i = 1:m,
      g(i) = wk(i)*(dn(i)^2 - norm(Am(:,i))^2);
  end
  btg = B'*g;
```

```
B1 = B'*B;
P1 = sqrtm(inv(B1));
W = P1*C*P1;
[V,D] = eig(W);
L = diag(D);
alp = sort(-1./L,'descend');
alp0 = alp(1);
alp1 = alp(2);
alp2 = alp(3);
aU = alp0;
aL = alp1;
dt = aU - aL;
while dt > epsi,
      lam = 0.5*(aL + aU);
      yt = inv(B1+lam*C)*btg;
      phit = yt'*C*yt;
      if phit > 0,
       aL = lam;
      else
         aU = lam;
      end
      dt = aU - aL;
end
lam = 0.5*(aL + aU);
yt = inv(B1+lam*C)*btg;
yn1 = yt(3);
if yn1 >= 0,
   xk_new = yt(1:2);
else
   P = P1*V;
   f = P'*btg;
   f1 = f(1); f2 = f(2); f3 = f(3);
   f1s = f1^2; f2s = f2^2; f3s = f3^2;
   d1 = D(1,1); d1s = d1^2;
   d2 = D(2,2); d2s = d2^2;
   d3 = D(3,3); d3s = d3^2;
   a0 = f1s*d1 + f2s*d2 + f3s*d3;
   a1 = 2*f1s*d1*(d2+d3) + 2*f2s*d2*(d1+d3) + 2*f3s*d3*(d1+d2);
   a2 = f1s*d1*(d2s+d3s+4*d2*d3) + f2s*d2*(d1s+d3s+4*d1*d3) ...
   + f3s*d3*(d1s+d2s+4*d1*d2);
   a3 = 2*f1s*d1*(d2s*d3+d3s*d2) + 2*f2s*d2*(d1s*d3+d3s*d1) ...
    + 2*f3s*d3*(d1s*d2+d2s*d1);
   a4 = f1s*d1*d2s*d3s + f2s*d2*d1s*d3s + f3s*d3*d1s*d2s;
   b_lu = [a4 a3 a2 a1 a0];
   rts = roots(b_lu);
   lamq = [];
   for i = 1:4,
       if imag(rts(i)) == 0,
          lamq = [lamq real(rts(i))];
       end
   end
   I02 = [];
   I1 = [];
   Lq = length(lamq);
   for i = 1:Lq,
       ti = lamq(i);
       if ti > alp0,
          I02 = [I02 ti];
```

```
        elseif ti < alp0 && ti > alp1,
           I1 = [I1 0];
        elseif ti < alp1 && ti > alp2,
           I02 = [I02 ti];
        end
    end
    L02 = length(I02);
    Yt2 = zeros(3,1);
    for i = 1:L02,
        yi = inv(B1+I02(i)*C)*btg;
        if yi(3) >= 0,
           Yt2 = [Yt2 yi];
        end
    end
    L2s = size(Yt2,2);
    obj2 = zeros(L2s,1);
    for i = 1:L2s,
        obj2(i) = norm(B*Yt2(:,i) - g);
    end
    [ymin2,ind2] = min(obj2);
    yt = Yt2(:,ind2);
    xk_new = yt(1:2);
  end
  err = norm(xk_new - xk);
  xk = xk_new;
  k = k + 1;
end
x = xk;
```

A.2.2 Hybrid IRWSRD-LS

The followng m-files implement the *hybrid* iterative re-weighting range-difference based least squares (IRWSRD-LS) algorithm developed in Sec. 2.2 of the thesis.

Input parameters

$fname$, name of the function to calculate the value of the LS objective at a point.
gname, name of the function to calculate the gradient of the LS objective at a point.
hname, name of the function to calculate the Hessian of the LS objective at a point.
$\boldsymbol{x}_0$, solution of the findroots_w_m.m as an initial point for the hybrid IRWSRD-LS algorithm.
p = $[m\ d_1\ d_2 \ldots d_m\ \boldsymbol{a}_1^T\ \boldsymbol{a}_2^T \ldots \boldsymbol{a}_m^T]^T$, a column vector of sensor and measurement data, where m is the number of sensors, d_i is a noisy range-difference measurement obtained from the pair of ith and the refference sensors, and $\boldsymbol{a}_i$ is a column vector that stores the coordinates of the ith sensor.

Tunable parameters

opt, a parameter allowing the user to choose the implementation of the inexact line

search, *opt* $= 1$ toggles the algorithm to use inexact line search by backtracking, *opt* $= 2$ toggles the algorithm to use Fletcher's inexact line search.
epsi, convergence tolerance.

Otput parameters

$\boldsymbol{x}_s$, estimated location of the source.
f_s, value of the LS objective at the estimated soluion point.
k, number of iterations it took the algorithm to converge to the solution point with the convergence tolerance *epsi*.

Calling sequence

Function newton_RD.m calls inex_lsearch.m or bt_lsearch.m to perform the line search, Hx_RD_LS.m to evaluate the Hessian of the LS objective at a point, fx_RD_LS.m and gx_RD_LS.m to evaluate the objective and the gradient of the LS objective at a point respectively.
Inexact line search by backtracking bt_lsearch.m and Fletcher's inexact line search inex_lsearch.m are not included in listing, as the implementation is not original.

```
function [xs,fs,k] = newton_RD(fname,gname,hname,x0,p,epsi,opt)
k = 1;
xk = x0;
gk = feval(gname,xk,p);
Hk = feval(hname,xk,p);
dk = -inv(Hk)*gk;
if opt == 1
    ak = bt_lsearch(xk,dk,fname,gname,p);
elseif opt == 2
    ak = inex_lsearch(xk,dk,fname,gname,p);
end
adk = ak*dk;
gn = norm(gk);
while gn > epsi,
  xk = xk + adk;
  gk = feval(gname,xk,p);
  Hk = feval(hname,xk,p);
  dk = -inv(Hk)*gk;
    if opt == 1
        ak = bt_lsearch(xk,dk,fname,gname,p);
    elseif opt == 2
        ak = inex_lsearch(xk,dk,fname,gname,p);
    end
  adk = ak*dk;
  gn = norm(gk);
  k = k + 1;
end
xs = xk + adk;
fs = feval(fname,xs,p);

function f = fx_RD_LS(x,p)
```

```
p = p(:);
m = p(1);
dn = p(2:(m+1));
am = p(m+2:end);
Am = vec2mat(am,2)';
f = 0;
for i = 1:m,
    si = Am(:,i);
    fi = norm(x-si)- norm(x) - dn(i);
    f = f + fi^2;
end
f = 0.5*f;

function g = gx_RD_LS(x,p)
p = p(:);
m = p(1);
dn = p(2:(m+1));
am = p(m+2:end);
Am = vec2mat(am,2)';
nx = norm(x);
g = [0 0]';
for i = 1:m,
    si = Am(:,i);
    ri = dn(i);
    ni = norm(x-si);
    if nx > 0,
       xt = x/nx;
       if ni > 0,
          qi = (x - si)/ni;
          ci = ni - nx - ri;
          gi = qi - xt;
       elseif ni == 0,
          ci = -nx - ri;
          gi = -xt;
       end
    elseif nx == 0,
       if ni > 0,
          qi = (x - si)/ni;
          ci = ni - ri;
          gi = qi;
       elseif ni == 0,
          ci = -ri;
          gi = zeros(2,1);
       end
    end
    g = g + ci*gi;
end

function H = Hx_RD_LS(x,p)
p = p(:);
m = p(1);
dn = p(2:(m+1));
am = p(m+2:end);
Am = vec2mat(am,2)';
Hw = zeros(2,2);
nx = norm(x);
I = eye(2,2);
for i = 1:m,
```

```
    si = Am(:,i);
    ri = dn(i);
    ni = norm(x-si);
    ci = ni - nx - ri;
    if nx > 0,
       xt = x/nx;
       H2 = (I - xt*xt')/nx;
       if ni > 0,
          qi = (x - si)/ni;
          hi = (qi - xt);
          H3i = (I - qi*qi')/ni;
          Hi = hi*hi' + ci*(H3i - H2);
       elseif ni == 0,
          Hi = xt*xt' - ci*H2 ;
       end
    elseif nx == 0,
       if ni > 0,
          qi = (x - si)/ni;
          H3i = (I - qi*qi')/ni;
          Hi = qi*qi' + ci*H3i;
       elseif ni == 0,
          Hi = zeros(2,2);
       end
    end
    Hw = Hw + Hi;
end
[V,D] = eig(Hw);
d = diag(D);
dm = max(d,5e-2);
H = V*diag(dm)*V';
```

A.3 PCCP-Based LS Algorithm

The followng m-files implement the constrained penalty convex-concave procedure for solving the localization problem based on range measurements. The implementation follow the alrorithm developed in Chapter 3.

Input parameters

Am $= [\boldsymbol{a}_1\ \boldsymbol{a}_2 \ldots \boldsymbol{a}_m]$, matrix of sensor locations, where $\boldsymbol{a}_i$ is a column vector that stores the coordinates of the ith sensor.

$\boldsymbol{r}_n = [r_1\ r_2 \ldots r_m]^T$, noisy range measurements, where r_i is a measurement obtained from the ith sensor.

Tunable parameters

gam a parameter that controls the bounds in the constraints.

$\boldsymbol{x}_0$, initial point for the source location.

K, number of iterations for re-weighting procedure.

Otput parameters

$\boldsymbol{x}_w$, estimated location of the source.

```
function [x_w] = pccp(Am,rn,gam,x0,K)
[n,m] = size(Am);
k = 0;
xk = x0;
rnb_p = rn + gam*sig;
rnb_n = rn - gam*sig;
ab = (mean(Am'))';
tau = 1;
tau_max = 100000;
mi = 1/m;
m2 = 2*m;
nn = 1000;
while (k < K)
    cvx_begin quiet
       variable x(n)
       variable s(m2);
       v = ab;
       for i = 1:m,
           xai = xk - Am(:,i);
           v = v + (mi*rn(i)/norm(xai))*xai;
       end
       minimize(x'*x - 2*x'*v + tau*sum(s));
       subject to
       for i = 1:m,
           norm(x-Am(:,i)) <= rnb_p(i) + s(i);
           xai = xk - Am(:,i);
           ni = norm(xai);
           xain = xai/ni;
           -xain'*(x-xk) - ni + rnb_n(i) <= s(m+i);
       end
       s >= 0;
    cvx_end
    nn = norm(xk-x);
    xk = x;
    tau = min(1.5*tau,tau_max);
    k = k + 1;
end
xk_w = xk;
```

A.4 SCR-RDLS Algorithm

The followng m-files implement the sequential convex relaxation for solving the localization problem based on range-difference measurements. The implementation follow the alrorithm developed in Sec.4.1.

Input parameters

Am $= [\boldsymbol{a}_1\ \boldsymbol{a}_2 \ldots \boldsymbol{a}_m]$, matrix of sensor locations, where $\boldsymbol{a}_i$ is a column vector that stores the coordinates of the ith sensor.

$\boldsymbol{d}_n = [d_1\ d_2 \ldots d_m]^T$, noisy range-difference measurements, where d_i is a measurement obtained from the pair of ith and the refference sensors.

Tunable parameters

bt, bound for increments of design variables.

epsi, convergence tolerance.

Otput parameters

$\boldsymbol{x}_w$, estimated location of the source.

```
function x_w = scr_rdls(Am,dn,bt,epsi)
m = length(dn);
em = ones(m,1);
zm = zeros(m,1);
Im = eye(m);
cvx_begin quiet
   variable x0(2);
   variable y(1);
   variable z(m);
   minimize((z-y*em-dn)'*(z-y*em-dn));
   subject to
   norm(x0) <= y;
   for i = 1:m,
       norm(x0-Am(:,i)) <= z(i);
   end
cvx_end
xk = x0;
yk = norm(xk);
zk = zeros(m,1);
for i = 1:m,
    zk(i) = norm(xk-Am(:,i));
end
mu = 20;
P = zeros(m+1,m+3);
q = zeros(m+1,1);
k = 0;
err = 1;
Kmax = 100;
while (err >= epsi && k <= Kmax),
    for i = 1:m,
```

```
        vi = xk - Am(:,i);
        nvi = norm(vi);
        q(i) = zk(i) - nvi;
        if nvi < 1e-6,
            gvi = zeros(2,1);
        else
            gvi = vi/nvi;
        end
        P(i,:) = [gvi', 0, -Im(i,:)];
    end
    nx = norm(xk);
    if nx < 1e-6,
       P(m+1,:) = [0 0 -1 zm'];
    else
       P(m+1,:) = [xk'/nx, -1, zm'];
    end
    q(m+1) = yk - nx;
    cvx_begin quiet
       variable dx(2,1)
       variable dy(1,1)
       variable dz(m,1)
       variable u(m+1,1)
       variable v(m+1,1)
       variable w(1,1)
       dtt = [dx; dy; dz];
       pk = dn - zk + yk*em;
       minimize(norm(dz - dy*em - pk) + mu*(sum(u+v)+sum(w)));
       subject to
       P*dtt - q == u - v;
       abs(dtt) <= bt + w;
       u >= 0;
       v >= 0;
       w >= 0;
       mu = min(1.5*mu,1e4);
     cvx_end
     err = norm(dtt);
     xk = xk + dx;
     yk = yk + dy;
     zk = zk + dz;
     k = k + 1;
     % current_status = [k err norm(u) norm(v) norm(w)]
end
x_w = xk;
```

A.5 SCR-RLS Algorithm

The followng m-files implement the sequential convex relaxation for solving the localization problem based on range measurements. The implementation follow the alrorithm developed in Sec. 4.2.

Input parameters

$\mathbf{Am} = [\boldsymbol{a}_1 \ \boldsymbol{a}_2 \ldots \boldsymbol{a}_m]$, matrix of sensor locations, where $\boldsymbol{a}_i$ is a column vector that stores the coordinates of the ith sensor.

$\boldsymbol{r}_n = [r_1 \ r_2 \ldots r_m]^T$, noisy range measurements, where r_i is a measurement obtained from the ith sensor.

Tunable parameters

gamma a parameter that controls the size of the convex hull that defines a feasibility region of the proble.

$\boldsymbol{x}_0$, initial point for the source location.

K, number of iterations for re-weighting procedure.

Otput parameters

$\boldsymbol{x}_w$, estimated location of the source.

```
function x_w = scr_rls(Am,rn,x0,gamma,K)
m = length(rn);
xk = x0(:);
rn = rn(:);
k = 0;
g0 = 1-gamma;
% quadratically updates the value of \gamma
a = (1-g0)/(K-1)^2;
% epsi = 1e-8;
while k < K,
    gk = 1 - a*(k-K+1)^2;
    cvx_begin quiet
      variable x(2,1)
      variable z(m,1)
      minimize(norm(z-rn))
      subject to
      for i = 1:m,
          ai = Am(:,i);
% the arc portion of the convex region
% is updated as iteration proceeds.
          norm(x-ai) <= (2-gk)*z(i);
          xai = (xk-ai)/norm(xk-ai);
% the line portion of the convex region
% is pushed towards the arc portion.
          xai'*(x-ai) >= gk*z(i);
      end
    cvx_end
    xk = x;
    k = k + 1;
end
x_w = xk;
```

Bibliography

[1] A. Antoniou and W.-S. Lu, *Practical Optimization: Algorithms and Engineering Applications*, Springer, New-York, 2007.

[2] A. Beck, "On the convergence of alternating minimization for convex programming with applications to iteratively re-weighted least squares and decomposition schemes," *SIAM J. Optim.*, vol. 25, no. 1, pp. 185-209, Jan. 2015.

[3] A. Beck, and D. Pan, "On the solution of the GPS localization and circle fitting problems," *SIAM J. Optim.*, vol. 22, no. 1, pp. 108–134, Jan. 2012.

[4] A. Beck, P. Stoica and J. Li, "Exact and approximate solutions of source localization problems," *IEEE Trans. on Signal Proc.*, vol. 56, no. 5, pp. 1770–1777, May 2008.

[5] A. Beck, M. Teboulle, and Z. Chikishev, "Iterative minimization schemes for solving the single source localization problem," *SIAM J. Optim.*, vol. 19, no. 3, pp. 1397–1416, Nov. 2008.

[6] N. Bissantz, L. Dümbgen, A. Munk, and B. Stratmann, "Convergence analysis of generalized iteratively reweighted least squares algorithms on convex function spaces," *SIAM J. Optim.*, vol. 19, no. 4, pp 1828–1845, 2009.

[7] A.-C. Chang and J.-C. Chang, "Robust mobile location estimation using hybrid TOA/AOA measurements in cellular systems," *Wireless Pers. Commun.*, vol. 65, pp 1–13, 2012. doi:10.1007/s11277-011-0224-8

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

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

[10] K.W. Cheung, W.K. Ma, and H.C. So, "Accurate approximation algorithm for TOA-based maximum-likelihood mobile location using semidefinite programming," in *Proc. ICASSP*, vol. 2, pp. 145–148, 2004.

[11] K. W. Cheung and H. C. So, "A multidimensional scaling framework for mobile location using time-of-arrival measurements," *IEEE Trans. on Signal Proc.*, vol. 53, no. 2, pp. 460–470, Feb. 2005.

[12] K. W. Cheung, H. C. So, W. K. Ma, and Y. T. Chan, "Least squares algorithms for time-of-arrival-based mobile location," *IEEE Trans. on Signal Proc.*, vol. 52, no. 4, pp. 1121–1228, Apr. 2004.

[13] Cisco Systems, Inc., http://www.cisco.com/c/en/us/products/wireless/wireless-location-appliance, 2016.

[14] J.A. Costa, N. Patwari, and A. O. Hero, "Distributed weighted-multidimentional scaling for node localization in sensor networks,"ACD Trans. Sens. Netw., vol. 2, no. 1, pp. 39–64, 2006.

[15] I. Cotanis, *"Aspects of Geo-Location in Mobile Networks,"* Ascom, 2010.

[16] CVX Research, http://cvx.com/cvx, August 2012.

[17] M. R. Gholami, S. Gezici, and E. G. Ström, "TW-TOA based positioning in the presence of clock imperfections," *Digital Signal Processing*, vol. 59, pp. 19–30, 2016.

[18] M. R. Gholami, E. G. Ström, and M. Rydström, "Indoor sensor node positioning using UWB range measurements," in *17th European Signal Processing Conference (Eusipco)*, pp. 1943–1947, 2009.

[19] M. R. Gholami, E. G. Ström, F. Sottile, D. Dardari, A. Conti, S. Gezici, M. Rydström, and M. A. Spirito, "Static positioning using UWB range measurements," in *Future Network and Mobile Summit Conference Proceedings*, pp. 1–10, 2010.

[20] D. Dardari, A.Conti, U.J. Ferner, A. Giorgetti, and M.Z. Win, "Ranging with ultrawide bandwidth signals in multipath environments," *Proceedings of the IEEE*, vol. 97, pp. 427–450, 2009.

[21] I. Daubechies, R. DeVore, M. Fornasier, and C. S. Güntürk, "Iteratively reweighted least squares minimization for sparse recovery," *Comm. Pure Appl. Math.*, vol. 63, pp. 1–38, 2010.

[22] A.G. Dempster, "Dilution of precision in angle-of-arrival positioning systems," *Electron. Lett.*, vol. 42, no. 5, pp. 291–292, 2006.

[23] H. F. Durrant-Whyte, "Uncertain geometry in robotics," *IEEE J. Robot. Autom.*, vol. 4, no. 1, pp. 23–31, 1988. doi:10.1109/56.768. isnumber=50

[24] Fitbit, Inc., https://www.fitbit.com, 2016.

[25] C. Fortin and H. Wolkowicz, "The trust region subproblem and semidefinite programming," *Optim. Methods Softw.*, vol. 19, no.1, pp. 41–67, 2004.

[26] Garmin, Inc., https://www.garmin.com, 2016.

[27] C. Gentile, N. Alsindi, R. Raulefs, and C. Teolis, *Geolocation Techniques: Principles and Applications*, Springer, New-York, 2013.

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

[29] I. Guvenc and C.-C Chong, "A survey on TOA based wireless localization and NLOS mitigation techniques", *IEEE Commun. Surv. Tutor.*, vol. 11, no. 3, 3rd Quarter, 2009.

[30] P. Hartman, "On functions representable as a difference of convex functions," *Pacific Journal of Math*, vol. 9, no. 3, pp. 707–713, 1959.

[31] Y. Huang, J. Benesty, G. W. Elko, and R. M. Mersereau, "Realtime passive source localization: A practical linear correction least-squares approach," *IEEE Trans. on Speech Audio Proc.*, vol. 9, no. 8, pp. 943-956, Nov. 2002.

[32] D. Ismailova and W.-S. Lu,"Improved least-squares methods for source localization: an iterative re-weighting approach," *Proc. IEEE International Conference on Digital Signal Processing (DSP)*, pp. 665–669, 2015.

[33] D. Ismailova and W.-S. Lu, "Penalty Convex-Concave Procedure of Source Localization Problem," *IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, pp 1084–1088, 2016.

[34] Y. Kang, Z. Zhang, and W.-J. Li, "On the global convergence of majorization minimization algorithms for nonconvex optimization problems," in *arXiv: 1504.07791v2*, 2015.

[35] W. Kim, J. Lee, and G. Jee, "The interior-point method for an optimal treatment of bias in trilateration location," *IEEE Trans. Veh. Technol.*, vol. 55, no. 4, pp. 1291–1301, Jul. 2006.

[36] G. R. Lanckreit and B. K. Sriperumbudur, "On the convergence of the concave-convex procedure," in *Advances in Neural Information Proc. Systems*, pp. 1759–1767, 2009.

[37] D. Li and H. Hu, "Energy-Based Collaborative Source Localization Using Acoustic Microsensor Array," in *EURASIP Journal on Applied Signal Proc.*, vol. 4, 321–337, 2003.

[38] H. Liu, H. Darabi, P. B, and J. Liu, "Survey of Wireless Indoor Positioning Techniques and Systems," *IEEE Trans. on Systems, Man and Cybernetics. Part C: Applications and Revies*, vol. 37, no. 6, pp. 1067–1080, Nov. 2007.

[39] Y. Liu, Y.H. Hu, and Q. Pan, "Distributed, Robust Acoustic Source Localization in a Wireless Sensor Network," *IEEE Trans. on Signal Proc.*, vol. 60, no. 8, pp. 4350–4359, Aug. 2012.

[40] T. Lipp and S. Boyd, "Variations and extensions of the convex-concave procedure," *Research Report*, Stanford University, Aug. 2014.

[41] W.-S. Lu, *"Engineering Design by Optimization II"*, ELEC 573/603 Course Notes, University of Victoria, 2014.

[42] L. Lu, H.-C. Wu, K.Yan, and S.Iyengar, "Robust expectation maximization algorithm for multiple wideband acoustic source localization in the presence of nonuniform noise variances," *IEEE Sensors J.*, vol. 11, no. 3, pp. 536–544, Mar. 2011.

[43] The Mathworks, Inc., http://mathworks.com, 2015.

[44] G. Mao and B. Fidan, *Localization Algorithms and Strategies for Wireless Sensor Networks: Monitoring and Surveillance Techniques for Target Tracking*, Hershey, PA: IGI Global, New-York, 2009.

[45] J.J. More, "Generalizations of the trust region subproblem," *Optim. Methods Softw.*, vol. 2, pp. 189–209, 1993.

[46] Y. Nesterov, *Introductory Lectures on Convex Optimization*, Kluwer Academic Publishers, Boston, 2004.

[47] D.P. O'Leary, "Robust regression computation using iteratively reweighted least squares," *SIAM J. Matrix Anal. Appl.*, vol. 11, no. 3, pp. 466–480, 1990.

[48] K. Pahlavan Alavi, "Modeling of the TOA-based distance measurement error using UWB indoor radio measurements," *IEEE Commun. Lett.*, vol. 10, no. 4, pp. 275–277, 2006.

[49] T. Qiao, S. Redfield, A. Abbasi, Z. Su, and H. Liu, "Robust coarse position estimation for TDOA localization," *IEEE Wireless Commun. Lett.*, vol. 2, no. 6, pp. 623–626, Dec. 2013.

[50] S. Qin, Q. Wan, and L.-F. Duan, "Fast and efficient multidimentional scaling algorithm for mobile positioning," *IET Signal Processing*, vol. 6, no. 9, pp. 857–861, March 2012.

[51] R.E. Richton and G.M. Djuknic, "Geo-location and assisted GPS", *IEEE Comput.*, vol. 34, no. 2, pp. 123–125, 2001.

[52] Z. M. Saric, D. D. Kukolj, and N. D. Teslic, "Acoustic source localization in wireless sensor network", *Circuits Syst Signal Proc.*, vol. 29, pp. 837–856, 2010.

[53] A. H. Sayed, A. Tarighat, and N. Khajehnouri, "Network-based wireless location," *IEEE Signal Pro. Mag.*, vol. 22, no. 4, pp. 24–40, July 2005.

[54] H. Schau and A. Robinson, "Passive source localization employing intersecting spherical surfaces from time-of-arrival differences," *IEEE Trans. on Acoustic, Speech Signal Proc.*, vol. ASSP–35, pp. 1223–1225, Aug. 1987.

[55] X. Sheng and Y.-H. Hu, "ML Multiple-source localization using acoustic energy measurements with wireless sensor networks," *IEEE Trans. on Signal Proc.*, vol. 53, no.1, pp. 44–53, Jan. 2005.

[56] Skyhook Wireless Inc., http://www.skyhookwireless.com, 2016.

[57] J. O. Smith and J. S. Abel, "Closed-form least-squares source location estimation from range-difference measurements," *IEEE Trans. on Acoustic, Speech Signal Proc.*, vol. 12, pp. 1661–1669, Dec. 1987.

[58] H.C. So and F.K.W. Chan, "Generalized Subspace Approach for Mobile Positioning With Time-of-Arrival Measurements," *IEEE Trans. on Signal Proc.*, vol. 55, no. 10, pp. 5103–5107, October 2007.

[59] M. A. Sprito, "On the accuracy of cellular mobile station location estimation," *IEEE Trans. on Veh. Technol.*, vol. 50, pp. 674-685, May 2001.

[60] P. Stoica and J. Li, "Source localization from range-difference measurements," *IEEE Signal Proc. Mag.*, vol. 23, pp. 63–65,69, Nov. 2006.

[61] L. Vandenberghe and S. Boyd, "Semidefinite programming," *SIAM Rev.*, vol. 38, no. 1, pp. 40–95, Mar. 1996.

[62] A. L. Yuille and A. Rangarajan, "The concave-convex procedure,' *Neural Computation*, vol. 15, no. 4, pp. 915–936, 2003.

[63] K. Yao, R. Hudson, C. Reed, D. Chen, and F. Lorenzelli, "Blind beamforming on a randomly distributed sensor array system," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 1555-1567, Oct. 1998.