THE HONG KONG POLYTECHNIC UNIVERSITY
香港理工大學
Pao Yue-kong Library
包玉剛圖書館

Copyright Undertaking

This thesis is protected by copyright, with all rights reserved.

By reading and using the thesis, the reader understands and agrees to the following terms:

1. The reader will abide by the rules and legal ordinances governing copyright regarding the use of the thesis.
2. The reader will use the thesis for the purpose of research or private study only and not for distribution or further reproduction or any other purpose.
3. The reader agrees to indemnify and hold the University harmless from and against any loss, damage, cost, liability or expenses arising from copyright infringement or unauthorized usage.

IMPORTANT

If you have reasons to believe that any materials in this thesis are deemed not suitable to be distributed in this form, or a copyright owner having difficulty with the material being included in our database, please contact lbsys@polyu.edu.hk providing details. The Library will look into your claim and consider taking remedial action upon receipt of the written requests.

Pao Yue-kong Library, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong

http://www.lib.polyu.edu.hk

This thesis in electronic version is provided to the Library by the author. In the case where its contents is different from the printed version, the printed version shall prevail.

The Hong Kong Polytechnic University
Department of Applied Mathematics

Optimal Design of Distributed Microphone Array

Mingjie GAO

A thesis submitted in partial fulfilment of
the requirements for the degree of Doctor of Philosophy

June 2013

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgment has been made in the text.

Mingjie GAO

Abstract

This thesis concentrates on the study of distributed broadband beamforming system and source localization problem.

The main contributions of this thesis consist of the following four parts.

1. For the design of distributed broadband beamforming system, each microphone is equipped with wireless communications capability. It is designed such that the error between the actual response and the desired response is minimized which is then formulated as a minimax optimization problem. Since we find that the performance of the optimized designs is very sensitive to the perturbations in microphone locations, we first use sensor network technology which solved by semi-definite programming method to estimate the microphone locations, and then incorporate it into the design process. We propose a suitable robust formulation as a remedy to regain the performance. The minimax optimization problem is transformed into a semi-definite programming problem so that interior point algorithms can be applied. We illustrate the proposed method by several designs and demonstrate that this approach is essential to regain accuracy in the optimized designs.

2. The broadband beamforming design problem is formulated as a non-strictly convex semi-infinite programming problem. The approach to solve it is that adding a small perturbation quadratic function to the objective function to make it strictly convex. We demonstrate that the solution of the per-

turbation semi-infinite programming problem approximates the solution of the original problem as the perturbation going to 0. The new exchange algorithm is applied successfully to the filter design problem.

3. We present a new method to solve the source localization problem with time-difference information. Fist we formulate a mixed SDP-SOCP relaxation model and then state how to obtain the exact solution from the solutions of the mixed SDP-SOCP relaxation model and the second order polynomial equation. The estimator properties for the true source location under noises is proposed. We also give bi-level method to solve the source localization problem that formulated only as a semi-definite programming. Then a mixed SDP-SOCP relaxation model for source localization combined with sensor network localization problem is studied, also we give some statistical analyses for it. Many illustrated examples demonstrate those approaches can be applied successfully and some comparisons are presented.

4. We obtain a representation for the solution of the mixed SDP-SOCP model and the characterization such that the mixed SDP-SOCP model has an exact relaxation in two-dimensional case. We derive the geometry of the localizable region for the proposed mixed model. The characterization shows that the source localization with some time-difference information can be solved exactly by the mixed SDP-SOCP relaxation model in a larger region than the triangle region determined by three points.

In my PhD study period at Department of Applied Mathematics, The Hong Kong Polytechnic University, the following papers are written:

- K. F. C. Yiu, M. J. Gao and Z. G. Feng, Design of distributed beamforming system using semi-definite programming, International Journal of Innovative Computing, Information and Control, 8(5B), 2012, pp.3755-3768.

- K. F. C. Yiu, M. J. Gao, T. J. Shiu, S. Y. Wu, T. Tran and I. Claesson, A fast algorithm for the optimal design of high accuracy windows in signal processing, Optimization Methods and Software, iFirst, 2012, pp.1-17.

To be submitted

- M. J. Gao, K. F. C. Yiu and S. Y. Wu, Multiple exchange algorithm for the design of distributed beamforming system.

- M. J. Gao, K. F. C. Yiu and Y. Y. Ye, SDP-SOCP relaxation for source localization with TDOA information.

- M. J. Gao, K. F. C. Yiu and Y. Y. Ye, SDP-SOCP relaxation for source localization combined with sensor network localization problem.

Acknowledgments

I wish to express my sincere appreciation to my chief supervisor Dr. Cedric for his help throughout my study at the Hong Kong Polytechnic University. He has given me many distinctive ideas, careful guidance and valuable suggestions to this thesis. The seminars in almost every week stimulate my research interest and provide various chances for me to communicate, also give me a harmonious relationship with other people in our research group.

I would also like to thank Professor Yinyu Ye (Stanford University) for his ideas on source localization part in my thesis during his visit to the Hong Kong Polytechnic University. Moreover, I would like to thank Professor Soon-Yi Wu (National Cheng Kung University) for his suggestions on the semi-infinite programming part in my thesis. Without their help, this thesis would not have been possible.

I wish to thank Dr. Zhiguo Feng for his helpful discussion and encouragement. Thanks to Dr. Xinqiu Zhao for her attention on my research and daily life. Also I wish to thank Professor Xiaojun Chen for her many especially concerns.

To Dr. Yanfang Zhang, Dr. Jinzhen Liu, Mr Zhibao Li, Miss Yan Wei and Miss Cong Xie, I am grateful for their friendship and help to my life in Hong Kong. I own multiple thanks to my good friends: Zhangyou Chen, Chen Ma, Yi Xu, Chuanxin Bian, Yaohua Hu, Shirong Deng, Jing Li and Hongjun Zhou for their spiritual support. The help from the members in DE 405 and DE 407 (Department of Applied Mathematics) are deeply acknowledged.

I gratefully acknowledge The Hong Kong Polytechnic University for the financial support which enabled me to finish my PhD program more easily. I also express my gratitude to the supporting staffs in Department of Applied Mathematics for their kind help.

Last but far from least, I would like to thank my four parents and my husband, who not only provide me extra financial support but also the encouragement and guidance on my PhD study in Hong Kong.

Contents

List of Figures

Chapter 1

Introduction

1.1 Background

Current advancement of wireless communications has made its deployment in wider perspective. This facilitates the development of distributed systems, such as a microphone network, which overcomes some of the technical problems from a wired system by providing greater freedom of movements to the speaker and avoidance of cabling problems common with wired microphones caused by constant moving and stressing the cables. There are numerous applications that can be built on a microphone network. Speech is the preferred natural interface for controlling equipment in households or factories. However, signal degradation poses a serious problem in many environments, which affects the accuracy of the speech recognition and voice control system. As a result, beamforming techniques, a signal processing technique used in sensor arrays for directional signal transmission or reception, are required to enhance the received signals.

Broadband beamformers (Veen and Buckley [1988] Trees [2002] Khosravy et al. [2009] Zhang et al. [2010b]) have been studied extensively due to their wide applications in many areas such as radar, sonar, wireless communications, biomedicine, speech and acoustics. When microphone arrays are deployed, many beamforming algorithms exist (for example, Hoshuyama et al. [1999] Gannot et al. [2001] Chan and Chen [2007]) to reduce the level of localized and ambient noise signals from the desired direction via spatial filtering, which plays an important role in noise reduction and speech enhancement. For many applications, such as video conference and mobile telephony, the speaker doesn't stand very far from the array. There are various algorithms dedicated for the design of this kind of beamformer in the literature. In (Kennedy et al. [1998]), the near-field-far-field reciprocity relationship is derived and applied to design near-field beamformers via far-field design techniques. An interesting approach is presented in (Ryan and Goubran [2000]). It makes use of a signal propagation vector representing an ideal point source of acoustic radiation. When the desired frequency response is known, multidimensional filter design techniques can be applied. In (Nordebo et al. [1994]), the minimax problem is formulated as a quadratic programming problem and the SQP method is applied. A penalty function method is developed in (Nordholm et al. [1998]) to formulate the problem as an unconstrained nonlinear optimization problem. This method is modified in (Lau et al. [1999]) by replacing the penalty function with a root-catching method. In (Yiu et al. [2003]), the l_1 norm measure and the real rotation theorem are applied to formulate the problem as a semi-infinite linear programming problem.

For many applications, the design problem can be formulated as a minimax optimization problem. Similar to many filter design problems (Dam et al. [2000] Yu et al. [2005] Lee et al. [2006]), large-scale linear programming techniques (for example, Lim and Lian [1993] Choi and Lim [2008]) are often used. When the problem size increases as a result of an increase in the number of filters as well as the filter lengths, or a refinement in the discretization of the frequency-space domain, the number of constraints will be very large and these problems will be very expensive to solve if the methods above are applied. Hence, an efficient algorithm is necessary.

Semi-definite programming (SDP) is a generalization of linear programming (LP) where the decision variables are arranged in a symmetric matrix instead of a vector, and the non-negative orthant is replaced by the cone of positive semidefinite matrices (Wolkowicz et al. [2003] Li et al. [2003] Huang et al. [2003]). Since interior point algorithms can be employed, semi-definite programming has polynomial time computational complexity and can be solved efficiently. It has also been successfully applied to many signal processing problems, such as frequency response masking filter design (Lu and Hinamoto [2003]), antenna design (Wang et al. [2003]), filter bank design (Kha et al. [2007]), and sensor network (Biswas and Ye [2004]), achieving very good performance. Hence, in this thesis, first we will apply the SDP method for solving the broadband beamformer design problem.

Semi-infinite programming (SIP) is an optimization problem with a finite number of variables and an infinite number of constraints (Goberna and López [1998; 2002] Hettich and Kortanek [1993] Reemsten and Górner [1998]). It has many applications in approximation theory (Glashoff and Roleff [1981]), optimal control (Liu et al. [2001]), and engineering problems such as optimum filter design in signal processing (Potchinkov and Reemsten [1995]). The difficulty for solving the SIP is that it has infinite number of constraints. There are two common method to solve it, one is the discretization method (Hettich Still [2001] Teo et al. [2000]), the other is the reduction based method (Hettich and Kortanek [1993] Reemsten and Górner [1998]). For discretization methods, they are computationally costly when the number of discretization is very large. On the other hand, the reduction method may need strong assumptions. However, another families called exchange methods are also very important and usually be used (Cheney [1982] Laurent and Carasso [1978] Hettich and Gramlich [1990] Tichatschke and Nebeling [1988] Kortanek and No [1993] Betró [2004]). Recently, Zhang, Wu and López (Zhang et al. [2010a]) proposed a new exchange method for the convex SIP problem (P) whose main feature is that only those active constraints with positive Lagrange multipliers are kept, and no global optimization needs to be carried out at each iteration to detect the (almost) most violated constraint. The algorithm associated with the method terminates in a finite iterations needs to be under the condition that $x \to f(x)$ or $x \to g(x, s)$ is strictly convex. For solving the non-strictly convex cases, the idea is that we add a small perturbation to the objective function in order to let it be strictly convex. Thus in this thesis, we will propose a perturbation method of exchange algorithm for solving the general convex semi-infinite programming (CSIP) problems. We will show that under some assumptions, an approximation solution of original problem can be obtained in a finite of iterations by the perturbation exchange algorithm. Then we will use this exchange method to solve the broadband beamformer design problem.

One of the assumptions in applying the aforementioned design techniques is that the locations of the microphones are required to be measured exactly. In practice, the microphones could be scattered around and could even be moving around occasionally. If very precise measurements are needed every time, this will make the design process very tedious and repetitive. If wireless microphones are deployed instead, we just need to make use of the accurate positions of a few anchor nodes in the network together the pairwise distance measurements between any two nodes to estimate the locations of the wireless microphones. Since the distance measurements always contain noise and the effect of the measurement uncertainty usually depends on the geometrical relationship between sensors which is not known *a priori*, optimization techniques are often deployed to find the best estimates. Here, we also adopt the SDP method for solving the problem (Biswas et al. [2006a] So and Ye [2007]). The basic idea behind the technique is to convert the nonconvex quadratic distance constraints into linear constraints by introducing a relaxation to remove the quadratic term in the formulation. The performance of this technique is highly satisfactory compared to other techniques (Biswas et al. [2006b] Wang et al. [2008a]). Very few anchor nodes are required to accurately estimate the position of all the unknown nodes in a network. Also the estimation errors are minimal even when the anchor nodes are placed arbitrarily within the network.

Owing to perturbations in the estimated sensor locations, a robust formulation is required to allow for certain amount of errors. In fact, the designed beamformers turn out to be very sensitive to errors in the microphone locations. In this thesis, the sensor network technology will be employed and incorporated into broadband beamforming design. In particular, an appropriate robust formulation is proposed to give more flexibility in the designs. We will demonstrate by examples that the proposed robust approach is essential to regain the accuracy in the designs if microphone locations are indeed erroneous.

Time difference of arrival estimation (TDOA) between signals received at two microphones has been considered to be a useful parameter for a variety of applications such as acoustic source localization, speech recognition and radar communication. There are various techniques that can be used to compute the TDOA. The most basic method to solve this problem is cross-correlation (CC) method (Carter [1993]) which first cross-correlates the two received signals and then consider the maximum peak in the output as the estimation. To improve the the accuracy and moderate computational complexity of the result, generalized cross-correlation method which is to find the maximum peak in the output of cross correlation between the filtered versions of two received signals is proposed by Knapp and Carter in 1976 (Knapp and Carter [1976]). To choose the weighting function, there are two common methods. One is the phase transform (PHAT) and the other is maximum likelihood estimator (ML). Also there are some other algorithms such as average square difference function (ASDF), adaptive algorithms, MUSIC (Chu and Mitra [1999]), ESPRIT (Jakobsson et al. [1998]) and wavelets (Barsanti and Tummala [2003]).

Acoustic source localization, aiming to locate the sound source given some measurements, remains to be an important problem in the signal processing literature owing to their importance to many applications including radar, sonar, teleconferencing, wireless communications and voice control. It can be solved based on TDOA, or angle-of-arrival (AOA) measurements, or a combination of them. TDOA information has been stated on the above. In the AOA approach, each sensor node is equipped with an antenna array which can be costly, thus using AOA to estimate the source location is less practical. Typically, the source location is estimated by two stages. In the first stage, the TDOA between each pair of microphones is estimated and then transformed into distance difference measurements between sensors, resulting in a set of nonlinear hyperbolic equations. In the next stage, efficient algorithms are needed to find the intersection of these nonlinear hyperbolic equations and obtain an estimate to the location of the source. As a matter of fact, finding the intersection of hyperbolic equations is a highly nonlinear problem. For many years, numerous iterative algorithms have been proposed for solving this problem, including the maximum likelihood estimation method (Hahn and Tretter [1973]) and the constrained optimization method (Yang et al. [2010]). Usually linear approximation and numerical techniques are used in these approaches. However, finding a good initial point to avoid local minima is a difficult task, and therefore the convergence can not be guaranteed. To avoid the reliance on a good initial solution guess, estimators with some kind of closed-form formulas are also widely adopted (Smith and Abel [1987] Chan and Ho [1994] Gillette and Silverman [2008]).

Another approach is to employ convex relaxation to convert the problem into a convex optimization problem. In this way, very efficient algorithms can be derived. In this thesis, we will also apply the SDP method for solving the source localization problem. Another method is the second order cone programming (SOCP) (Tseng [2007]) which has been applied successfully to localization problems. However, for the SOCP relaxation, it has been proven in (Tseng [2007]) that the optimized source location must lies in the convex hull of the microphone array. It is unfavourable for acoustic localisation since the speaker is usually standing in front of the microphone array instead of surrounded by the microphones.

Motivated by the works in SDP and SOCP in the literature, in this thesis, we will study the convex relaxation method for solving the source localization problem extensively. The basic idea behind the convex relaxation technique also comes from (Biswas et al. [2006a]). In order to achieve better solution, it is possible to couple the idea of SOCP together in the formulation. In view of this, a novel mixed SDP-SOCP relaxation model is proposed. We derive the characteristics of the optimal solution to the mixed SDP-SOCP method. Furthermore, for comparison with existing convex relaxation methods, we define the notion of a localizable region, in the sense that the source location can be sought exactly when the errors in TDOA are zeroes. We aim to propose a standard in comparing different convex relaxation methods using this defined localizable region. For localization problems on a plane, based on the derived solution characteristics, we derive the exact geometry for this localizable region for the proposed mixed model, and show that the region is larger than the convex hull formed by the microphone array alone and also larger than other convex relaxation methods. This is important for applications when the source is staying in front of the array instead of surrounded by the array. We continue to collect experimental data with two different array configurations and demonstrate our proposed method can indeed find very accurate estimates to the locations.

1.2 Main results

In Chapter 2, we study the design of distributed broadband beamforming system. In the configuration, we assume that each microphone is equipped with wireless communications capability. A broadband beamformer can then be designed such that the error between the actual response and the desired response is minimized. Many algorithms have been proposed to solve this optimization problem, similar to many filter design problems, large-scale linear programming techniques are often used. But the problem will be very expensive to solve if the number of filters as well as the filter lengths increase. Therefore we propose the semi-definite programming method which has polynomial time computational complexity and can be solved efficiently since the interior point algorithms can be employed. Once their mutual distance information are collected, the sensor network technology is used to estimate sensor microphone locations and is incorporated into the design process. The sensor network is also solved by the semi-definite programming method. We have studied the performance of the optimized designs and found that it was very sensitive to perturbations in microphone locations. We then propose a suitable robust formulation as a remedy to regain the performance. We illustrate the proposed method by several designs and demonstrate that this approach is essential to regain accuracy in the optimized designs.

For the design of distributed broadband beamforming system, we also apply the l_1 norm measure and the real rotation theorem to formulate it as a non-strictly convex semi-infinite linear programming problem. Zhang, Wu and López (Zhang et al. [2010a]) proposed an exchange method for the strictly convex SIP problem. For our non-strictly convex cases, we add a small perturbation quadratic function to the objective function in order to let it be strictly convex. We propose a perturbation method of exchange algorithm for solving the general convex semi-infinite programming (CSIP) problems. We prove that the solution of the perturbation semi-infinite programming problem approximates the solution of the original problem as the perturbation going to 0. The exchange algorithm for this semi-infinite programming program is proposed, whose main feature is that only those active constraints with positive Lagrange multipliers are kept, and no global optimization needs to be carried out at each iteration to detect the (almost) most violated constraint. This method is applied successfully to the broadband beamformer design problem.

The time of difference of arrival between each pair of microphones is estimated and transformed into distance difference measurements between sensors, resulting in a set of nonlinear hyperbolic equations. Sound Source is localized from these hyperbolic equations which is a highly nonlinear problem and difficult to solve. In chpater 3, we propose a novel formulation called the mixed SDP-SOCP relaxation model for source localization with time-difference information. We present a method to obtain the exact solution for the source localization. The method shows that the exact solution for the source localization can be obtained from the solutions of the mixed SDP-SOCP relaxation model and the second order polynomial equation. We also give the estimator properties for the true source location under noises. Moreover, the mixed SDP-SOCP relaxation model for source localization combined with sensor network localization problem is presented, and some statistical analyses are given.

In (Tseng [2007]), we know for single SOCP relaxation, the optimal solution must be in the convex hull of microphone array. In chapter 4, we study some properties for the solution of the mixed SDP-SOCP model in two-dimensional case. We obtain a representation theorem for the solution of the mixed SDP-SOCP model and the characterization such that the mixed SDP-SOCP model has an exact relaxation. The characterization theorem shows that the source localization with some time-difference information can be solved exactly by the mixed SDP-SOCP relaxation model in a larger region than the triangle region determined by three points, which means the exact solution region by our mixed SDP-SOCP model is not only the convex hull formed by the microphone array, but also outside the convex hull.

For illustration, many examples are presented in Chapter 5. In Chapter 6, we give four methods to find the time difference of arrival (TDOA), and use the cross correlation method to solve our actual problem.

1.3 Preliminaries

Positive semidefinite matrices

A symmetric $n \times n$ matrix A is said to be positive semidefinite, denoted by $A \succeq 0$, if $x^T Ax \geq 0$ for all $x \in \mathbb{R}^n$. On the other hand, a symmetric matrix A is said to be positive definite, denoted by $A \succ 0$, if $x^T Ax > 0$ for all non-zero $x \in \mathbb{R}^n$.

The next two theorems present several equivalent characterizations of symmetric positive semidefinite and positive definite matrices.

Theorem 1.3.1 *Let A be an $n \times n$ real symmetric matrix of rank r. Then the following statements are equivalent:*

1. *A is positive semidefinite.*
2. *All eigenvalues of A are nonnegative.*
3. *There exists an $n \times r$ matrix S such that $A = SS^T$*
4. *All principal minors of A are nonnegative.*

Theorem 1.3.2 *Let A be an $n \times n$ real symmetric matrix, then the following statements are equivalent:*

1. *A is positive definite.*
2. *All eigenvalues of A are positive.*
3. *There exists a nonsingular matrix S such that $A = SS^T$.*
4. *All leading principal minors of A are positive.*

Theorem 1.3.3 *Let A be an $n \times n$ matrix and let B be a nonsingular $n \times n$ matrix. Then $A \succeq 0$ if and only if $B^T AB \succeq 0$.*

Proof. Assume that $A \succeq 0$ and let x be any vector in $\mathbb{R}^n$. Then $x^T B^T ABx = (Bx)^T A(Bx) \geq 0$ since $A \succeq 0$. Therefore, $B^T AB \succeq 0$.

On the other hand, assume that $B^T AB \succeq 0$, let x be any vector in $\mathbb{R}^n$. Since B is nonsingular, there exists $y \in \mathbb{R}^n$ such that $By = x$. Thus $x^T Ax = y^T B^T ABy \geq 0$ since $B^T AB \succeq 0$. Therefore, $A \succeq 0$. ■

The next theorem, known as Schur's Complement Lemma, is well known.

Theorem 1.3.4 (Schur) *Let*

$$A = \begin{pmatrix} A_1 & A_2 \\ A_2^T & A_3 \end{pmatrix}$$

be a symmetric matrix, and assume that $A_1 \succ 0$. Then $A \succeq 0$ if and only if $A_3 - A_2^T A_1^{-1} A_2 \succeq 0$.

Proof.

$$\begin{pmatrix} I & 0 \\ -A_2^T A_1^{-1} & I \end{pmatrix} \begin{pmatrix} A_1 & A_2 \\ A_2^T & A_3 \end{pmatrix} \begin{pmatrix} I & -A_1^{-T} A_2 \\ 0 & I \end{pmatrix} = \begin{pmatrix} A_1 & 0 \\ 0 & A_3 - A_2^T A_1^{-1} A_2 \end{pmatrix}.$$

Since $\begin{pmatrix} I & -A_1^{-T}A_2 \\ 0 & I \end{pmatrix}$ is nonsingular, then it follows from Theorem 1.3.3 that A is positive semidefinite if and only if $A_3 - A_2^T A_1^{-1} A_2 \succeq 0$. ■

Semidefinite programming

Semidefinite programming (SDP) (Wolkowicz et al. [2003]) is a generalization of linear programming (LP) where the decision variables are arranged in a symmetric matrix instead of a vector, and the non-negative orthant is replaced by the cone of positive semidefinite matrices. Thus the following is an SDP problem.

$$\begin{aligned} \inf \;\; & \operatorname{tr}(CX) \\ \text{s.t.} \;\; & \operatorname{tr}(A^i X) = b_i, \quad for\ i = 1, \cdots, m \\ & X \succeq 0, \end{aligned} \tag{1.3.1}$$

where $C, A^1, \ldots, A^m$ are given $n \times n$ symmetric matrices, $b_1, \ldots, b_m$ are given scalars, and X is an $n \times n$ symmetric positive semidefinite matrix whose entries are the decision variables.

Semidefinite programming has been a very active area of research since the early 1990's for its many important applications in several areas of sciences and engineering especially in the areas of combinatorial optimization and control theory. One of the early successful applications of SDP in combinatorial optimization was the Goemans-Williamson's randomized algorithm (Goemans and Williamson [1995]) for the well-known max-cut problem.

Semi-infinite programming

Semi-infinite programming is an optimization problem with a finite number of variables and an infinite number of constraints, or an infinite number of variables and a finite number of constraints. It can be stated as follows:

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & g(x,s) \leq 0, \quad \forall s \in \Omega \end{aligned} \tag{1.3.2}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $g : \mathbb{R}^n \times \Omega \to \mathbb{R}$ are continuous functions, and Ω is a given nonempty compact set in $\mathbb{R}^p$.

Second order cone programming

Second order cone programming (SOCP) is a convex optimization problem in which a linear function is minimized over the intersection of an affine linear manifold with the Cartesian product of second order (Lorentz) cones. We define second order cone:

$$\mathcal{Q} = \{\boldsymbol{x} = (x_0, \bar{\boldsymbol{x}}) \in \mathbb{R}^n : x_0 \geq \|\bar{\boldsymbol{x}}\|\},$$

where $\|\cdot\|$ is the standard Euclidean norm.

The second order cone programming can be written as follows:

$$\begin{aligned} \min \quad & c_1^T x_1 + \cdots c_r^T x_r \\ \text{s.t.} \quad & A_1 x_1 + \cdots + A_r x_r = b \\ & x_i \succeq_{\mathcal{Q}} 0, \quad for \ i = 1, \cdots, r \end{aligned} \tag{1.3.3}$$

where $c_i \in \mathbb{R}^{n_i}$, A_i are given $m \times n_i$ matrices, b is a given m-dimension vector, and x_i are n_i-dimension decision variables.

We denote arrow-shaped matrix $\mathrm{Arw}(\boldsymbol{x})$ as:

$$\mathrm{Arw}(\boldsymbol{x}) = \begin{pmatrix} x_0 & \bar{\boldsymbol{x}}^T \\ \bar{\boldsymbol{x}} & x_0 I \end{pmatrix}$$

Since $\mathrm{Arw}(\boldsymbol{x}) \succeq 0$ if and only if either $\boldsymbol{x} = 0$, or $x_0 > 0$ and the Schur complement $x_0 - \bar{\boldsymbol{x}}^T (x_0 I)^{-1} \bar{\boldsymbol{x}} \geq 0$. It means that $\boldsymbol{x} \succeq_{\mathcal{Q}} 0$ if and only if $\mathrm{Arw}(\boldsymbol{x})$ is a positive semidefinite matrix and $\boldsymbol{x} \succ_{\mathcal{Q}} 0$ if and only if $\mathrm{Arw}(\boldsymbol{x})$ is a positive definite matrix. Thus we can see that the SOCP is a special case of semidefinite programming.

Chapter 2

Design of distributed beamforming system using semi-definite programming and semi-infinite programming

2.1 Introduction

In this chpater, broadband beamformers design with wireless microphone array is studied. For many applications, the design problem can be formulated as a minimax optimization problem. Here, we use an efficient method called Semi-definite programming (SDP) to solve it. Then we use sensor network technique to estimate the wireless microphones, which just need a few anchor nodes in the network together with the pairwise distance measurements between any two nodes. SDP method is also used for solving the problem. Owing to perturbations in the estimated sensor locations, an appropriate robust formulation is represented to allow for certain amount of errors. The sensor network technology is incorporated into broadband beamforming design. We will demonstrate by examples that the proposed robust approach is essential to regain the accuracy in the designs if microphone locations are indeed erroneous.

Zhang, Wu and López (Zhang et al. [2010a]) proposed an exchange method for the strictly convex SIP problem. The algorithm associated with the method terminates in a finite iterations needs to be under some strictly convex conditions. Our filter design problem can be formulated as a non-strictly convex semi-infinite programming (SIP) problem. Because this SIP problem is not strictly convex, the exchange method in (Zhang et al. [2010a]) can not be directly applied to the filter design problem. For solving the non-strictly convex cases, our idea is to add a small perturbation quadratic function to the objective function in order to let it be strictly convex. We will prove in this chapter that the solution of the perturbation SIP problem approximates the solution of the original problem as the perturbation going to 0. The method is applied successfully to the filter design problem in the chapter.

The rest of the paper is organized as follows. In Section 2, we formulate the wireless beamformer design problem and the localization problem of microphones. Then, we introduce its corresponding robust problem. We transform all the problems into equivalent SDP problems. For illustration, several examples are solved. In Section 3, a new SIP problem associated with the filter design problem is formulated. A perturbation method and the convergence theorem of general convex semi-definite programming problem are presented. The application of the method to the filter design problem is given.

2.2 Design of distributed beamforming system using semi-definite programming

2.2.1 Formulation

The structure of a wireless near-field broadband beamformer can be found in Figure 2.1, where the positions of microphones can be arbitrary and the sound signal is received by the microphone array and processed by the FIR filters behind.

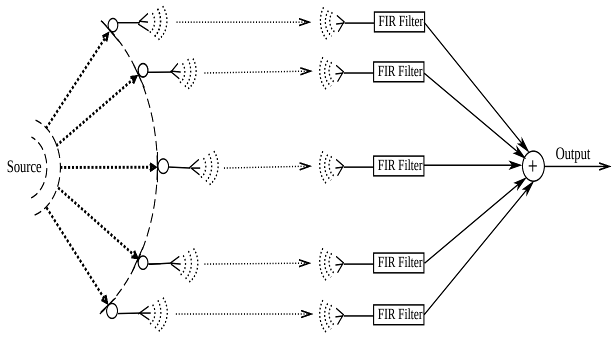

Figure 2.1: The structure of a wireless near-field beamformer.

We assume that there are N elements in the array. Using a simple spherical model, the transfer function from the source point $\boldsymbol{r}$ to the i-th element of the broadband beamformer is given by

$$A_i(\boldsymbol{r}, f) = \frac{1}{\|\boldsymbol{r} - \boldsymbol{r}_i\|} e^{-j2\pi f \|\boldsymbol{r} - \boldsymbol{r}_i\|/c}, \tag{2.2.1}$$

where $\boldsymbol{r}$ is the position vector of the source signal, $\boldsymbol{r}_i$ is the position vector of the i-th microphone, f is the frequency, and c is the sound speed. Then, the array response vector is therefore given by

$$\boldsymbol{a}(\boldsymbol{r}, f) = (A_1(\boldsymbol{r}, f), \ldots, A_N(\boldsymbol{r}, f))^\intercal. \tag{2.2.2}$$

Let each microphone signal be sampled at a rate of f_s, and suppose that each FIR filter has L taps. Denote the filter response vector by

$$\boldsymbol{d}_0(f) = \left(1, e^{-j2\pi f/f_s}, \ldots, e^{-j2\pi f(L-1)/f_s}\right)^\intercal \tag{2.2.3}$$

and the filter coefficients by

$$\boldsymbol{w} = (\boldsymbol{w}_1^\intercal, \ldots, \boldsymbol{w}_N^\intercal)^\intercal, \tag{2.2.4}$$

where

$$\boldsymbol{w}_i = (w_i(0), \ldots, w_i(L-1))^\intercal, \qquad i = 1, \ldots, N,$$

then the actual response of the broadband beamformer is given by

$$G(\boldsymbol{r}, f) = \boldsymbol{w}^\intercal \boldsymbol{d}(\boldsymbol{r}, f)$$

with

$$\boldsymbol{d}(\boldsymbol{r}, f) = \boldsymbol{a}(\boldsymbol{r}, f) \otimes \boldsymbol{d}_0(f),$$

where $\otimes$ denotes the Kronecker product and the dimension of $\boldsymbol{w}$ is $n = N \times L$.

Let $G_d(\boldsymbol{r}, f)$ be the specified desired response of the broadband beamformer, and consider a region $\Omega = \cup_{i=1}^m \Omega_i$ in the space-frequency domain where each Ω_i is a convex set and $\Omega_i \cap \Omega_j = \emptyset$ for $i \neq j$. Then, the minimax design problem can be formulated as

$$\min_{\boldsymbol{w} \in \mathbb{R}^n} \max_{(\boldsymbol{r}, f) \in \Omega} |\boldsymbol{w}^\intercal \boldsymbol{d}(\boldsymbol{r}, f) - G_d(\boldsymbol{r}, f)|.$$

Obviously, if the term $|\boldsymbol{w}^\intercal \boldsymbol{d}(\boldsymbol{r}, f) - G_d(\boldsymbol{r}, f)|$ above is replaced by $|\boldsymbol{w}^\intercal \boldsymbol{d}(\boldsymbol{r}, f) - G_d(\boldsymbol{r}, f)|^2$, the optimal solution will not be changed. Hence, we can formulate the filter design problem as

Problem 2.2.1 *Find a coefficient vector $\boldsymbol{w} \in \mathbb{R}^n$ of the FIR filters to minimize the following cost function*

$$\max_{(\boldsymbol{r}, f) \in \Omega} |\boldsymbol{w}^\intercal \boldsymbol{d}(\boldsymbol{r}, f) - G_d(\boldsymbol{r}, f)|^2. \tag{2.2.5}$$

For wireless microphones, since they can be placed in anywhere, it is more practical if we can estimate the locations spontaneously. In fact, the locations of the microphones can be estimated by a method whose principle is the same as the localization problem in sensor network. That is, to estimate the locations of the microphones, we need to have some points whose locations are known. These points are called anchors and can be denoted by $\boldsymbol{a} = \{\boldsymbol{a}_k :\in \mathbb{R}^h, k \in M^{(1)}\}$, where $M^{(1)}$ is the index set of the anchors and h is the dimension which can be 1, 2 or 3, depending on the structure of the anchors. The unknown microphones are called sensors, which can be denoted by $\boldsymbol{r} = \{\boldsymbol{r}_j :\in \mathbb{R}^h, j \in M^{(2)}\}$, where $M^{(2)}$ is the index set of the sensors. An example of sensors and anchors can be seen in Figure 2.2, where three diamond points are the anchors and two circle points are the sensors.

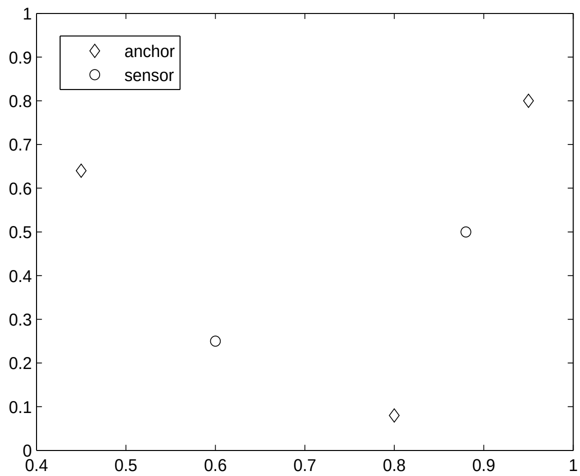

Figure 2.2: An example of sensor network.

For every pair of points, we can estimate the distance. That is, we have Euclidean distance measures $\hat{d}_{kj}$ between anchor $\boldsymbol{a}_k$ and sensor $\boldsymbol{r}_j$ for some k, j, and $\hat{d}_{ij}$ between sensor $\boldsymbol{r}_i$ and sensor $\boldsymbol{r}_j$ for some $i < j$. Denoting $N_a = \{(k, j) : k \in M^{(1)}, j \in M^{(2)}\}$ and $N_r = \{(i, j) : i < j, i \in M^{(1)}, j \in M^{(1)}\}$, we have

$$\begin{aligned} \|\boldsymbol{a}_k - \boldsymbol{r}_j\|^2 &= \hat{d}_{kj}^2 \qquad \forall (k, j) \in N_a, \\ \|\boldsymbol{r}_i - \boldsymbol{r}_j\|^2 &= \hat{d}_{ij}^2 \qquad \forall (i, j) \in N_r. \end{aligned} \tag{2.2.6}$$

From these information of distances, we can then estimate the locations of the sensors. The localization problem is to find the sensor coordinates $\boldsymbol{r}$ such that (2.2.6) is satisfied. The localization problem is equivalent to the optimization problem below:

Problem 2.2.2 *Find the locations $\boldsymbol{r}$ of the sensors to minimize*

$$F(\boldsymbol{r}) = \sum_{(i,j)\in N_r} \left| \|\boldsymbol{r}_i - \boldsymbol{r}_j\|^2 - \hat{d}_{ij}^2 \right| + \sum_{(k,j)\in N_a} \left| \|\boldsymbol{a}_k - \boldsymbol{r}_j\|^2 - \hat{d}_{kj}^2 \right|. \tag{2.2.7}$$

Thus, we can find the locations $\boldsymbol{r}$ of the sensors by optimizing the cost function (2.2.7). If the optimal value F^* is zero, then the solution obtained is the exact locations. However, in most cases, the optimal value F^* is strictly greater than zero. To see this, we suppose that the number of $\boldsymbol{a}$ is m. Then, the number of N_a is mN and the number of N_r is $N(N-1)/2$. Hence, the total number of the equalities in (2.2.6) is $mN + N(N-1)/2$. On the other hand, the number of decision variables is hN. Then, when $mN + N(N-1)/2 > hN$, that is, $m + (N-1)/2 > h$, this problem is over-determined. Since $h \leq 3$, this condition is satisfied in most cases. Hence, the optimal value of Problem 2.2.2 is strictly greater than zero and the obtained locations are not exact in most cases. Since the locations $\boldsymbol{r}$ is not exact and the performance of the designed beamformer is very sensitive to the errors in the locations, a robust design is needed.

Similar to Problem 2.2.1, we consider a corresponding robust problem where the location vector contains certain uncertainties. Denote the position vector by $\tilde{\boldsymbol{r}} = p(\boldsymbol{r}, \theta)$, where $\boldsymbol{r}$ is a position vector in Problem 2.2.1 and $\theta \in [-\eta, \eta]$ is the parameter for uncertainty. Without loss of generality, we define $p(\boldsymbol{r}, 0) = \boldsymbol{r}$. We can formulate the robust filter design problem as

Problem 2.2.3 *Find a coefficient vector $\boldsymbol{w} \in \mathbb{R}^n$ of the FIR filters to minimize the following cost function*

$$\max_{\theta\in[-\eta,\eta]} \max_{(\boldsymbol{r},f)\in\Omega} |\boldsymbol{w}^\intercal \boldsymbol{d}(\tilde{\boldsymbol{r}}, f) - G_d(\tilde{\boldsymbol{r}}, f)|^2. \tag{2.2.8}$$

Both Problem 2.2.1 and Problem 2.2.3 are nonlinear minimax optimization problems. After the discretization of the space-frequency domain $\tilde{\Omega} = [-\eta, \eta] \times \Omega$, gradient-based methods can be applied to solve for numerical solutions. However, if the discretization of $\tilde{\Omega}$ is very large, these problems become very expensive to solve. Thus, an algorithm with polynomial time computational complexity is desirable.

2.2.2 Methodology

Robust broadband beamformer design

The cost functions in Problem 2.2.1 and Problem 2.2.3 are quadratic. They can be rearranged as SDP problems as follows. Expanding the complex functions

$$\boldsymbol{d}(\boldsymbol{r}, f) = \boldsymbol{d}_1(\boldsymbol{r}, f) + j\boldsymbol{d}_2(\boldsymbol{r}, f),$$

$$G_d(\boldsymbol{r}, f) = G_{d_1}(\boldsymbol{r}, f) + jG_{d_2}(\boldsymbol{r}, f),$$

and denoting

$$u(\boldsymbol{r}, f) = (\boldsymbol{w}^\intercal \boldsymbol{d}_1(\boldsymbol{r}, f) - G_{d_1}(\boldsymbol{r}, f)),$$

$$v(\boldsymbol{r}, f) = (\boldsymbol{w}^\intercal \boldsymbol{d}_2(\boldsymbol{r}, f) - G_{d_2}(\boldsymbol{r}, f)),$$

by adding an additional variable z, Problem 2.2.1 becomes

$$\begin{aligned} &\min_{\boldsymbol{w}\in\mathbb{R}^n, z\in\mathbb{R}} && z \\ &s.t. && u(\boldsymbol{r}, f)^2 + v(\boldsymbol{r}, f)^2 \le z, \quad \forall(\boldsymbol{r}, f) \in \Omega. \end{aligned} \tag{2.2.9}$$

We will make use of the following theorem proven in (Chen [2001]):

Theorem 2.2.1 *Let A be an $n \times n$ real symmetric matrix of rank r. Then, the following statements are equivalent:*

1. *A is positive semi-definite.*
2. *All eigenvalues of A are nonnegative.*
3. *There exists an $n \times r$ matrix S such that $A = SS^T$.*

4. All principal minors of A are nonnegative.

By Theorem 2.2.1, the constraint in the above problem holds if and only if

$$\Phi(z, \boldsymbol{w}, \boldsymbol{r}, f) = \begin{pmatrix} z & u(\boldsymbol{r}, f) & v(\boldsymbol{r}, f) \\ u(\boldsymbol{r}, f) & 1 & 0 \\ v(\boldsymbol{r}, f) & 0 & 1 \end{pmatrix} \succeq 0, \ \forall (r, f) \in \Omega, \tag{2.2.10}$$

where " $\succeq$ " denotes the positive semi-definite symbol. Denote

$$G(z, \boldsymbol{w}) = diag\{\Phi(z, \boldsymbol{w}, \boldsymbol{r}^1, f^1), \ldots, \Phi(z, \boldsymbol{w}, \boldsymbol{r}^k, f^k)\}, \tag{2.2.11}$$

where $\Omega_d = \{(\boldsymbol{r}^1, f^1), \ldots, (\boldsymbol{r}^k, f^k)\} \subset \Omega$ is a set of dense grid points. Then, we transformed Problem 2.2.1 into a SDP optimization problem:

Problem 2.2.4 *Find a coefficient vector $\boldsymbol{w} \in \mathbb{R}^n$ of the FIR filters and z, such that z is minimized, subject to the constraint*

$$G(z, \boldsymbol{w}) \succeq 0. \tag{2.2.12}$$

Similarly to Problem 2.2.1, Problem 2.2.3 can also be transformed into an SDP optimization problem. Denote

$$\tilde{G}(z, \boldsymbol{w}) = diag\{\Phi(z, \boldsymbol{w}, p(\boldsymbol{r}^1, \theta^1), f^1), \ldots, \Phi(z, \boldsymbol{w}, p(\boldsymbol{r}^k, \theta^k), f^k)\}, \tag{2.2.13}$$

where $\tilde{\Omega}_d = \{(\theta^1, \boldsymbol{r}^1, f^1), \ldots, (\theta^k, \boldsymbol{r}^k, f^k)\} \subset \tilde{\Omega}$ is a set of dense grid points. Then, Problem 2.2.3 is transformed into an SDP optimization problem:

Problem 2.2.5 *Find a coefficient vector $\boldsymbol{w} \in \mathbb{R}^n$ of the FIR filters and z, such that z is minimized, subject to the constraint*

$$\tilde{G}(z, \boldsymbol{w}) \succeq 0. \tag{2.2.14}$$

Basically, since there is a discretization of the interval $[-\eta, \eta]$, Problem 2.2.5 is more expensive to solve than Problem 2.2.4. However, if η is small, it's not necessary to do the whole discretization of the interval $[-\eta, \eta]$ and Problem 2.2.5 can be simplified. This can be seen in the next theorem.

Theorem 2.2.2 *Suppose that η is small. Then, for any given coefficients vector $\boldsymbol{w} \in \mathbb{R}^n$ and frequency f, we have*

$$\begin{aligned}
&\max_{\theta\in[-\eta,\eta]} |G(p(\boldsymbol{r},\theta),f) - G_d(p(\boldsymbol{r},\theta),f)|^2 = \\
&\max\left\{|G(p(\boldsymbol{r},-\eta),f) - G_d(p(\boldsymbol{r},-\eta),f)|^2, |G(p(\boldsymbol{r},\eta),f) - G_d(p(\boldsymbol{r},\eta),f)|^2\right\} + |o(\eta)|.
\end{aligned} \tag{2.2.15}$$

Proof. Denote $H(f) = (H_1(f), \ldots, H_N(f))^\intercal$, where $H_i(f) = \boldsymbol{w}_i^\intercal \boldsymbol{d}_0(f)$. Then, the actual frequency response $G(p(\boldsymbol{r},\theta),f)$ can be reformulated as

$$G(p(\boldsymbol{r},\theta),f) = H^\intercal(f)\boldsymbol{b}(p(\boldsymbol{r},\theta),f). \tag{2.2.16}$$

Denote the gradient of $\boldsymbol{b}(p(\boldsymbol{r},\theta),f)$ with respect to the parameter θ as

$$\frac{\partial \boldsymbol{b}(p(\boldsymbol{r},\theta),f)}{\partial\theta} = \left(\frac{\partial B_1(p(\boldsymbol{r},\theta),f)}{\partial\theta}, \ldots, \frac{\partial B_N(p(\boldsymbol{r},\theta),f)}{\partial\theta}\right)^\intercal. \tag{2.2.17}$$

Then, since $p(\boldsymbol{r},0) = \boldsymbol{r}$ and η is small, we can rewrite $G(p(\boldsymbol{r},\theta),f)$ as

$$G(p(\boldsymbol{r},\theta),f) = H^\intercal(f)\left(\boldsymbol{b}(\boldsymbol{r},f) + \frac{\partial \boldsymbol{b}(p(\boldsymbol{r},0),f)}{\partial\theta}\theta\right) + o(\theta). \tag{2.2.18}$$

Similarly, $G_d(p(\boldsymbol{r},\theta),f)$ can be rewritten as

$$G_d(p(\boldsymbol{r},\theta),f) = G_d(\boldsymbol{r},f) + \frac{\partial G_d(p(\boldsymbol{r},0),f)}{\partial\theta}\theta + o(\theta). \tag{2.2.19}$$

Then, we have

$$\begin{aligned}
&|G(p(\boldsymbol{r},\theta),f) - G_d(p(\boldsymbol{r},\theta),f)|^2 \\
=&\left|G(\boldsymbol{r},f) - G_d(\boldsymbol{r},f) + \left(H^\intercal(f)\frac{\partial \boldsymbol{b}(p(\boldsymbol{r},0),f)}{\partial\theta} - \frac{\partial G_d(p(\boldsymbol{r},0),f)}{\partial\theta}\right)\theta + o(\theta)\right|^2 \\
=&\left|G(\boldsymbol{r},f) - G_d(\boldsymbol{r},f) + \left(H^\intercal(f)\frac{\partial \boldsymbol{b}(p(\boldsymbol{r},0),f)}{\partial\theta} - \frac{\partial G_d(p(\boldsymbol{r},0),f)}{\partial\theta}\right)\theta\right|^2 + |o(\theta)|.
\end{aligned} \tag{2.2.20}$$

Note that the first term of the right hand side of (2.2.20) is convex with respect to θ and the maximum of a convex function exists when θ is in the boundary. Then, we have

$$
\begin{aligned}
&\max_{\theta\in[-\eta,\eta]}\left|G(\boldsymbol{r},f)-G_d(\boldsymbol{r},f)+\left(H^{\intercal}(f)\frac{\partial \boldsymbol{b}(p(\boldsymbol{r},0),f)}{\partial\theta}-\frac{\partial G_d(p(\boldsymbol{r},0),f)}{\partial\theta}\right)\theta\right|^2\\
=&\max\left\{\left|G(\boldsymbol{r},f)-G_d(\boldsymbol{r},f)+\left(H^{\intercal}(f)\frac{\partial \boldsymbol{b}(p(\boldsymbol{r},0),f)}{\partial\theta}-\frac{\partial G_d(p(\boldsymbol{r},0),f)}{\partial\theta}\right)\eta\right|^2,\right.\\
&\left.\left|G(\boldsymbol{r},f)-G_d(\boldsymbol{r},f)-\left(H^{\intercal}(f)\frac{\partial \boldsymbol{b}(p(\boldsymbol{r},0),f)}{\partial\theta}-\frac{\partial G_d(p(\boldsymbol{r},0),f)}{\partial\theta}\right)\eta\right|^2\right\}\\
=&\max\left\{|G(p(\boldsymbol{r},-\eta),f)-G_d(p(\boldsymbol{r},-\eta),f)+o(\eta)|^2,\right.\\
&\left.|G(p(\boldsymbol{r},\eta),f)-G_d(p(\boldsymbol{r},\eta),f)+o(\eta)|^2\right\}\\
=&\max\left\{|G(p(\boldsymbol{r},-\eta),f)-G_d(p(\boldsymbol{r},-\eta),f)|^2,\right.\\
&\left.|G(p(\boldsymbol{r},\eta),f)-G_d(p(\boldsymbol{r},\eta),f)|^2\right\}+|o(\eta)|. \qquad (2.2.21)
\end{aligned}
$$

This completes the proof. ■

By Theorem 2.2.2, if η is small, Problem 2.2.5 can be simplified to

Problem 2.2.6 *Find a coefficient vector $\boldsymbol{w}\in\mathbb{R}^n$ of the beamformer filters and z, such that z is minimized, subject to the constraint*

$$\hat{G}(z,\boldsymbol{w})\succeq 0, \qquad (2.2.22)$$

where

$$
\begin{aligned}
\hat{G}(z,\boldsymbol{w})=diag\{&\Phi(z,\boldsymbol{w},p(\boldsymbol{r}^1,\eta),f^1),\Phi(z,\boldsymbol{w},p(\boldsymbol{r}^1,-\eta),f^1),\\
&\ldots,\Phi(z,\boldsymbol{w},p(\boldsymbol{r}^k,\eta),f^k),\Phi(z,\boldsymbol{w},p(\boldsymbol{r}^k,-\eta),f^k)\}. \qquad (2.2.23)
\end{aligned}
$$

Localization of microphones

Let $\boldsymbol{R}(=[\boldsymbol{r}_1,\boldsymbol{r}_2,\ldots,\boldsymbol{r}_n])\in\mathbb{R}^{h\times N}$ be the unknown matrix. Then, we have

$$
\begin{aligned}
\|\boldsymbol{r}_i-\boldsymbol{r}_j\|^2&=\boldsymbol{e}_{ij}^{\intercal}\boldsymbol{R}^{\intercal}\boldsymbol{R}\boldsymbol{e}_{ij},\\
\|\boldsymbol{a}_k-\boldsymbol{r}_j\|^2&=\begin{pmatrix}\boldsymbol{a}_k^{\intercal} & \boldsymbol{e}_j^{\intercal}\end{pmatrix}\begin{bmatrix}\boldsymbol{I}\\ \boldsymbol{R}^{\intercal}\end{bmatrix}[\boldsymbol{I}\ \boldsymbol{R}]\begin{pmatrix}\boldsymbol{a}_k\\ \boldsymbol{e}_j\end{pmatrix},
\end{aligned}
$$

where $\boldsymbol{e}_{ij}$ is the vector with 1 at the i-th position, -1 at the j-th position and 0 elsewhere, $\boldsymbol{e}_j$ is the vector with -1 at the j-th position and 0 elsewhere. Let $\boldsymbol{Y} = \boldsymbol{R}^\intercal \boldsymbol{R}$, then (2.2.6) is equivalent to find a symmetric matrix $\boldsymbol{Y} \in \mathbb{R}^{N\times N}$ and a matrix $\boldsymbol{R} \in \mathbb{R}^{h\times N}$ such that the following equations are satisfied:

$$\begin{aligned}
&\boldsymbol{e}_{ij}^\intercal \boldsymbol{Y} \boldsymbol{e}_{ij} = \hat{d}_{ij}^2, && \forall (i,j) \in N_r \\
&\left(\boldsymbol{a}_k^\intercal\ \boldsymbol{e}_j^\intercal\right) \begin{pmatrix} \boldsymbol{I} & \boldsymbol{R} \\ \boldsymbol{R}^\intercal & \boldsymbol{Y} \end{pmatrix} \begin{pmatrix} \boldsymbol{a}_k \\ \boldsymbol{e}_j \end{pmatrix} = \hat{d}_{kj}^2, && \forall (k,j) \in N_a \\
&\boldsymbol{Y} = \boldsymbol{R}^\intercal \boldsymbol{R}.
\end{aligned} \tag{2.2.24}$$

To relax the sensor network localization problem, we relax $\boldsymbol{Y} = \boldsymbol{R}^\intercal \boldsymbol{R}$ to $\boldsymbol{Y} \succeq \boldsymbol{R}^\intercal \boldsymbol{R}$ which is equivalent to (Boyd et al. [1994]):

$$\boldsymbol{Z} := \begin{pmatrix} \boldsymbol{I} & \boldsymbol{R} \\ \boldsymbol{R}^\intercal & \boldsymbol{Y} \end{pmatrix} \succeq 0.$$

Then, the relaxed version of the problem (2.2.24) can be represented as a standard semi-definite programming model, that is, we need to find a symmetric matrix $\boldsymbol{Z} \in \mathbb{R}^{(h+N)\times(h+N)}$ such that the following equations are satisfied:

$$\begin{aligned}
&(\boldsymbol{b}^\intercal\ \boldsymbol{0}^\intercal)\boldsymbol{Z} \begin{pmatrix} \boldsymbol{b} \\ \boldsymbol{0} \end{pmatrix} = \boldsymbol{b}^\intercal \boldsymbol{b}, && \text{for some vectors } \boldsymbol{b} \in \mathbb{R}^h \\
&(\boldsymbol{0}^\intercal\ \boldsymbol{e}_{ij}^\intercal)\boldsymbol{Z} \begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{e}_{ij} \end{pmatrix} = \hat{d}_{ij}^2, && \forall (i,j) \in N_r \\
&\left(\boldsymbol{a}_k^\intercal\ \boldsymbol{e}_j^\intercal\right) \boldsymbol{Z} \begin{pmatrix} \boldsymbol{a}_k \\ \boldsymbol{e}_j \end{pmatrix} = \hat{d}_{kj}^2, && \forall (k,j) \in N_a \\
&\boldsymbol{Z} \succeq 0.
\end{aligned} \tag{2.2.25}$$

The first set of equations in (2.2.25) is to assure that the first $h \times h$ submatrix of $\boldsymbol{Z}$ is $\boldsymbol{I}$. The number of the vector $\boldsymbol{b}$ depends on the dimension h. If $h = 1$, there is only one element in $\boldsymbol{I}$ and the minimum number of $\boldsymbol{b}$ is 1. If $h = 2$, since $\boldsymbol{I}$ is symmetric, there are three elements to be determined and the minimum number of $\boldsymbol{b}$ is 3. If $h = 3$, there are six elements in the symmetric matrix $\boldsymbol{I}$ to be determined and the minimum number of $\boldsymbol{b}$ is 6. There are many choices for $\boldsymbol{b}$. An example can be seen in Table 2.1.

h	$\boldsymbol{b}$
1	1
2	$(1\ 0)^\intercal$, $(0\ 1)^\intercal$, $(1\ 1)^\intercal$
3	$(1\ 0\ 0)^\intercal$, $(0\ 1\ 0)^\intercal$, $(0\ 0\ 1)^\intercal$, $(1\ 1\ 0)^\intercal$, $(1\ 0\ 1)^\intercal$, $(0\ 1\ 1)^\intercal$

Table 2.1: A typical choice of the vector $\boldsymbol{b}$.

A relaxed solution $\boldsymbol{Z}$ can be obtained by solving equations (2.2.25). However, as we discuss in previous section, the second and third set of equations in (2.2.25) are not satisfied in most cases and the solution does not exist. Therefore, we need to consider Problem 2.2.2. To transform Problem 2.2.2 into a semi-definite programming problem, we add some nonnegative slack variables as $\boldsymbol{\alpha} = \{\alpha_{ij}^+, \alpha_{ij}^-, \alpha_{kj}^+, \alpha_{kj}^- :\geq 0, \forall (i,j) \in N_r, \forall (k,j) \in N_a\}$. Then, Problem 2.2.2 is reformulated as

Problem 2.2.7 *Find* $\boldsymbol{\alpha}$ *and the locations* $\boldsymbol{r}$ *of the sensors, such that the cost function*

$$\sum_{(i,j)\in N_r} (\alpha_{ij}^+ + \alpha_{ij}^-) + \sum_{(k,j)\in N_a} (\alpha_{kj}^+ + \alpha_{kj}^-)$$

is minimized, subject to the constraints

$$\begin{aligned} \|\boldsymbol{r}_i - \boldsymbol{r}_j\|^2 - \hat{d}_{ij}^2 &= \alpha_{ij}^+ - \alpha_{ij}^-, \quad \forall (i,j) \in N_r \\ \|\boldsymbol{a}_k - \boldsymbol{r}_j\|^2 - \hat{d}_{kj}^2 &= \alpha_{kj}^+ - \alpha_{kj}^-, \quad \forall (k,j) \in N_a. \end{aligned} \tag{2.2.26}$$

With the introduced relaxed matrix $\boldsymbol{Z}$, Problem 2.2.7 is transformed into a standard SDP problem:

Problem 2.2.8 *Find* $\boldsymbol{\alpha}$ *and the symmetric matrix* $\boldsymbol{Z}$*, such that*

$$\sum_{(i,j)\in N_r} (\alpha_{ij}^+ + \alpha_{ij}^-) + \sum_{(k,j)\in N_a} (\alpha_{kj}^+ + \alpha_{kj}^-)$$

is minimized, subject to the constraints

$$
\begin{aligned}
&(\boldsymbol{b}^\intercal\ \mathbf{0}^\intercal)\boldsymbol{Z}\begin{pmatrix}\boldsymbol{b}\\ \mathbf{0}\end{pmatrix} = \boldsymbol{b}^\intercal\boldsymbol{b}, && \text{for some vectors } \boldsymbol{b}\in\mathbb{R}^h\\
&(\mathbf{0}^\intercal\ \boldsymbol{e}_{ij}^\intercal)\boldsymbol{Z}\begin{pmatrix}\mathbf{0}\\ \boldsymbol{e}_{ij}\end{pmatrix} - \alpha_{ij}^+ + \alpha_{ij}^- = \hat{d}_{ij}^2, && \forall(i,j)\in N_r\\
&\left(\boldsymbol{a}_k^\intercal\ \boldsymbol{e}_j^\intercal\right)\boldsymbol{Z}\begin{pmatrix}\boldsymbol{a}_k\\ \boldsymbol{e}_j\end{pmatrix} - \alpha_{kj}^+ + \alpha_{kj}^- = \hat{d}_{kj}^2, && \forall(k,j)\in N_a\\
&\boldsymbol{Z}\succeq 0.
\end{aligned}
\tag{2.2.27}
$$

Problem 2.2.8 is a semi-definite programming problem which can be solved by any SDP software. Note that any solution of Problem 2.2.8 has at least rank h. For a localizable system, we need to impose certain conditions on the rank of $\boldsymbol{Z}$ and the relaxation of $\boldsymbol{Y}$. This is summarized in the following.

Definition 2.2.1 *The localization problem is localizable if there is a unique localization in $\mathbb{R}^h$ and there is no $\boldsymbol{r}_j\in\mathbb{R}^{h'}$, $j=1,\ldots,n$, where $h'>h$, such that*

$$\|\boldsymbol{r}_i-\boldsymbol{r}_j\|^2 = d_{ij}^2,\quad \forall(i,j)\in N_r$$

$$\left\|\begin{pmatrix}\boldsymbol{a}_k\\ \mathbf{0}\end{pmatrix}-\boldsymbol{r}_j\right\|^2 = d_{kj}^2,\quad \forall(k,j)\in N_a.$$

The latter says that the problem cannot be localized in a higher dimension space where the locations of the anchors are augmented to $(\boldsymbol{a}_k^\intercal\ \mathbf{0}^\intercal)^\intercal\in\mathbb{R}^{h'}, k\in M^{(1)}$.

Then, we have the following theorems (proven in So and Ye [2007]):

Theorem 2.2.3 *The following statements are equivalent:*

1. *The problem is localizable.*
2. *The max rank of the solution $\boldsymbol{Z}$ has rank h.*

3. *The solution* $\boldsymbol{Z}$ *satisfy* $\boldsymbol{Y} = \boldsymbol{R}^{\intercal}\boldsymbol{R}$ *or* $Trace(\boldsymbol{Y} - \boldsymbol{R}^{\intercal}\boldsymbol{R}) = 0$.

Theorem 2.2.4 *If a problem contains a subproblem that is localizable, then the submatrix solution corresponding to the subproblem in the SDP solution has rank* h*. That is, the SDP relaxation computes a solution that localize all possibly localizable unknown sensor points.*

From these two theorems, we can see that the solution to the SDP problem provides the first and second moment information on $\boldsymbol{R}$ (Bertsimas and Ye [1998]). After we find a solution $\boldsymbol{Z}$ by solving Problem 2.2.8, $\boldsymbol{r}_j$ will be the estimated position of j-th microphone and $Y_{jj} - \|\boldsymbol{r}_j\|^2$ will be used as its perturbation. The total perturbation of the microphones is then given by

$$Trace(\boldsymbol{Y} - \boldsymbol{R}^{\intercal}\boldsymbol{R}) = \sum_{j=1}^{N} \left(Y_{jj} - \|\boldsymbol{r}_j\|^2 \right).$$

2.2.3 Illustrative examples

In solving the formulated linear SDP problems (Problem 2.2.4, Problem 2.2.5, Problem 2.2.6 and Problem 2.2.8), interior point algorithms can be applied. There are several software packages available, such as LMI control toolbox (Gahinet et al. [1995]), SDPA-M (Fujisawa et al. [2005]), SDPSOL (Wu and Boyd [1996]), SeDuMi (Sturm [1998]). All these software packages can be applied. In this section, we use SDPA-M (Fujisawa et al. [2005]) and the computation was performed in Matlab.

The proposed method is first used to design several broadband beamformers with different target performances. At the same time, we will study the performances of the designs towards errors in speaker and microphone locations. We focus on multimedia applications and the desired frequency response function will include the frequency range of human voice together with a range of positions that the speaker is located. We choose the desired response function as

$$G_d(\boldsymbol{r}, f) = \begin{cases} e^{-j2\pi f\left(\frac{\|r-r_c\|}{c}+\frac{L-1}{2}T\right)}, & \text{if } (\boldsymbol{r}, f) \text{ is in passband region} \\ 0, & \text{if } (\boldsymbol{r}, f) \text{ is in stopband region} \end{cases},$$

where $\boldsymbol{r}_c$ is the coordinate for the center element, the sound speed is $c = 340.9m/s$ and the sample increment is $T = 125\mu s$, that is, the sampling rate is set as 8kHz.

In the first example, we consider an equispaced linear array with five elements. To avoid spatial aliasing for the frequency of interest, the element spacing is 5cm. That is, they are located at the coordinates $\{(-0.1, 0), (-0.05, 0), \ldots, (0.1, 0)\}$. A seven-tap FIR filter behind each element is used. The passband region and stopband region are specified on an x-axis parallel with, and $y = 1$ meter in front of, the array. The passband region is defined as

$$\{(\boldsymbol{r}, f) : -0.4m \le x \le 0.4m,\ y = 1m,\ 0.5kHz \le f \le 1.5kHz\}$$

while the stopband region is the union of several parts as

$$\{(\boldsymbol{r}, f) : -0.4m \le x \le 0.4m,\ y = 1m,\ 2.5kHz \le f \le 4kHz\},$$

$$\{(\boldsymbol{r}, f) : 1.5m \le |x| \le 2.5m,\ y = 1m,\ 0.5kHz \le f \le 1.5kHz\},$$

$$\{(\boldsymbol{r}, f) : 1.5m \le |x| \le 2.5m,\ y = 1m,\ 2.5kHz \le f \le 4kHz\}.$$

The complexity of the implementation depends on the discretization of the space-frequency domain Ω in this problem. Suppose that the number of discretization of Ω is given by $m_x \times m_f$. Then, for different numbers of discretization, the comparison of our method with the SIP method (Yiu et al. [2003]) is given in Table 2.2.

From Table 2.2, we see that our method is more efficient than SIP method (Yiu et al. [2003]), especially when the number of discretization becomes very large. The amplitude of the actual response $G(\boldsymbol{r}, f)$ is shown in Figure 2.3.

Next, we consider the robust filter design for the above problem, where there is a small perturbation in y-axis in both the passband region and stopband region.

$m_x \times m_f$	LP	SDP
40×40	$17.66s$	$8.98s$
80×80	$299.63s$	$36.97s$
120×120	$1490.25s$	$85.04s$
130×130	$2192.06s$	$102.89s$

Table 2.2: Comparison of the running times (seconds)

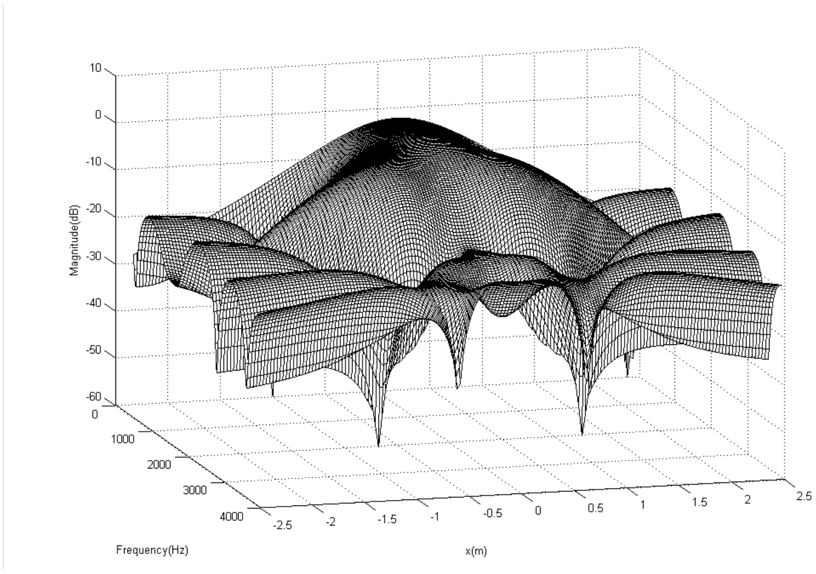

Figure 2.3: Amplitude of $G(\boldsymbol{r}, f)$ in Example 1 where $N = 5$, $L = 7$ and $y = 1m$.

We set $y = 1 + \theta$, $\theta \in [-0.1, 0.1]$. Then, the number of discretization of Ω is denoted by $m_x \times m_f \times m_\theta$, where m_θ is the number of discretization of θ. In order to demonstrate the design, we employ a grid size of $40 \times 40 \times 10$ for the discretization. The optimized function value is 0.0388 (-14.11dB). However, if we apply Theorem 2.2.2 to solve this problem, that is, θ only take values at the boundary point $\{-0.1, 0.1\}$, then the optimal function value obtained is the same as -14.11dB (in fact, the difference between these two values is less than the allowable error 10^{-8}).

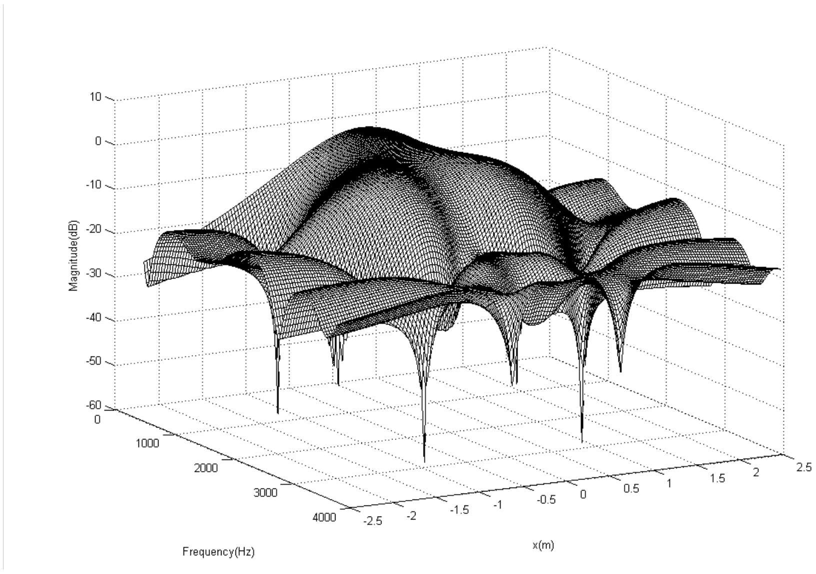

Figure 2.4: Amplitude of $G(\boldsymbol{r}, f)$ with robust design in Example 1, where $N = 5$, $L = 7$ and $y = 1m$.

The robust property of the result can be depicted when the source moves. In Figure 2.4, the filter has been designed with $y = 1m$. Then, the source is subsequently displaced to calculate the final amplitude response. Here, when the source moves to $y = 0.9m$ and $y = 1.1m$, the amplitudes of the actual response are depicted in Figure 2.5 and Figure 2.6, respectively. When the speaker location changes, we found that the optimized performance is very similar with or without robustness in the formulation. This again confirms the findings in (Yiu et al. [2003]) that the optimal design is not too sensitive to the movement of speaker.

Next, we consider another case of this robust filter design, where the first

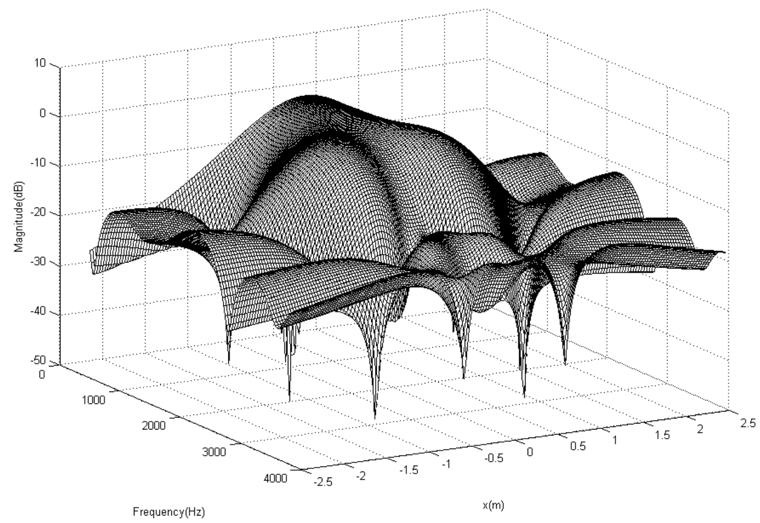

Figure 2.5: Amplitude of $G(\boldsymbol{r}, f)$ with robust design in Example 1, where $N = 5$, $L = 7$ and $y = 0.9m$.

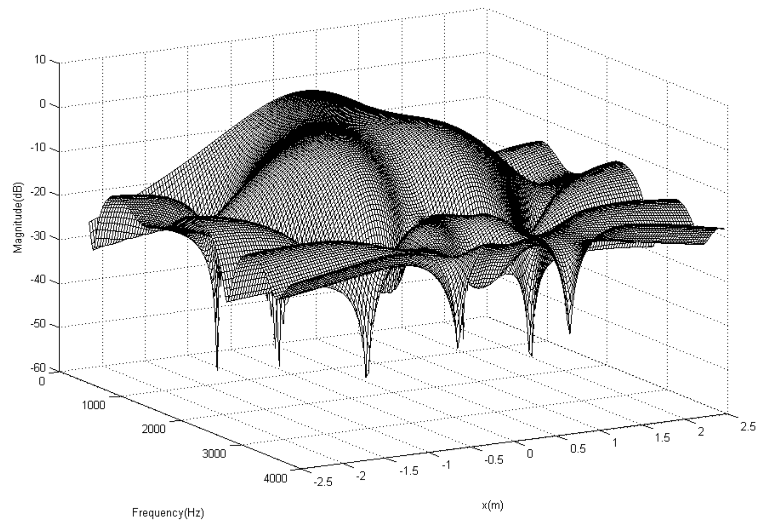

Figure 2.6: Amplitude of $G(\boldsymbol{r}, f)$ with robust design in Example 1, where $N = 5$, $L = 7$ and $y = 1.1m$.

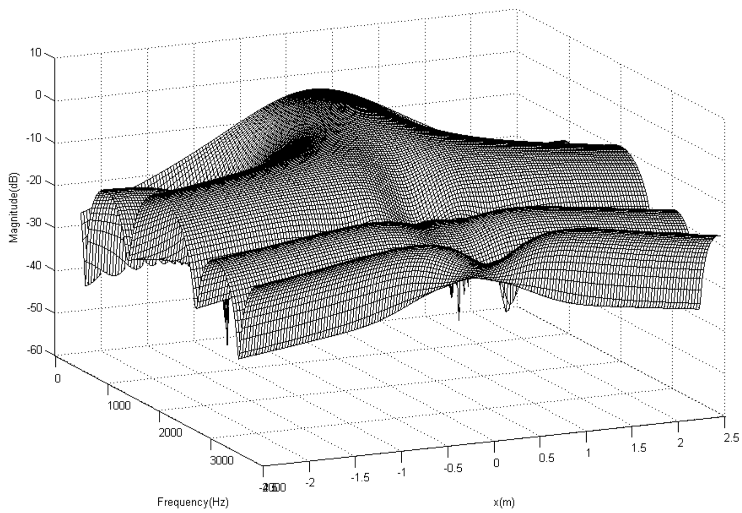

Figure 2.7: Amplitude of $G(\boldsymbol{r}, f)$ with robust design in Example 1, where $N = 5$, $L = 7$, $x_1 = -0.15m$ and $x_5 = 0.15m$.

and the last microphones have perturbations. We set $x_1 \in [-0.15, -0.07]$ and $x_5 \in [0.07, 0.15]$. In order to demonstrate the design, we employ a grid size of $40 \times 40 \times 2$ for the discretization. The optimized function value is 0.0379 (-14.21dB). When the x-coordinate of the first and the last microphones move to $-0.15m$ and $0.15m$, respectively, the amplitude of the actual responses are shown in Figure 2.7.

For this case, if we do not use the robust filter design to design the beamformer, then the amplitude of the actual response is depicted in Figure 2.8. This performance is poor, compared to Figure 2.7. Detail of the comparison between using robust filter design and not using robust filter design is given in Table 2.3. From the results, it is clear that the optimized beamformer is very sensitive to perturbations in microphone locations. With the use of the robust formulation, the optimized performance is recovered in spite of the perturbations.

In the second example, we consider a filter design problem, where there is an equispaced linear array with seven elements. They are located at the coordinates $\{(-0.15, 0), (-0.1, 0), \ldots, (0.15, 0)\}$. A twenty one-tap FIR filter behind each

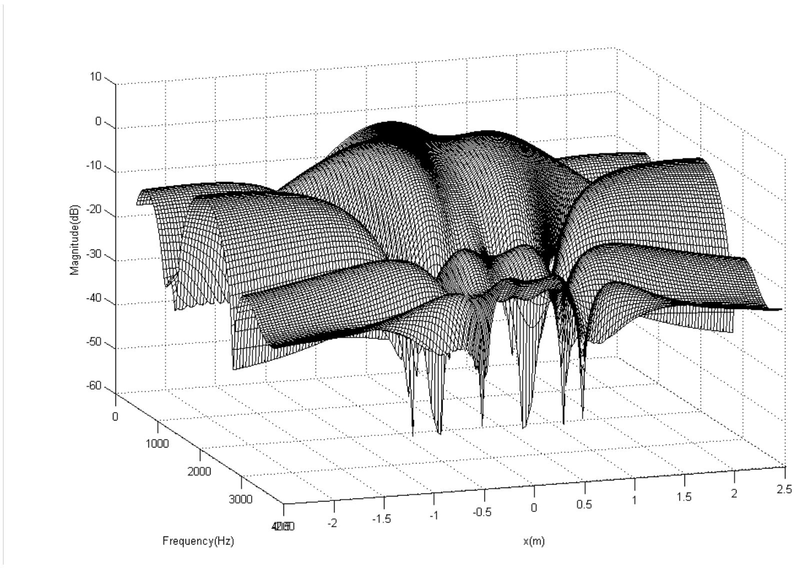

Figure 2.8: Amplitude of $G(\boldsymbol{r}, f)$ without robust design in Example 1, where $N = 5$, $L = 7$, $x_1 = -0.15m$ and $x_5 = 0.15m$.

Methods	Passband gain	Passband ripple(dB)	Stopband ripple(dB)
1	1.03146	0.20281	-14.23836
2	1.05639	0.36851	-6.87912
3	0.98606	0.34450	-5.32115

Table 2.3: 1. using robust filter formulated in Example 1; 2. using filter formulated in Example 1 without robust consideration; 3. using traditional LP method (Yiu et al. [2003]) without robust consideration.

element is used. The passband region is defined as

$$\{(\boldsymbol{r}, f) : -0.4m \leq x \leq 0.4m,\ 0.15m \leq y \leq 0.25m,\ 0.3kHz \leq f \leq 3kHz\}$$

while the stopband region is the union of several parts as

$$\{(\boldsymbol{r}, f) : -0.4m \leq x \leq 0.4m,\ 0.1m \leq y \leq 0.3m,\ 3.3kHz \leq f \leq 4kHz\},$$

$$\{(\boldsymbol{r}, f) : 1.5m \leq |x| \leq 2.5m,\ 0.1m \leq y \leq 0.3m,\ 0.3kHz \leq f \leq 3kHz\},$$

$$\{(\boldsymbol{r}, f) : 1.5m \leq |x| \leq 2.5m,\ 0.1m \leq y \leq 0.3m,\ 3.3kHz \leq f \leq 4kHz\}.$$

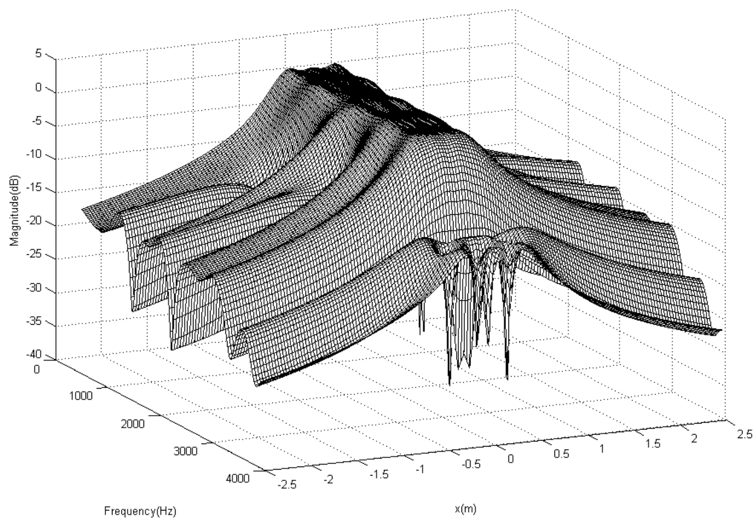

Figure 2.9: Amplitude of $G(\boldsymbol{r}, f)$ in Example 2, where $N = 7$, $L = 21$, and $y = 0.2m$.

For this example, we can treat it as a robust filter design problem by setting $y = 0.2 + \theta$, where $\theta \in [-0.1, 0.1]$. Then, we can apply Theorem 2.2.2 to solve this problem. In order to demonstrate the design, we employ a grid size of 40×40 for the discretization. The optimized function value is 0.0856 (-10.68dB). The amplitude of the actual response $G(\boldsymbol{r}, f)$ (θ=0) is shown in Figure 2.9.

To see the robust property of the result when the source moves, we move the source to $y = 0.1m$ and $y = 0.3m$. The corresponding amplitudes of the actual responses are depicted in Figure 2.10 and Figure 2.11. From these two figures, we see that the changes of the performances are not much when the source moves.

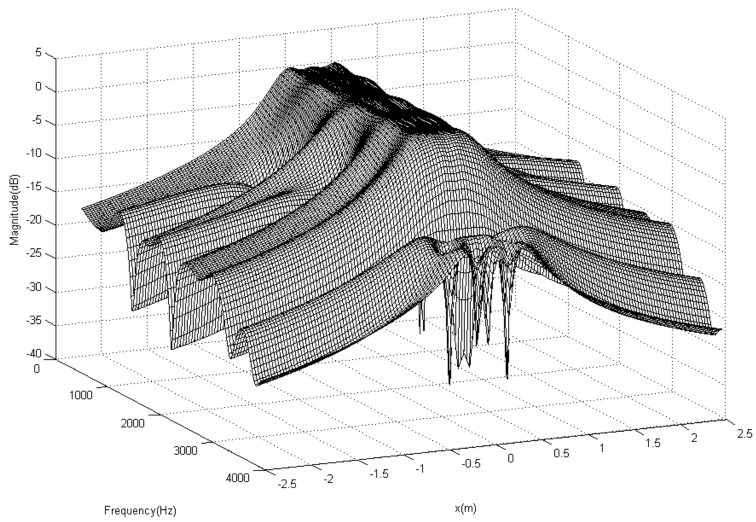

Figure 2.10: Amplitude of $G(\boldsymbol{r}, f)$ in Example 2, where $N = 7$, $L = 21$, and $y = 0.1m$.

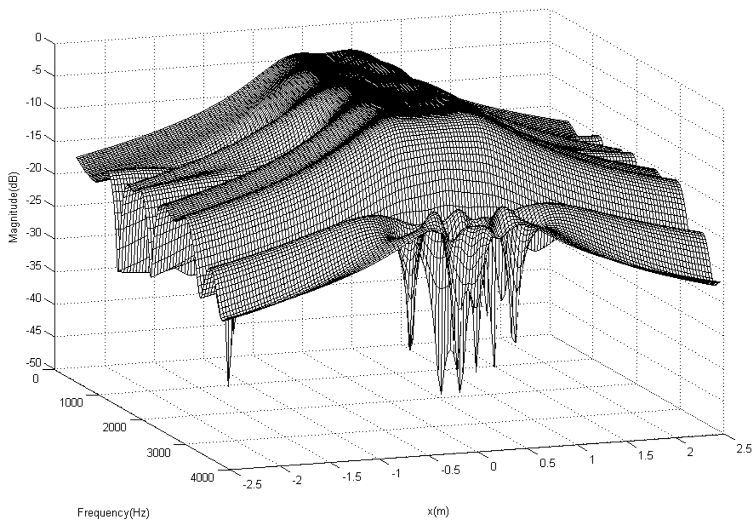

Figure 2.11: Amplitude of $G(\boldsymbol{r}, f)$ in Example 2, where $N = 7$, $L = 21$, and $y = 0.3m$.

In the third example, we consider five microphones which are located at the coordinates $\{(-0.1, -0.2),$
$(-0.05, -0.1), (0, 0), (0.05, -0.1), (0.1, -0.2)\}$. A seven-tap FIR filter is used behind each element. In this example, the passband region and stopband region are the same as those in Example 1. In order to demonstrate the design, we employ a grid size of 40×40 for the discretization. The optimized function value is 0.0475 (-13.23dB). The amplitude of the actual response $G(r, f)$ is shown in Figure 2.12.

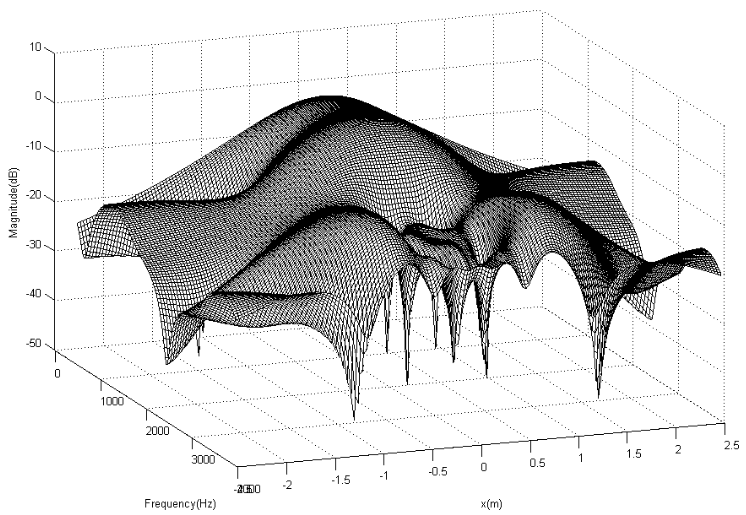

Figure 2.12: Amplitude of $G(\boldsymbol{r}, f)$ in Example 3 where $N = 5$, $L = 7$ and $y = 1m$.

Next, we consider the robust filter design of this example. The same as Example 1, we assume that $x_1 \in [-0.15, -0.07]$ and $x_5 \in [0.07, 0.15]$. In order to demonstrate the design, we employ a grid size of $40 \times 40 \times 2$ for the discretization. The optimized function value is 0.0520 (-12.84dB). When the x-coordinate of the first and the last microphones move to -0.15m and 0.15m, respectively, the amplitude of the actual response is shown in Figure 2.13.

If we do not use the robust design, the amplitude of the actual response is shown in Figure 2.14 when the x-coordinate of the first and the last microphones move to -0.15m and 0.15m, respectively. The detail of the comparison between using robust filter design and not using robust filter design is given in Table 2.4. We can see that the performance of the robust design scheme is the best.

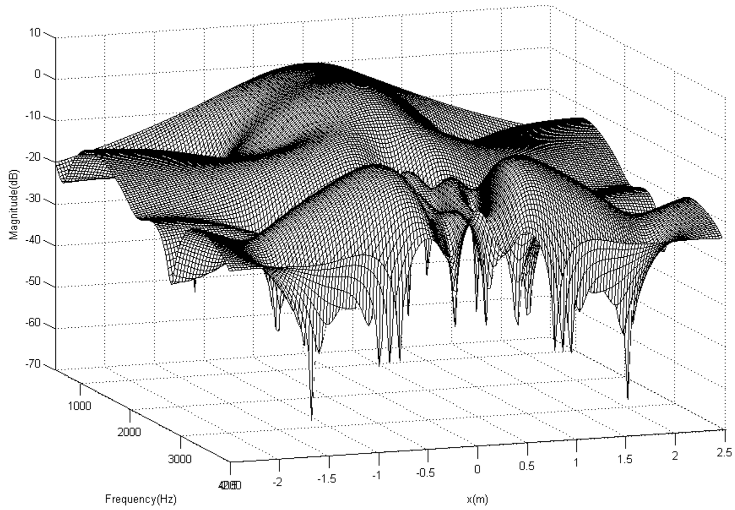

Figure 2.13: Amplitude of $G(\boldsymbol{r}, f)$ with robust design in Example 3, where $N = 5$, $L = 7$, $x_1 = -0.15m$ and $x_5 = 0.15m$.

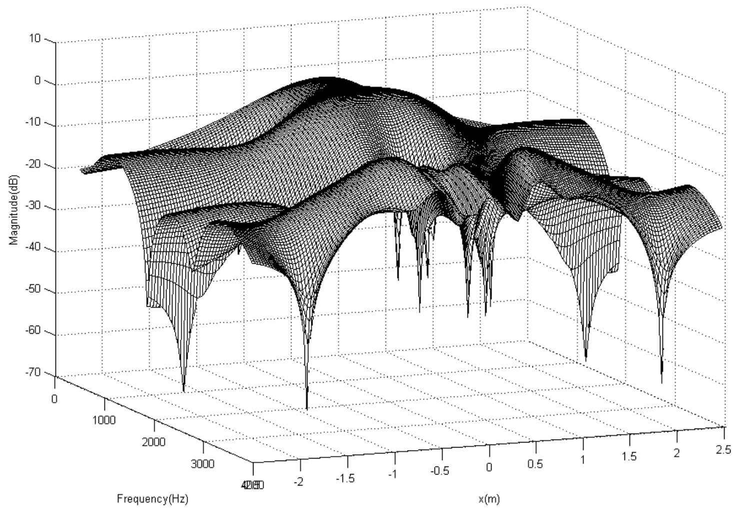

Figure 2.14: Amplitude of $G(\boldsymbol{r}, f)$ without robust design in Example 3, where $N = 5$, $L = 7$, $x_1 = -0.15m$ and $x_5 = 0.15m$.

Methods	Passband gain	Passband ripple(dB)	Stopband ripple(dB)
1	1.03246	0.23434	−12.82194
2	1.03612	0.28552	−8.57016
3	1.06463	0.26534	−10.30237

Table 2.4: 1. using robust filter formulated in Example 3; 2. using filter formulated in Example 3 without robust consideration; 3. using traditional LP method (Yiu et al. [2003]) without robust consideration.

In the last example, we demonstrate how to incorporate the sensor localization method together with the robust design formulation. We consider a 5m×5m classroom and the speaker stands in the middle of the room. At the corner of the room, there are two anchors with coordinates $\{(-2.5, 2.5), (2.5, 2.5)\}$. Seven microphones are located at the coordinates $\{(-0.15, 0.7), (-0.1, 0.8), (-0.05, 0.9), (0, 1),$ $(0.05, 0.9),$
$(0.1, 0.8), (0.15, 0.7)\}$. We assume the distances between the nodes can be estimated and there exist errors in the estimated distances. We simulate the estimated distances similar to (Wang et al. [2008b]). That is, we add a random error to the estimated distance:

$$\hat{d_{ij}} = d_{ij} \cdot (1 + \epsilon \times N_f)$$

where N_f is a given noisy factor between $[0, 1]$ and ϵ is a standard normal random variable. For this example, we further assume that the microphone with coordinate $(0, 1)$ is also an anchor, and all the other microphones are sensors. The noisy factor is chosen as 0.005 and the distances between the nodes can be estimated. In the first stage, we estimate the microphones' positions by solving Problem 2.2.8. The estimated positions are illustrated in Figure 2.15 and the perturbations of these six sensors are $0.0139, 0.0007, 0.0004, 0.0005, 0.0004$ and 0.0018, respectively. Since the perturbations of three sensors with coordinates $(-0.05, 0.9)$, $(0.05, 0.9)$ and $(0.1, 0.8)$ are very small, they are neglected and we just consider the uncertainties of the other three sensors.

In the second stage, with the estimated positions and the perturbations, we design a robust beamformer with a seven-tap filter behind each microphone. The

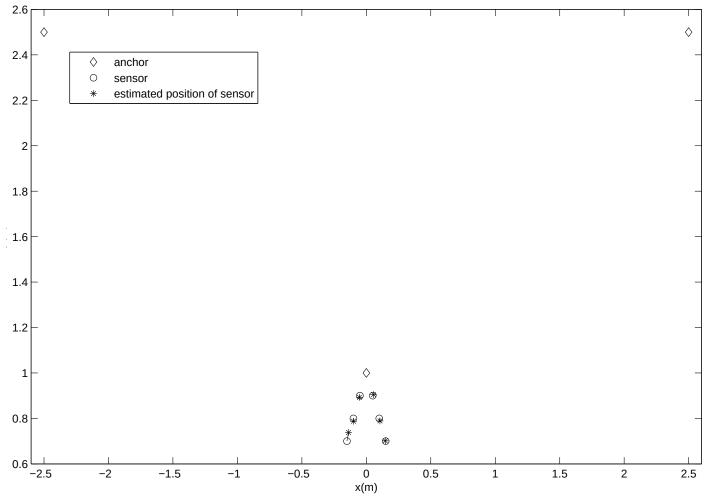

Figure 2.15: Estimated positions via solving the localization problem.

passband region, stopband region and the desired frequency response function are chosen in the same manner as in Example 1. By solving Problem 2.2.6, we obtain the optimal design. The amplitude of the actual response $G(r, f)$ is shown in Figure 2.16.

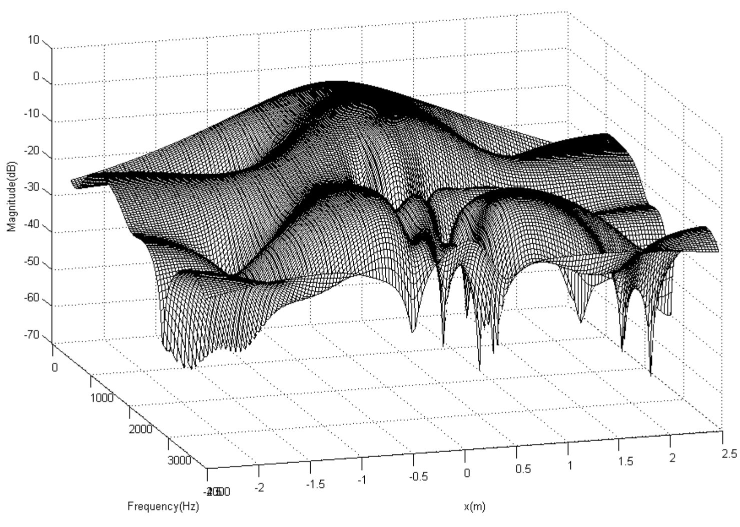

Figure 2.16: Amplitude of $G(\boldsymbol{r}, f)$ in Example 4.

2.3 Design of distributed beamforming system using semi-infinite programming

2.3.1 Formulation

In this section, we transform the filter design problem into a new semi-infinite programming problem.

In the last section, the filter design problem and the robust filter design problem are formulated as the following

Problem 2.3.1 *Find a coefficient vector $\boldsymbol{w} \in \mathbb{R}^n$ of the FIR filters, such that the cost function*

$$\max_{(\boldsymbol{r},f)\in\Omega} |\boldsymbol{w}^\intercal \boldsymbol{d}(\boldsymbol{r}, f) - G_d(\boldsymbol{r}, f)| \tag{2.3.28}$$

is minimized.

Problem 2.3.2 *Find a coefficient vector $\boldsymbol{w} \in \mathbb{R}^n$ of the FIR filters, such that the cost function*

$$\max_{\theta\in[-\eta,\eta]} \max_{(\boldsymbol{r},f)\in\Omega} |\boldsymbol{w}^\intercal \boldsymbol{d}(\tilde{\boldsymbol{r}}, f) - G_d(\tilde{\boldsymbol{r}}, f)| \tag{2.3.29}$$

is minimized.

We expand the complex functions as

$$\boldsymbol{d}(\boldsymbol{r}, f) = \boldsymbol{d}_1(\boldsymbol{r}, f) + j\boldsymbol{d}_2(\boldsymbol{r}, f),$$

$$G_d(\boldsymbol{r}, f) = G_{d_1}(\boldsymbol{r}, f) + jG_{d_2}(\boldsymbol{r}, f),$$

and denote

$$u(\boldsymbol{r}, f) = (\boldsymbol{w}^\intercal \boldsymbol{d}_1(\boldsymbol{r}, f) - G_{d_1}(\boldsymbol{r}, f)), \tag{2.3.30}$$

$$v(\boldsymbol{r}, f) = (\boldsymbol{w}^\intercal \boldsymbol{d}_2(\boldsymbol{r}, f) - G_{d_2}(\boldsymbol{r}, f)). \tag{2.3.31}$$

An alternative and more flexible way to convert the design problem into a semi-infinite linear programming problem is to control the real part and the imaginary part separately by introducing two new variables as:

$$z_1 = \max_{(\boldsymbol{r},f)\in\Omega} |u(\boldsymbol{r},f))|, \quad z_2 = \max_{(\boldsymbol{r},f)\in\Omega} |v(\boldsymbol{r},f))|.$$

Thus, the design problem can be formulated as the following semi-infinite programming:

$$\begin{cases} \qquad\qquad \min_{\boldsymbol{z}\in\mathbb{R}^{n+2}} \boldsymbol{b}^T\boldsymbol{z} \\ \text{subject to} \\ \qquad\qquad A(\boldsymbol{r},f)\boldsymbol{z} - \boldsymbol{G}(\boldsymbol{r},f) \leq 0 \qquad \forall(\boldsymbol{r},f)\in\Omega \end{cases} \qquad LP(\Omega)$$

where $\boldsymbol{z} = (\boldsymbol{w}, z_1, z_2)^T$, $\boldsymbol{b} = (\boldsymbol{0}, \phi_1, \phi_2)^T$,

$$A(\boldsymbol{r},f) = \begin{pmatrix} \boldsymbol{d}_1(\boldsymbol{r},f)^T & -1 & 0 \\ -\boldsymbol{d}_1(\boldsymbol{r},f)^T & -1 & 0 \\ \boldsymbol{d}_2(\boldsymbol{r},f)^T & 0 & -1 \\ -\boldsymbol{d}_2(\boldsymbol{r},f)^T & 0 & -1 \end{pmatrix}, \quad \boldsymbol{G}(\boldsymbol{r},f) = \begin{pmatrix} G_{d_1}(\boldsymbol{r},f) \\ -G_{d_1}(\boldsymbol{r},f) \\ G_{d_2}(\boldsymbol{r},f) \\ -G_{d_2}(\boldsymbol{r},f) \end{pmatrix},$$

in which ϕ_1 and ϕ_2 are two different weights for the real and imaginary parts, respectively.

Define

$$V(\boldsymbol{z}) = \boldsymbol{b}^T\boldsymbol{z} \text{ and } g(\boldsymbol{z},(\boldsymbol{r},f)) = A(\boldsymbol{r},f)\boldsymbol{z} - \boldsymbol{G}(\boldsymbol{r},f)$$

Then the above problem $LP(\Omega)$ can be represented as

$$\begin{cases} \min \quad V(\boldsymbol{z}) \\ \text{s.t.} \quad g(\boldsymbol{z},(\boldsymbol{r},f)) \leq 0 \qquad \forall(\boldsymbol{r},f)\in\Omega. \end{cases} \qquad (BP)$$

2.3.2 Perturbation exchange algorithm and its convergence

Zhang, Wu and López (Zhang et al. [2010a]) proposed an exchange method for the convex SIP problem (P) whose main feature is that only those active constraints with positive Lagrange multipliers are kept, and no global optimization needs to be carried out at each iteration to detect the (almost) most violated constraint. The algorithm associated with the method terminates in a finite iterations needs to be under some strictly convex conditions.

In this section, we propose a perturbation method of exchange algorithm for solving general convex semi-infinite programming (CSIP) problems and prove that the solution of the perturbation SIP problem approximates the solution of the original CSIP as the perturbation going to 0.

Perturbation exchange algorithm

A convex semi-infinite programming (CSIP) problem can be written as follows:

$$\begin{cases} \min \quad f(x) \\ \text{s.t.} \quad g(x,s) \leq 0 \text{ for any } s \in \Omega. \end{cases} \tag{P}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $g(s,\cdot) : \mathbb{R}^n \to \mathbb{R}$ are continuous convex functions, and Ω is a given nonempty compact set in $\mathbb{R}^p$ (or in $\mathbb{C}^p$).

We also focus on the convex SIP problem (P) satisfying the following condition (A):

Assumption A.

(i). f is convex and continuously differentiable on $\mathbb{R}^n$;

(ii). For any $s \in \Omega$, $g(\cdot, s)$ is convex and $\nabla_x g(x,s)$ exists and is continuous on $\mathbb{R}^n$;

(iii). Slater constraint qualification (SCQ) holds; i.e., there exists $\hat{x} \in \mathbb{R}^n$ such that $g(\hat{x}, s) < 0$ for all $s \in \Omega$.

(iv). There exists a finite subset Ω_0 such that for every $\lambda \in \mathbb{R}$, the set

$$\mathcal{F}_\lambda^0 := \{x \in \mathbb{R}^n; f(x) \leq \lambda \text{ and } g(x, s) \leq 0 \text{ for all } s \in \Omega_0\}$$

is bounded when nonempty.

We consider a perturbation of the problem (P) as follows:

$$\begin{cases} \min & f_\epsilon(x) \\ \text{s.t.} & g(x, s) \leq 0 \text{ for any } s \in \Omega. \end{cases} \tag{P_ϵ}$$

where

$$f_\epsilon(x) = f(x) + \epsilon \|x\|^2. \tag{2.3.32}$$

For given finite set $\mathcal{R} = \{s_j, j = 1, \cdots, m\} \subset \Omega$, we consider the finitely constrained convex programming problem:

$$\begin{cases} \min & f(x) + \epsilon \|x\|^2 \\ \text{s.t.} & g(x, s_j) \leq 0 \text{ for any } j = 1, \cdots, m. \end{cases} \tag{$P_\epsilon(\mathcal{R})$}$$

Remark 2.3.1 *Let $x^* \in \mathbb{R}^n$ be a feasible solution of $(P_\epsilon(\mathcal{R}))$. It is known that (Luenberger [2004]) x^* is optimal if and only if there exist multipliers $\lambda^* \in \mathbb{R}^m$ such that (x^*, λ^*) satisfies the following Karush-Kuhn-Tucker (KKT) conditions:*

$$\begin{aligned} &\nabla f(x) + \sum_{j=1}^{m} \lambda(s_j) \nabla_x g(x, s_j) = 0, \\ &\lambda(s_j) \geq 0,\ g(x, s_j) \leq 0,\quad \lambda(s_j) g(x, s_j) = 0,\ j = 1, \cdots, m, . \end{aligned} \tag{KKT}$$

Now, applying Algorithm 2.1 in (Zhang et al. [2010a]) to $(P_\epsilon(\mathcal{R}))$, we obtain a **perturbation exchange algorithm (PEA)** which is described as follows.

For given a small $\eta \in (0, 1/2)$, choose $\epsilon = \epsilon(\eta) > 0$ small enough such that

$$\epsilon \sup_{\boldsymbol{z} \in \mathcal{F}_{\bar{\lambda}}^0} \|\boldsymbol{z}\|^2 \leq \eta,$$

and $\bar{\lambda} = 1 + f(\bar{\boldsymbol{z}})$ for some given $\bar{\boldsymbol{z}} \in \{x \in \mathbb{R}^n; g(x,s) \leq 0, \forall s \in \Omega\}$.

Step 0. Choose a finite reference set $\mathcal{R}_0 = \{s_j^0, j = 1, \cdots, m_0\} \subset \Omega$ such that $\Omega_0 \subset \mathcal{R}_0$, and f_ϵ is level bounded on the feasible set of $(P_\epsilon(\mathcal{R}_0))$. Let x_0 be an optimal solution to $(P_\epsilon(\mathcal{R}_0))$ and let $\{\lambda(s_j^0)), j = 1, \cdots, m_0\} \in \mathbb{R}^m$ be the set of associated multipliers. Set $k = 0$.

Step 1. Find a point $s_{new}^k \in \Omega$ such that

$$g(x_k, s_{new}^k) > \eta.$$

If such a point does not exist, then stop. Otherwise, put $\bar{\mathcal{R}}_{k+1} = \mathcal{R}_k \cup \{s_{new}^k\}$.

Step 2. Let x_{k+1} be an optimal solution to $(P_\epsilon(\bar{\mathcal{R}}_{k+1}))$ and let $\{\lambda_{k+1}(s), s \in \bar{\mathcal{R}}_{k+1}\}$ be the set of associated multipliers.

Step 3. Define new reference sets $\mathcal{R}_{k+1} = \bar{\mathcal{R}}_{k+1} \setminus \mathcal{R}_{k+1}^l$, where the leaving reference set

$$\mathcal{R}_{k+1}^l := \left\{ s \in \bar{\mathcal{R}}_{k+1}; \lambda_{k+1}(s) = 0 \right\}.$$

Set $k = k + 1$, and return to Step 1.

Remark 2.3.2 *It is obvious that the optimal solution x_{k+1} to $(P_\epsilon(\bar{\mathcal{R}}_{k+1}))$ also solves $(P_\epsilon(\mathcal{R}_{k+1}))$.*

Let x_k be an optimal solution to $(P_\epsilon(\mathcal{R}_k))$ and let v_k denote the optimal value of $(P_\epsilon(\mathcal{R}_k))$. Let $\Lambda_k = \{\lambda_k(s_j), j = 1, \cdots, m,\}$ be the corresponding Lagrange multiplier. Since f_ϵ is strictly convex, by the KKT's condition (KKT), if the algorithm (PEA) does not terminate in k iterations, then

$$\begin{aligned} v_{k+1} - v_k =& f_\epsilon(x_{k+1}) - f_\epsilon(x_k) > \nabla f_\epsilon(x_k)^T (x_{k+1} - x_k) \\ =& - \sum_{s_j \in \mathcal{R}_k} \lambda_k(s_j) \nabla_x g(x_k, s_j)^T (x_{k+1} - x_k) \\ \geq& \sum_{s_j \in \mathcal{R}_k} \lambda_k(s_j) \left(g(x_k, s_j) - g(x_{k+1}, s_j) \right) \\ =& - \sum_{s_j \in \mathcal{R}_k} \lambda_k(s_j) g(x_{k+1}, s_j) \geq 0 \end{aligned} \tag{2.3.33}$$

where the last equality is due to $\sum_{s_j \in \mathcal{R}_k} \lambda_k(s_j) g(x_k, s_j) = 0$. In particular,

$$v_{k+1} - v_k > 0. \tag{2.3.34}$$

The following result is a consequence of Theorem 3.1 in (Zhang et al. [2010a]).

Theorem 2.3.1 *The algorithm (PEA) terminates in a finite number of iterations.*

Convergence

Our main convergence result is the following theorem.

Theorem 2.3.2 *For given $\eta > 0$, Let x^*_η be the point determinated by the algorithm (PEA) in a finite number of iterations. Then*

(1). Every accumulation point of $\{x^*_\eta, \eta \to 0\}$ is an optimal solution of (P).

(2). $\lim_{\eta \to 0} f_{\epsilon(\eta)}(x^*_\eta) = v^*$, where v^* is the optimal value of (P).

(3). For any $\eta > 0$,

$$0 \le v^* - f_{\epsilon(\eta)}(x^*_\eta) \le M_1(\eta) \mathrm{dist}\left(\mathcal{F} \cap \{f \le \alpha\}, \mathcal{F}^\eta \cap \{f \le \alpha + \eta\}\right),$$

where $\alpha \ge v^*$,

$$\mathcal{F}^\eta := \{x; g(s,x) \le 0 \text{ for all } s \in \Omega_0 \text{ and } g(x,s) \le \eta \text{ for all } s \in \Omega\},$$

$$\mathcal{F} = \{x; g(s,x) \le 0 \text{ for all } s \in \Omega\},$$

$$M_1(\eta) := \sup_{x \in \mathcal{F}^\eta \cap \{f \le \alpha + \eta\}} \left(\|\nabla f(x)\| + 2\epsilon(\eta)\|x\|\right),$$

and $\mathrm{dist}(A, B) := \max_{x \in B} \min_{y \in A} \|x - y\|$ *for compact set $A \subset B$.*

(4). For any $\eta > 0$,

$$0 \le v^* - f_{\epsilon(\eta)}(x_\eta^*) \le \frac{\eta M_1(\eta)}{\eta + \rho}\|x_\eta^* - \hat{x}\|,$$

where $\hat{x}$ *is a point such that* $g(\hat{x}, s) < 0$ *for all* $s \in \Omega$, $\rho := -\max_{s\in\Omega} g(\hat{x}, s)$, *and* $\alpha \ge \max\{v^*, f(\hat{x})\}$.

Proof. Since the set $\{x_\eta^*, \eta > 0\} \subset \mathcal{F}_\lambda^0$ is bounded. Therefore, there exists at least an accumulation point x^* of $\{x_\eta^*, \eta \to 0\}$. By $g(x_\eta^*, s) \le \eta$ for all $s \in \Omega$, we have $x^* \in \mathcal{F}$. It is clear that there exists finite positive integer $N = N_\eta$, such that x_η^* is an optimal solution of the problem

$$\begin{cases} \min\limits_{x\in\mathbb{R}^n} f_{\epsilon(\eta)}(x) \\ s.t. \quad g(x,s) \le 0 \text{ for all } s \in \mathcal{R}_N \text{ and } g(x,s) \le \eta \text{ for all } s \in \Omega\backslash\mathcal{R}_N. \end{cases}$$

Therefore, if x is such that $g(x, s) \le 0$ for all $s \in \mathcal{R}_N$ and $g(x, s) \le \eta$ for all $s \in \Omega\backslash\mathcal{R}_N$, then

$$f_{\epsilon(\eta)}(x_\eta^*) \le f_{\epsilon(\eta)}(x).$$

For any subsequence $\eta_n \to 0$ such that $x_{\eta_n}^* \to x^*$, Since $\mathcal{F}_{f(x^*)}^0 \supset \{x; f(x) \le f(x^*)\} \cap \mathcal{F}$ and $\{x_{\eta_n}^*, n \ge 1\}$ are bounded, we have that as $\eta \to 0$,

$$\sup_{x\in\mathcal{F}_{f(x^*)}^0} |f_{\epsilon(\eta)}(x) - f(x)| \le \epsilon(\eta) \sup_{x\in\mathcal{F}_{f(x^*)}^0} \|x\|^2 \to 0,$$

and as $n \to \infty$,

$$|f_{\epsilon(\eta_n)}(x_{\eta_n}^*) - f(x^*)| \le |f(x_{\eta_n}^*) - f(x^*)| + \epsilon(\eta_n) \sup_{k\ge 1} \|x_{\eta_k}^*\| \to 0.$$

Hence, $f_{\epsilon(\eta_n)}(x_{\eta_n}^*) \to f(x^*)$, and $f(x^*) \le f(x)$ on $\{x; f(x) \le f(x^*)\} \cap \mathcal{F}$. Noting that $f(x^*) \le f(x)$ on $\{x; f(x) > f(x^*)\} \cap \mathcal{F}$, it follows that $f(x^*) \le f(x)$ for all $x \in \mathcal{F}$. Thus, x^* is an optimal solution of (P), and $\lim_{\eta\to 0} f_{\epsilon(\eta)}(x_\eta^*) = v^*$. (1) and (2) are valid.

Next, let us show (3). Let $\hat{x}^*_\eta$ be the orthogonal projection of x^*_η onto $\mathcal{F} \cap \{f \leq \alpha\}$. Then $f_{\epsilon(\eta)}(\hat{x}^*_\eta) \geq v^*$ and

$$\begin{aligned}
0 \leq & v^* - f_{\epsilon(\eta)}(x^*_\eta) \\
= & v^* - f_{\epsilon(\eta)}(\hat{x}^*_\eta) + f_{\epsilon(\eta)}(\hat{x}^*_\eta) - f_{\epsilon(\eta)}(x^*_\eta) \\
\leq & f_{\epsilon(\eta)}(\hat{x}^*_\eta) - f_{\epsilon(\eta)}(x^*_\eta) \\
= & \nabla f_{\epsilon(\eta)}(\tilde{x}^*_\eta)(\hat{x}^*_\eta - x^*_\eta) \\
\leq & \left(\|\nabla f(\tilde{x}^*_\eta)\| + 2\epsilon(\eta)\|\tilde{x}^*_\eta\|\right) \|\hat{x}^*_\eta - x^*_\eta\|
\end{aligned}$$

where $\tilde{x}^*_\eta$ is a point of the segment determined by $\hat{x}^*_\eta$ and x^*_η. Noting that $f(x^*_\eta) \leq f_{\epsilon(\eta)}(x^*_\eta) \leq \inf_{x \in \mathcal{F}}\{f(x) + \epsilon(\eta)\|x\|^2\} \leq v^* + \eta$, we have that $\tilde{x}^*_\eta \in \mathcal{F}^\eta \cap \{f \leq \alpha + \eta\}$ which is a compact set. Therefore, (3) is valid.

Finally, we prove (4). It is obvious that

$$\begin{aligned}
g\left(\frac{\rho}{\eta+\rho}x^*_\eta + \frac{\eta}{\eta+\rho}\hat{x}, s\right) \leq & \frac{\rho}{\eta+\rho}g(x^*_\eta, s) + \frac{\eta}{\eta+\rho}g(\hat{x}, s) \\
\leq & \frac{\rho}{\eta+\rho} \times \eta + \frac{\eta}{\eta+\rho} \times (-\rho) = 0,
\end{aligned}$$

and so $\hat{z}^*_\eta := \frac{\rho}{\eta+\rho}x^*_\eta + \frac{\eta}{\eta+\rho}\hat{x} \in \mathcal{F}$. Then

$$\begin{aligned}
0 \leq & v^* - f_{\epsilon(\eta)}(x^*_\eta) \\
= & v^* - f_{\epsilon(\eta)}(\hat{z}^*_\eta) + f_{\epsilon(\eta)}(\hat{z}^*_\eta) - f_{\epsilon(\eta)}(x^*_\eta) \\
\leq & f_{\epsilon(\eta)}(\hat{z}^*_\eta) - f_{\epsilon(\eta)}(x^*_\eta) \\
= & \nabla f_{\epsilon(\eta)}(\tilde{z}^*_\eta)(\hat{z}^*_\eta - x^*_\eta) \\
\leq & \left(\|\nabla f(\tilde{z}^*_\eta)\| + 2\epsilon(\eta)\|\tilde{z}^*_\eta\|\right) \|\hat{z}^*_\eta - x^*_\eta\|
\end{aligned}$$

where $\tilde{z}^*_\eta$ is a point of the segment determined by $\hat{z}^*_\eta$ and x^*_η, and so, $\tilde{z}^*_\eta \in \mathcal{F}^\eta \cap \{f \leq \alpha + \eta\}$ with $\alpha \geq \max\{v^*, f(\hat{x})\}$. Noting that $\|\hat{z}^*_\eta - x^*_\eta\| \leq \frac{\eta}{\eta+\rho}\|\hat{x} - x^*_\eta\|$, we obtain (4).

■

Remark 2.3.3 *(1). If (P) has a unique optimal solution, denoted by x^*, then by Theorem 2.3.2 (1), $\lim_{\eta\to 0} x^*_\eta = x^*$, and $\lim_{\eta \to 0} f_\eta(x^*_\eta) = f(x^*)$. Therefore, the* *perturbation algorithm (PEA) provides an approximate optimal solution to (P) after finitely many iterations.*

(2). Theorem 2.3.2 (3) and (4) provide error bounds for the approximate optimal solution x^*_η of (P).

(3). Since

$$\mathcal{F} \cap \{f \leq \alpha\} \subset \mathcal{F}^\eta \cap \{f \leq \alpha + \eta\} \subset \mathcal{F}^0_{\alpha+\eta},$$

it is obvious that

$$\operatorname{dist}\left(\mathcal{F} \cap \{f \leq \alpha\}, \mathcal{F}^\eta \cap \{f \leq \alpha + \eta\}\right) \leq \sup_{x,y \in \mathcal{F}^0_{\alpha+\eta}} \|x - y\|,$$

and

$$M_1(\eta) = \sup_{x \in \mathcal{F}^\eta \cap \{f \leq \alpha+\eta\}} \left(\|\nabla f(x)\| + 2\epsilon(\eta)\|x\|\right) \leq \sup_{x \in \mathcal{F}^0_{\alpha+\eta}} \left(\|\nabla f(x)\| + 2\epsilon(\eta)\|x\|\right).$$

2.3.3 Multiple exchange algorithm for beamforming problem

Algorithm

We consider a perturbation of the Beamforming problem (BP) as follows:

$$\begin{cases} \min & V_\epsilon(\boldsymbol{z}) \\ \text{s.t.} & g(\boldsymbol{z}, (\boldsymbol{r}, f)) \leq 0 \quad \forall (\boldsymbol{r}, f) \in \Omega, \end{cases} \tag{BP_ϵ}$$

where

$$V_\epsilon(\boldsymbol{z}) = V(\boldsymbol{z}) + \epsilon\|\boldsymbol{z}\|^2. \tag{2.3.35}$$

Assumption: There exists a finite subset Ω_0 such that for every $\lambda \in \mathbb{R}$, the set

$$\mathcal{F}_\lambda := \{\boldsymbol{z} \in \mathbb{R}^{n+2}; V(\boldsymbol{z}) \leq \lambda, g(\boldsymbol{z}, (\boldsymbol{r}, f)) \leq 0, \forall (\boldsymbol{r}, f) \in \Omega\}$$

is bounded when nonempty.

For given finite set $\mathcal{R} = \mathcal{R}^1 \cup \mathcal{R}^2 \cup \mathcal{R}^3 \cup \mathcal{R}^4 = \{(\boldsymbol{r}_j, f_j), j = 1, \cdots, m\} \subset \Omega$, we consider the finitely constrained convex programming problem:

$$\begin{cases} \min \quad V(\boldsymbol{z}) + \epsilon \|\boldsymbol{z}\|^2 \\ \text{s.t.} \quad g(\boldsymbol{z}, (\boldsymbol{r}_j, f_j)) \leq 0 \text{ for any } j = 1, \cdots, m. \end{cases} \tag{$BP_\epsilon(\mathcal{R})$}$$

For given a small $\eta \in (0, 1/2)$, choose $\epsilon = \epsilon(\eta) > 0$ small enough such that

$$\epsilon \sup_{\boldsymbol{z} \in \mathcal{F}_{\bar{\lambda}}} \|\boldsymbol{z}\|^2 \leq \eta,$$

where $\bar{\lambda} = 1 + f(\bar{\boldsymbol{z}})$ for some given $\bar{\boldsymbol{z}} \in \mathcal{F}$. We describe the perturbation exchange algorithm (PEA) for beamforming problem as follows.

Step 0. Choose a finite reference set $\mathcal{R}_0 = \{(\boldsymbol{r}_j^0, f_j^0), j = 1, \cdots, m_0\} \subset \Omega$ such that $\Omega_0 \subset \mathcal{R}_0$, and f_ϵ is level bounded on the feasible set of $(P_\epsilon(\mathcal{R}_0))$. Let $\boldsymbol{z}_0$ be an optimal solution to $(P_\epsilon(\mathcal{R}_0))$ and let $\{\lambda(\boldsymbol{r}_j^0, f_j^0), j = 1, \cdots, m_0\} \in \mathbb{R}^m$ be the set of associated multipliers. Set $k = 0$.

Step 1. Find a set $\{(\boldsymbol{r}_{new}^k, f_{new}^k), new = 1, \cdots, n\} \subset \Omega$ such that

$$g_i(\boldsymbol{z}_k, (\boldsymbol{r}_{new}^k, f_{new}^k)) > \eta.$$

If such a point does not exist, then stop. Otherwise, put $\bar{\mathcal{R}}_{k+1}^i = \mathcal{R}_k^i \cup \{(\boldsymbol{r}_{new}^k, f_{new}^k)\}$.

Step 2. Let $\boldsymbol{z}_{k+1}$ be an optimal solution to $(P_\epsilon(\bar{\mathcal{R}}_{k+1}))$ and let $\{\lambda_{k+1}(\boldsymbol{r}, f), (\boldsymbol{r}, f) \in \bar{\mathcal{R}}_{k+1}\}$ be the set of associated multipliers.

Step 3. Define new reference sets $\mathcal{R}_{k+1} = \bar{\mathcal{R}}_{k+1} \setminus \mathcal{R}_{k+1}^l$, where the leaving reference set

$$\mathcal{R}_{k+1}^l := \left\{(\boldsymbol{r}, f) \in \bar{\mathcal{R}}_{k+1}; \lambda_{k+1}(\boldsymbol{r}, f) = 0\right\}.$$

Set $k = k + 1$, and return to Step 1.

Error bounds

From Theorem 2.3.2 (3) and (4), we obtain error bounds for the approximate optimal solution $\boldsymbol{z}_\eta^*$ of the Beamforming problem.

For any $\eta > 0$ and $\alpha \geq v^*$,

$$0 \leq v^* - V_{\epsilon(\eta)}(\boldsymbol{z}_\eta^*) \leq M_1(\eta)\mathrm{dist}\left(\mathcal{F} \cap \{V \leq \alpha\}, \mathcal{F}^\eta \cap \{V \leq \alpha + \eta\}\right);$$

and for $\hat{\boldsymbol{z}}$ with $g(\hat{\boldsymbol{z}}, (\boldsymbol{r}, f)) < 0$ for all $(\boldsymbol{r}, f) \in \Omega$, for any $\eta > 0$ and $\alpha \geq \max\{v^*, f(\hat{\boldsymbol{z}})\}$,

$$0 \leq v^* - V_{\epsilon(\eta)}(\boldsymbol{z}_\eta^*) \leq \frac{\eta M_1(\eta)}{\eta + \rho}\|\boldsymbol{z}_\eta^* - \hat{\boldsymbol{z}}\|,$$

where $\rho := -\max_{(\boldsymbol{r},f)\in\Omega} g(\hat{\boldsymbol{z}}, (\boldsymbol{r}, f))$, and

$$M_1(\eta) := \sqrt{\phi_1^2 + \phi_2^2} + 2\epsilon(\eta) \sup_{\boldsymbol{z}\in\mathcal{F}^0_{\alpha+\eta}} \|\boldsymbol{z}\|.$$

In particular, if $\alpha = v^*$, then $\mathcal{F}^0_{\alpha+\eta} \subset \mathcal{F}^0_{\bar{\lambda}}$, thus

$$M_1(\eta) \leq \sqrt{\phi_1^2 + \phi_2^2} + 2\eta,$$

and

$$\mathrm{dist}\left(\mathcal{F} \cap \{f \leq \alpha\}, \mathcal{F}^\eta \cap \{f \leq \alpha + \eta\}\right) \leq \sup_{x,y\in\mathcal{F}^0_{\bar{\lambda}}} \|x - y\|.$$

Illustrative examples

In this section we give one example to demonstrate the performance of the algorithm for non-strictly convex example and apply the proposed algorithm to solve the broadband beamformer design problem. The quadprog function in Matlab is used to solve.

The same as the previous illustrative examples in the last chapter, we choose the desired response function as

$$G_d(\boldsymbol{r}, f) = \begin{cases} e^{-j2\pi f\left(\frac{||r-r_c||}{c} + \frac{L-1}{2}T\right)}, & \text{if } (\boldsymbol{r}, f) \text{ is in passband region} \\ 0, & \text{if } (\boldsymbol{r}, f) \text{ is in stopband region} \end{cases},$$

where $\boldsymbol{r}_c$ is the coordinate for the center element, the sound speed is $c = 340.9m/s$ and the sample increment is $T = 125\mu s$, that is, the sampling rate is set as 8kHz.

In this example, we also consider an equispaced linear array with five elements with coordinates $\{(-0.1, 0), (-0.05, 0), \ldots, (0.1, 0)\}$. A seven-tap FIR filter behind each element is used. The passband region and stopband region are specified on an x-axis parallel with, and $y = 1$ meter in front of, the array. The passband region is defined as

$$\{(\boldsymbol{r}, f) : -0.4m \leq x \leq 0.4m,\ y = 1m,\ 0.5kHz \leq f \leq 1.5kHz\}$$

while the stopband region is the union of several parts as

$$\{(\boldsymbol{r}, f) : -0.4m \leq x \leq 0.4m,\ y = 1m,\ 2.5kHz \leq f \leq 4kHz\},$$

$$\{(\boldsymbol{r}, f) : 1.5m \leq |x| \leq 2.5m,\ y = 1m,\ 0.5kHz \leq f \leq 1.5kHz\},$$

$$\{(\boldsymbol{r}, f) : 1.5m \leq |x| \leq 2.5m,\ y = 1m,\ 2.5kHz \leq f \leq 4kHz\}.$$

The amplitude of the actual response $G(\boldsymbol{r}, f)$ is shown in Figure 2.17.

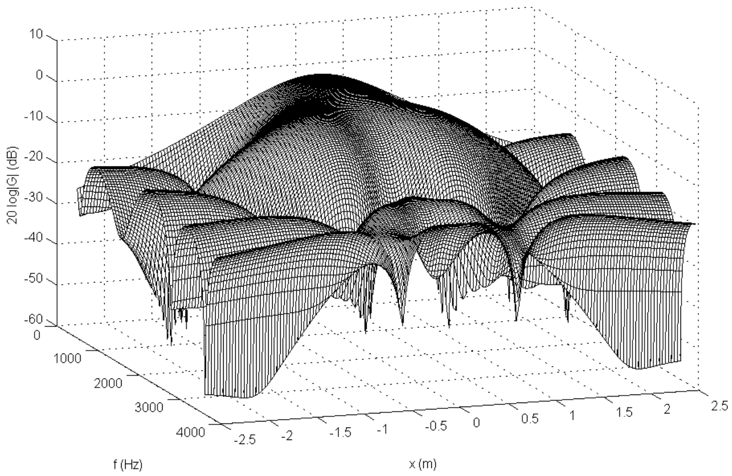

Figure 2.17: Amplitude of $G(\boldsymbol{r}, f)$ where $N = 5$, $L = 7$ and $y = 1m$.

Chapter 3

A mixed SDP-SOCP relaxation model for source localization problem with time-difference information and sensor network localization

3.1 Introduction

In this chapter, we will apply the SDP method for solving the source localization problem. The basic idea behind the technique comes from (Biswas et al. [2006a]), that is to convert the nonconvex quadratic distance constraints into linear constraints by introducing a relaxation to remove the quadratic term in the formulation. Here we propose a novel idea, combining the second order cone programming (SOCP), then a mixed SDP-SOCP relaxation model is expressed for source localization problem with time-difference information. We present a method to obtain the exact solution for the source localization. The method shows that the exact solution for the source localization can be obtained from the solutions of the mixed SDP-SOCP relaxation and the second order polynomial equation. We also give the estimator properties for the true source location under noises and present a bi-level method. In (Tseng [2007]), we know for single SOCP relaxation, the optimal solution must be in the convex hull of microphone array. In the next chapter, we will study some properties for the solution of the mixed SDP-SOCP and present a characterization such that the mixed SDP-SOCP has an exact relaxation. The characterization shows that the exact solution region is not only the convex hull of microphone array, but also outside the convex hull. The source localization problem can be combined with the sensor network localization problem, we also give a mixed SDP-SOCP relaxation model for it and give some statistical analyses.

The chapter is organized as follows. In Section 2, we formulate a mixed SDP-SOCP relaxation model for source localization problem with time-difference information. We present a method to obtain the exact solution for the source localization from the solutions of the mixed SDP-SOCP relaxation and the second order polynomial equation. The estimator properties for the true source location under noises is studied. In section 3, we present the bi-level method. A mixed SDP-SOCP relaxation model for source localization combined with sensor network localization problem is studied in section 4, also we give some statistical analyses for it.

3.2 Mixed SDP-SOCP relaxation model for source localization problem with time-difference information

3.2.1 Convex relaxation models

Assume we have microphones $\boldsymbol{a}_i = (a_{i1}, a_{i2}, \cdots, a_{id})^T$, $i = 1, \cdots, m$, whose locations are known. Given a true source location $\mathbf{s}$, we can derive the true time difference of arrival(TDOA)

$$T(\{\boldsymbol{a}_i, \boldsymbol{a}_j\}, \mathbf{s}) = \frac{\|\mathbf{s} - \boldsymbol{a}_i\| - \|\mathbf{s} - \boldsymbol{a}_j\|}{c_0}, \quad i, j = 1, \cdots, m, \tag{3.2.1}$$

where c_0 is the speed of sound in the air. The estimated TDOA will be given by $\hat{\tau}_{ij}$ using the signals received at the two microphones. The source localization problem is to estimate the source location $\hat{\mathbf{s}}$ from using these nonlinear hyperbolic equations. In this section, we will introduce four relaxation models for solving this source localization problem.

SDP relaxation model

With a set of delay estimates, the problem is to find $\hat{\mathbf{s}}$ such that

$$\|\mathbf{s} - \boldsymbol{a}_i\| - \|\mathbf{s} - \boldsymbol{a}_j\| = c_0 \hat{\tau}_{ij}, \qquad i, j = 1, \cdots, m. \tag{3.2.2}$$

These equations also can be expressed as

$$\begin{cases} \beta_i - \beta_j = c_0 \hat{\tau}_{ij}, \qquad i, j = 1, \cdots, m, \\ \|\mathbf{s} - \boldsymbol{a}_i\|^2 = \alpha_i, \qquad i = 1, \cdots, m, \\ \alpha_i = \beta_i^2, \qquad i = 1, \cdots, m, \\ \beta_i \geq 0, \qquad i = 1, \cdots, m. \end{cases} \tag{3.2.3}$$

We relax the equality constraints $\alpha_i = \beta_i^2$ to $\begin{pmatrix} 1 & \beta_i \\ \beta_i & \alpha_i \end{pmatrix} \succeq 0$ which ensure that $\alpha_i \geq \beta_i^2$ and transform the source localization problem into

$$\begin{cases} \min \ \delta \sum_{i=1}^{m} \alpha_i \\ \text{s.t.} \ \ \beta_i - \beta_j = c_0 \hat{\tau}_{ij}, \quad i,j = 1, \cdots, m, \\ \qquad \|\mathbf{s} - \boldsymbol{a}_i\|^2 = \alpha_i, \quad i = 1, \cdots, m, \\ \qquad \begin{pmatrix} 1 & \beta_i \\ \beta_i & \alpha_i \end{pmatrix} \succeq 0, \qquad i = 1, \cdots, m, \\ \quad \beta_i \geq 0, \qquad i = 1, \cdots, m, \end{cases} \tag{3.2.4}$$

where δ is a positive constant for penalization.

Noting that $\|\boldsymbol{a}_i - \mathbf{s}\|^2 = \begin{pmatrix} \boldsymbol{a}_i^T & -1 \end{pmatrix} \begin{pmatrix} \boldsymbol{I} \\ \mathbf{s}^T \end{pmatrix} [\boldsymbol{I} \ \mathbf{s}] \begin{pmatrix} \boldsymbol{a}_i \\ -1 \end{pmatrix}$. Let $\boldsymbol{Y} = \mathbf{s}^T\mathbf{s}$. Then the above problem is equivalent to

$$\begin{cases} \min \ \delta \sum_{i=1}^{m} \alpha_i \\ \text{s.t.} \ \ \beta_i - \beta_j = c_0 \hat{\tau}_{ij}, \quad i,j = 1, \cdots, m, \\ \qquad \begin{pmatrix} \boldsymbol{a}_i^T & -1 \end{pmatrix} \begin{pmatrix} \boldsymbol{I} & \mathbf{s} \\ \mathbf{s}^T & \boldsymbol{Y} \end{pmatrix} \begin{pmatrix} \boldsymbol{a}_i \\ -1 \end{pmatrix} = \alpha_i, \quad i = 1, \cdots, m, \\ \qquad \begin{pmatrix} 1 & \beta_i \\ \beta_i & \alpha_i \end{pmatrix} \succeq 0, \quad i = 1, \cdots, m, \\ \qquad \beta_i \geq 0, \qquad i = 1, \cdots, m, \\ \qquad \boldsymbol{Y} = \mathbf{s}^T\mathbf{s}. \end{cases} \tag{3.2.5}$$

An effective method for solving this problem is to relax $\boldsymbol{Y} = \mathbf{s}^T\mathbf{s}$ to $\boldsymbol{Y} \succeq \mathbf{s}^T\mathbf{s}$ which is equivalent to

$$\boldsymbol{Z} := \begin{pmatrix} \boldsymbol{I} & \mathbf{s} \\ \mathbf{s}^T & \boldsymbol{Y} \end{pmatrix} \succeq 0.$$

Then, the relaxed version of the problem (3.2.5) can be represented as the following mixed SDP relaxation model

$$\begin{cases} \min \ \delta \sum_{i=1}^{m} \alpha_i \\ \text{s.t.} \ \beta_i - \beta_j = c_0 \hat{\tau}_{ij}, \quad i, j = 1, \cdots, m, \\ \begin{pmatrix} \boldsymbol{a}_i^T & -1 \end{pmatrix} \boldsymbol{Z} \begin{pmatrix} \boldsymbol{a}_i \\ -1 \end{pmatrix} = \alpha_i, \quad i = 1, \cdots, m, \\ \begin{pmatrix} 1 & \beta_i \\ \beta_i & \alpha_i \end{pmatrix} \succeq 0, \quad i = 1, \cdots, m, \\ \beta_i \geq 0, \qquad i = 1, \cdots, m, \\ \boldsymbol{Z}_{1:d,1:d} = I_d, \boldsymbol{Z} \succeq 0, \mathbf{s} = \boldsymbol{Z}_{1:d,d+1}. \end{cases} \tag{3.2.6}$$

SOCP relaxation model

For equation (3.2.2), instead of (3.2.3), it also can be expressed as

$$\begin{cases} \beta_i - \beta_j = c_0 \hat{\tau}_{ij}, \qquad i, j = 1, \cdots, m, \\ \|\mathbf{s} - \boldsymbol{a}_i\| = \beta_i, \qquad i = 1, \cdots, m. \end{cases} \tag{3.2.7}$$

Similar to the method in (Tseng [2007]), we relax the equality constraints $\|\mathbf{s} - \boldsymbol{a}_i\| = \beta_i$ to " $\leq$ " inequality constraints, which yields a second order cone problem. Then we transform the source localization problem into the following SOCP relaxation model

$$\begin{cases} \min \ \delta \sum_{i=1}^{m} \beta_i \\ \text{s.t.} \ \beta_i - \beta_j = c_0 \hat{\tau}_{ij}, \quad i, j = 1, \cdots, m, \\ \qquad \|\mathbf{s} - \boldsymbol{a}_i\| \leq \beta_i, \quad i = 1, \cdots, m. \end{cases} \tag{3.2.8}$$

where δ is a positive constant for penalization.

YWL's model

In this section, we present another convex relaxation model introduced in (Yang et al. [2009a]). They use a maximum likelihood formulation

$$\min \sum_{j=1}^{m} \sum_{i=1, i\neq j}^{m} (\frac{1}{c_0}\|\mathbf{s}-\boldsymbol{a}_i\| - \frac{1}{c_0}\|\mathbf{s}-\boldsymbol{a}_j\| - \hat{\tau}_{ij})^2. \tag{3.2.9}$$

which can be equivalently written as

$$\begin{cases} \min & \sum_{j=1}^{m} \sum_{i=1, i\neq j}^{m} (t_i - t_j - \hat{\tau}_{ij})^2 \\ \text{s.t.} & \frac{1}{c_0}\|\mathbf{s}-\boldsymbol{a}_i\| = t_i, \quad i = 1, \cdots, m. \end{cases} \tag{3.2.10}$$

They transform the objective function of (3.2.10) as θ and notice that

$$\theta = \|\bar{\boldsymbol{t}} - \tilde{\boldsymbol{\tau}}\|^2 + \|\bar{\boldsymbol{t}} + \bar{\boldsymbol{\tau}}\|^2 \tag{3.2.11}$$

where

$$\bar{\boldsymbol{t}} = (t_1 - t_2, \cdots, t_1 - t_m, t_2 - t_3, \cdots, t_2 - t_m, \cdots, t_{m-1} - t_m)^T = \boldsymbol{G}\boldsymbol{t}$$

$$\boldsymbol{t} = (t_1, \cdots, t_m)^T$$

$$\boldsymbol{G} = \begin{pmatrix} 1 & -1 & 0 & \cdots & \cdots & 0 \\ \vdots & & \ddots & \ddots & & \vdots \\ 1 & 0 & \cdots & \cdots & 0 & -1 \\ 0 & 1 & -1 & 0 & \cdots & 0 \\ \vdots & \vdots & & \ddots & \ddots & \vdots \\ 0 & 1 & 0 & \cdots & 0 & -1 \\ \vdots & \vdots & & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 & 1 & -1 \end{pmatrix} \tag{3.2.12}$$

$$\tilde{\boldsymbol{\tau}} = (\tau_{12}, \cdots, \tau_{1m}, \tau_{23}, \cdots, \tau_{2m}, \cdots, \tau_{m-1,m})^T$$

$$\bar{\boldsymbol{\tau}} = (\tau_{21}, \cdots, \tau_{m1}, \tau_{32}, \cdots, \tau_{m2}, \cdots, \tau_{m,m-1})^T.$$

Then the objective function of (3.2.10) can be written as

$$\begin{aligned}\theta &= \|\bar{\boldsymbol{t}} - \tilde{\boldsymbol{\tau}}\|^2 + \|\bar{\boldsymbol{t}} + \bar{\boldsymbol{\tau}}\|^2 \\ &= \operatorname{tr}\left\{\begin{pmatrix}\boldsymbol{t} \\ 1\end{pmatrix}\begin{pmatrix}\boldsymbol{t}^T & 1\end{pmatrix}\boldsymbol{F}\right\} \\ &= \operatorname{tr}\left\{\begin{pmatrix}\boldsymbol{T} & \boldsymbol{t} \\ \boldsymbol{t}^T & 1\end{pmatrix}\boldsymbol{F}\right\}\end{aligned} \tag{3.2.13}$$

where

$$\begin{aligned}\boldsymbol{T} &= \boldsymbol{t}\boldsymbol{t}^T \\ \boldsymbol{F} &= \begin{pmatrix} 2\boldsymbol{G}^T\boldsymbol{G} & \boldsymbol{G}^T(\bar{\boldsymbol{\tau}} - \tilde{\boldsymbol{\tau}}) \\ (\bar{\boldsymbol{\tau}} - \tilde{\boldsymbol{\tau}})^T\boldsymbol{G} & \bar{\boldsymbol{\tau}}^T\bar{\boldsymbol{\tau}} + \tilde{\boldsymbol{\tau}}^T\tilde{\boldsymbol{\tau}}\end{pmatrix}\end{aligned} \tag{3.2.14}$$

The constraints $\frac{1}{c_0}\|\mathbf{s} - \boldsymbol{a}_i\| = t_i$ can be represented as

$$\begin{aligned}\boldsymbol{T}_{ii} &= t_i^2 \\ &= \frac{1}{c_0^2}\|\mathbf{s} - \boldsymbol{a}_i\|^2 \\ &= \frac{1}{c_0^2}\begin{pmatrix}\boldsymbol{a}_i^T & -1\end{pmatrix}\begin{pmatrix} I & \mathbf{s} \\ \mathbf{s}^T & z\end{pmatrix}\begin{pmatrix}\boldsymbol{a}_i \\ -1\end{pmatrix},\end{aligned} \tag{3.2.15}$$

where $z = \mathbf{s}^T\mathbf{s}$.

By Cauchy-Schwartz inequality,

$$\begin{aligned}\boldsymbol{T}_{ij} &= t_i t_j \\ &= \frac{1}{c_0^2}\|\mathbf{s} - \boldsymbol{a}_i\|\|\mathbf{s} - \boldsymbol{a}_j\| \\ &\geq \frac{1}{c_0^2}|(\mathbf{s} - \boldsymbol{a}_i)^T(\mathbf{s} - \boldsymbol{a}_j)|\end{aligned} \tag{3.2.16}$$

where

$$(\mathbf{s} - \boldsymbol{a}_i)^T(\mathbf{s} - \boldsymbol{a}_j) = \begin{pmatrix}\boldsymbol{a}_i^T & -1\end{pmatrix}\begin{pmatrix} I & \mathbf{s} \\ \mathbf{s}^T & z\end{pmatrix}\begin{pmatrix}\boldsymbol{a}_j \\ -1\end{pmatrix} \tag{3.2.17}$$

Using the above argument together with the following relaxations:

$$
\begin{aligned}
\mathbf{s}^T\mathbf{s} \preceq z \Leftrightarrow \begin{pmatrix} I & \mathbf{s} \\ \mathbf{s}^T & z \end{pmatrix} \succeq 0 \\
\begin{pmatrix} \boldsymbol{t} \\ 1 \end{pmatrix} \begin{pmatrix} \boldsymbol{t}^T & 1 \end{pmatrix} \succeq 0 \Rightarrow \begin{pmatrix} \boldsymbol{T} & \boldsymbol{t} \\ \boldsymbol{t}^T & 1 \end{pmatrix} \succeq 0
\end{aligned}
\tag{3.2.18}
$$

then (3.2.10) can be cast into the following SDP model

$$
\begin{cases}
\min \ \operatorname{tr}\left\{\begin{pmatrix} \boldsymbol{T} & \boldsymbol{t} \\ \boldsymbol{t}^T & 1 \end{pmatrix}\right\}\boldsymbol{F} + \delta \sum_{i=1}^{m}\sum_{j=1}^{m} \boldsymbol{T}_{ij} \\
\text{s.t. } \boldsymbol{T}_{ii} = \frac{1}{c_0^2}\begin{pmatrix} \boldsymbol{a}_i^T & -1 \end{pmatrix}\begin{pmatrix} I & \mathbf{s} \\ \mathbf{s}^T & z \end{pmatrix}\begin{pmatrix} \boldsymbol{a}_i \\ -1 \end{pmatrix} \\
\qquad \boldsymbol{T}_{ij} \geq \frac{1}{c_0^2}\left|\begin{pmatrix} \boldsymbol{a}_i^T & -1 \end{pmatrix}\begin{pmatrix} I & \mathbf{s} \\ \mathbf{s}^T & z \end{pmatrix}\begin{pmatrix} \boldsymbol{a}_j \\ -1 \end{pmatrix}\right| \\
\qquad \begin{pmatrix} \boldsymbol{T} & \boldsymbol{t} \\ \boldsymbol{t}^T & 1 \end{pmatrix} \succeq 0 \\
\qquad \begin{pmatrix} I & \mathbf{s} \\ \mathbf{s}^T & z \end{pmatrix} \succeq 0, \quad i, j = 1, \cdots, m, j > i
\end{cases}
\tag{3.2.19}
$$

where δ is a positive constant for penalization.

Mixed SDP-SOCP relaxation model

For equation (3.2.2), we expressed it as

$$
\begin{cases}
\beta_i - \beta_j = c_0\hat{\tau}_{ij}, \quad i, j = 1, \cdots, m, \\
\|\mathbf{s} - \boldsymbol{a}_i\| = \beta_i, \quad i = 1, \cdots, m.
\end{cases}
\tag{3.2.20}
$$

Adding some redundant constrains

$$\begin{cases} \beta_i - \beta_j = c_0\hat{\tau}_{ij}, \quad i,j = 1,\cdots,m, \\ \|\mathbf{s} - \boldsymbol{a}_i\| = \beta_i, \quad i = 1,\cdots,m, \\ \|\mathbf{s} - \boldsymbol{a}_i\|^2 = \alpha_i, \quad i = 1,\cdots,m, \\ \alpha_i = \beta_i^2, \quad i = 1,\cdots,m. \end{cases} \tag{3.2.21}$$

It is equivalent to the following problem

$$\begin{cases} \beta_i - \beta_j = c_0\hat{\tau}_{ij}, \quad i,j = 1,\cdots,m, \\ \|\mathbf{s} - \boldsymbol{a}_i\| = \beta_i, \quad i = 1,\cdots,m, \\ \|\mathbf{s} - \boldsymbol{a}_i\|^2 = \alpha_i, \quad i = 1,\cdots,m, \\ \alpha_i \geq \beta_i^2, \quad i = 1,\cdots,m, \\ \alpha_i \leq \beta_i^2, \quad i = 1,\cdots,m. \end{cases} \tag{3.2.22}$$

The inequality constraints $\alpha_i \geq \beta_i^2$ is equivalent to $\begin{pmatrix} 1 & \beta_i \\ \beta_i & \alpha_i \end{pmatrix} \succeq 0$, and $\alpha_i \leq \beta_i^2$ is equivalent to $\|\mathbf{s} - \boldsymbol{a}_i\| \leq \beta_i$.

Thus we relax the equality constraints $\|\mathbf{s} - \boldsymbol{a}_i\| = \beta_i$ to " $\leq$ " inequality constraints, which yields a second order cone problem. Then we transform the source localization problem into

$$\begin{cases} \min \ \delta \sum_{i=1}^{m} \alpha_i \\ \text{s.t.} \ \beta_i - \beta_j = c_0\hat{\tau}_{ij}, \quad i,j = 1,\cdots,m, \\ \qquad \|\mathbf{s} - \boldsymbol{a}_i\| \leq \beta_i, \quad i = 1,\cdots,m, \\ \qquad \|\mathbf{s} - \boldsymbol{a}_i\|^2 = \alpha_i, \quad i = 1,\cdots,m, \\ \qquad \begin{pmatrix} 1 & \beta_i \\ \beta_i & \alpha_i \end{pmatrix} \succeq 0, \quad i = 1,\cdots,m, \end{cases} \tag{3.2.23}$$

where δ is a positive constant for penalization.

Then using a similar method as described in previous section, the above problem is equivalent to

$$\begin{cases} \min \ \delta \sum_{i=1}^{m} \alpha_i \\ \text{s.t. } \ \beta_i - \beta_j = c_0 \hat{\tau}_{ij}, \quad i,j = 1, \cdots, m, \\ \qquad \|\mathbf{s} - \boldsymbol{a}_i\| \leq \beta_i, \quad i = 1, \cdots, m, \\ \qquad \begin{pmatrix} \boldsymbol{a}_i^T & -1 \end{pmatrix} \begin{pmatrix} \boldsymbol{I} & \mathbf{s} \\ \mathbf{s}^T & \boldsymbol{Y} \end{pmatrix} \begin{pmatrix} \boldsymbol{a}_i \\ -1 \end{pmatrix} = \alpha_i, \quad i = 1, \cdots, m, \\ \qquad \begin{pmatrix} 1 & \beta_i \\ \beta_i & \alpha_i \end{pmatrix} \succeq 0, \quad i = 1, \cdots, m, \\ \qquad \boldsymbol{Y} = \mathbf{s}^T \mathbf{s}. \end{cases} \tag{3.2.24}$$

The relaxed version of the problem (3.2.24) can be represented as the following mixed SDP-SOCP relaxation model

$$\begin{cases} \min \ \delta \sum_{i=1}^{m} \alpha_i \\ \text{s.t. } \ \beta_i - \beta_j = c_0 \hat{\tau}_{ij}, \quad i,j = 1, \cdots, m, \\ \qquad \|\mathbf{s} - \boldsymbol{a}_i\| \leq \beta_i, \quad i = 1, \cdots, m, \\ \qquad \begin{pmatrix} \boldsymbol{a}_i^T & -1 \end{pmatrix} \boldsymbol{Z} \begin{pmatrix} \boldsymbol{a}_i \\ -1 \end{pmatrix} = \alpha_i, \quad i = 1, \cdots, m, \\ \qquad \begin{pmatrix} 1 & \beta_i \\ \beta_i & \alpha_i \end{pmatrix} \succeq 0, \quad i = 1, \cdots, m, \\ \qquad \boldsymbol{Z}_{1:d,1:d} = I_d, \boldsymbol{Z} \succeq 0, \mathbf{s} = \boldsymbol{Z}_{1:d,d+1}. \end{cases} \tag{3.2.25}$$

It is obvious that the solution of (3.2.2) is a feasible solution for (3.2.25). The following question is natural and important:

(Q). Whether the solution of (3.2.2) can be obtained from the solutions of the mixed SDP-SOCP relaxation?

In next section, we present an approach how to obtain the exact solution for the source localization from the solutions of the mixed SDP-SOCP relaxation.

3.2.2 An error correction algorithm

In this section, we consider source localization with time-difference information of m points. By solving order polynomial equations, we obtain the exact solution for the source localization from the solution of the mixed SDP-SOCP relaxation model.

Let $\alpha = (\alpha_1, \alpha_2, \cdots, \alpha_m)^T$, $\beta = (\beta_1, \beta_2, \cdots, \beta_m)^T$, $\mathbf{s} = (s_1, s_2, \cdots, s_d)^T$. Let $\boldsymbol{a}_i = (a_{i1}, a_{i2}, \cdots, a_{id})^T$, $i = 1, 2, \cdots, m$, be m points in $\mathbb{R}^d$, where $m \geq d + 1$. Assume

$$\operatorname{rank}\{\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_m\} = d.$$

Then the mixed SDP-SOCP (3.2.25) can be written the equivalent form:

$$\begin{cases} \min & \alpha_1 + \alpha_2 + \cdots + \alpha_m \\ \text{s.t.} & \beta_i - \beta_j = c_0 \hat{\tau}_{ij}, \quad i, j = 1, 2, \cdots, m, \\ & \|\mathbf{s} - \boldsymbol{a}_i\|^2 + (y - \|\mathbf{s}\|^2) = \alpha_i, \quad i = 1, 2, \cdots, m, \\ & y \geq \|\mathbf{s}\|^2, \quad \alpha_i \geq \beta_i^2, \quad \beta_i \geq \|\mathbf{s} - \boldsymbol{a}_i\|, \quad i = 1, 2, \cdots, m. \end{cases} \tag{3.2.26}$$

For convenience, we use the notation $(\mathbf{s}, \beta, \alpha, y)$ to denote a feasible solution for (3.2.26).

Let $\mathbf{s}^* = (s_1^*, s_2^*, \cdots, s_d^*)^T$ be true source localization, i.e., it satisfies

$$\|\mathbf{s}^* - \boldsymbol{a}_i\| - |\mathbf{s}^* - \boldsymbol{a}_j\| = c_0 \hat{\tau}_{ij}, i, j = 1, 2, \cdots, m.$$

Denote by $\beta_i^* = \|\mathbf{s}^* - \boldsymbol{a}_i\|$, $i = 1, 2, \cdots, m$.

Lemma 3.2.1 *Let $(\hat{\mathbf{s}}, \hat{y}, \hat{\beta}, \hat{\alpha})$ be an optimal solution of (3.2.26). Then there exists $\lambda \geq 0$ such that*

$$\hat{\beta}_i = \beta_i^* - \lambda, \quad i = 1, 2, \cdots, m.$$

Proof. Set $\beta^* = (\beta_1^*, \beta_2^*, \cdots, \beta_m^*)^T$, $\alpha^* = ((\beta_1^*)^2, (\beta_2^*)^2, \cdots, (\beta_m^*)^2)^T$, and $y^* = \|\mathbf{s}^*\|^2$. Then $(\mathbf{s}^*, \beta^*, \alpha^*, y^*)$ is a feasible solution of the mixed SDP-SOCP (3.2.26). Since

$$\alpha_i^* = (\beta_i^*)^2 = \|\mathbf{s}^* - \boldsymbol{a}_i\|^2, \; i = 1, 2, \cdots, m$$

and $(\hat{\mathbf{s}}, \hat{y}, \hat{\beta}, \hat{\alpha})$ is an optimal solution of (3.2.26), then we have

$$\sum_{i=1}^m \hat{\beta}_i^2 \le \sum_{i=1}^m \hat{\alpha}_i \le \sum_{i=1}^m \alpha_i^* = \sum_{i=1}^m (\beta_i^*)^2$$

Notice that $\hat{\beta}_i, i = 1, \cdots, m$ and $\beta_i^*, i = 1, \cdots, m$ are the solution of the linear equations with m variables and d equations:

$$x_m - x_i = c_0 \hat{\tau}_{m\,i}, \quad i = 1, 2, \cdots, m-1,$$

and the rank of the coefficient matrix of the above linear equations is d, then there exists $\lambda \in \mathbb{R}$ such that

$$\hat{\beta}_i = \beta_i^* - \lambda, \quad i = 1, \cdots, m.$$

From

$$\sum_{i=1}^m (\beta_i^* - \lambda)^2 \le \sum_{i=1}^m (\beta_i^*)^2,$$

we have $\lambda > 0$. ■

Therefore, there exists $\lambda \ge 0$ such that $\beta^* = \hat{\beta} + \lambda$, i.e, $\mathbf{s}^*$ is a solution of the equations

$$\|\mathbf{s} - \boldsymbol{a}_i\| = \hat{\beta}_i + \lambda, i = 1, 2, \cdots, m. \tag{3.2.27}$$

This implies that

$$\|\mathbf{s} - \boldsymbol{a}_m\|^2 - \|\mathbf{s} - \boldsymbol{a}_i\|^2 = (\hat{\beta}_m + \lambda)^2 - (\hat{\beta}_i + \lambda)^2, i = 1, 2, \cdots, m-1.$$

That is

$$\begin{aligned} &\sum_{j=1}^d 2(a_{ij} - a_{mj}) s_j \\ =&\sum_{j=1}^d (a_{ij}^2 - a_{mj}^2) + (\hat{\beta}_m^2 - \hat{\beta}_i^2) + 2\lambda(\hat{\beta}_m - \hat{\beta}_i), i = 1, 2, \cdots, m-1. \end{aligned} \tag{3.2.28}$$

Define

$$\boldsymbol{B} = (b_{ij})_{(m-1)\times d}, \quad \boldsymbol{C} = (c_1, \cdots, c_{m-1})^T, \quad \boldsymbol{D} = (d_1, \cdots, d_{m-1})^T,$$

where

$$b_{ij} = 2(a_{ij} - a_{mj}), \quad i = 1, 2, \cdots, m-1, j = 1, \cdots, d,$$

and

$$c_i = \sum_{j=1}^{d}(a_{ij}^2 - a_{mj}^2) + (\hat{\beta}_m^2 - \hat{\beta}_i^2), \quad d_i = 2(\hat{\beta}_m - \hat{\beta}_i), \quad i = 1, 2, \cdots, m-1.$$

Then the equations (3.2.28) can be written by

$$\boldsymbol{B}\mathbf{s} = \boldsymbol{C} + \lambda\boldsymbol{D}, \tag{3.2.29}$$

which implies that

$$\boldsymbol{B}^T\boldsymbol{B}\mathbf{s} = \boldsymbol{B}^T\boldsymbol{C} + \lambda\boldsymbol{B}^T\boldsymbol{D}.$$

Therefore,

$$\mathbf{s} = (\boldsymbol{B}^T\boldsymbol{B})^{-1}\boldsymbol{B}^T\boldsymbol{C} + \lambda(\boldsymbol{B}^T\boldsymbol{B})^{-1}\boldsymbol{B}^T\boldsymbol{D}. \tag{3.2.30}$$

Substituting $(\boldsymbol{B}^T\boldsymbol{B})^{-1}\boldsymbol{B}^T\boldsymbol{C}+\lambda(\boldsymbol{B}^T\boldsymbol{B})^{-1}\boldsymbol{B}^T\boldsymbol{D}$ for $\mathbf{s}$ in $\|\mathbf{s}-\boldsymbol{a}_m\|^2 = (\hat{\beta}_m+\lambda)^2$, we obtain

$$\|(\boldsymbol{B}^T\boldsymbol{B})^{-1}\boldsymbol{B}^T\boldsymbol{C} + \lambda(\boldsymbol{B}^T\boldsymbol{B})^{-1}\boldsymbol{B}^T\boldsymbol{D} - \boldsymbol{a}_m\|^2 = (\hat{\beta}_m + \lambda)^2. \tag{3.2.31}$$

The above equation can be written the following form:

$$\begin{aligned}
&\lambda^2\left(1 - ((\boldsymbol{B}^T\boldsymbol{B})^{-1}\boldsymbol{B}^T\boldsymbol{D})^T(\boldsymbol{B}^T\boldsymbol{B})^{-1}\boldsymbol{B}^T\boldsymbol{D}\right)\\
&+\lambda\left(2\hat{\beta}_m - 2\left((\boldsymbol{B}^T\boldsymbol{B})^{-1}\boldsymbol{B}^T\boldsymbol{C} - \boldsymbol{a}_m\right)^T(\boldsymbol{B}^T\boldsymbol{B})^{-1}\boldsymbol{B}^T\boldsymbol{D}\right)\\
&+\hat{\beta}_m^2 - ((\boldsymbol{B}^T\boldsymbol{B})^{-1}\boldsymbol{B}^T\boldsymbol{C} - \boldsymbol{a}_m)^T((\boldsymbol{B}^T\boldsymbol{B})^{-1}\boldsymbol{B}^T\boldsymbol{C} - \boldsymbol{a}_m) = 0.
\end{aligned} \tag{3.2.32}$$

By solving the above second order polynomial equation, we can obtain the exact solution for the source localization from the solution of the mixed SDP-SOCP relaxation. That is, we have the following result.

Theorem 3.2.1 *Let* $\boldsymbol{a}_i = (a_{i1}, a_{i2}, \cdots, a_{id})^T$, $i = 1, 2, \cdots, m$, *be* m *points in* $\mathbb{R}^d$ *satisfying*

$$\operatorname{rank}\{\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_m\} = d.$$

Let $(\hat{\mathbf{s}}, \hat{y}, \hat{\beta}, \hat{\alpha})$ *be an optimal solution of (3.2.26) and let* $\hat{\lambda}$ *called error corrector be a positive solution of the equation (3.2.32). Define*

$$\mathbf{s}^* = (\boldsymbol{B}^T\boldsymbol{B})^{-1}\boldsymbol{B}^T\boldsymbol{C} + \hat{\lambda}(\boldsymbol{B}^T\boldsymbol{B})^{-1}\boldsymbol{B}^T\boldsymbol{D}, \tag{3.2.33}$$

Then $\mathbf{s}^*$ *is is a true source location, i.e.,*

$$\|\mathbf{s}^* - \boldsymbol{a}_i\| - |\mathbf{s}^* - \boldsymbol{a}_j\| = c_0\hat{\tau}_{ij}, \quad i, j = 1, 2, \cdots, m. \tag{3.2.34}$$

Remark 3.2.1 *If* $m = d + 1$, *then*

$$(\boldsymbol{B}^T\boldsymbol{B})^{-1}\boldsymbol{B}^T = \boldsymbol{B}^{-1}.$$

Therefore, the equation (3.2.32) can be simplified the following form:

$$\begin{aligned}
&\lambda^2\left(1 - \boldsymbol{D}^T(\boldsymbol{B}^{-1})^T\boldsymbol{B}^{-1}\boldsymbol{D}\right)\\
&+ \lambda\left(2\hat{\beta}_{d+1} - 2\left(\boldsymbol{B}^{-1}\boldsymbol{C} - \boldsymbol{a}_{d+1}\right)^T\boldsymbol{B}^{-1}\boldsymbol{D}\right)\\
&+ \hat{\beta}_{d+1}^2 - (\boldsymbol{B}^{-1}\boldsymbol{C} - \boldsymbol{a}_{d+1})^T(\boldsymbol{B}^{-1}\boldsymbol{C} - \boldsymbol{a}_{d+1}) = 0,
\end{aligned} \tag{3.2.35}$$

and

$$\mathbf{s}^* = \boldsymbol{B}^{-1}\boldsymbol{C} + \hat{\lambda}\boldsymbol{B}^{-1}\boldsymbol{D}, \tag{3.2.36}$$

Remark 3.2.2 *If* $d = 2$ *and* $m = 3$, *then*

$$B = \begin{pmatrix} 2(a_{11} - a_{31}) & 2(a_{12} - a_{32}) \\ 2(a_{21} - a_{31}) & 2(a_{22} - a_{32}) \end{pmatrix}$$

$$C = \begin{pmatrix} a_{11}^2 + a_{12}^2 - a_{31}^2 - a_{32}^2 + \hat{\beta}_3^2 - \hat{\beta}_1^2 \\ a_{21}^2 + a_{22}^2 - a_{31}^2 - a_{32}^2 + \hat{\beta}_3^2 - \hat{\beta}_2^2 \end{pmatrix}$$

$$D = \begin{pmatrix} 2(\hat{\beta}_3 - \hat{\beta}_1) \\ 2(\hat{\beta}_3 - \hat{\beta}_2) \end{pmatrix}$$

3.2.3 Estimator properties

In this section, we study the least square estimator of source localization with time-difference information of m points and noises.

Let $\boldsymbol{a}_i = (a_{i1}, a_{i2}, \cdots, a_{id})^T$, $i = 1, 2, \cdots, m$, be m points in $\mathbb{R}^d$, where $m \geq d + 1$. Assume

$$\text{rank}\{\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_m\} = d.$$

Let $(\hat{\mathbf{s}}, \hat{y}, \hat{\beta}, \hat{\alpha})$ be an optimal solution of (3.2.26) and let $\hat{\lambda}$ be a positive solution of the equation (3.2.32). Let r_i be a noisy observation of the distance between $\mathbf{s}^*$ and $\boldsymbol{a}_i$:

$$r_i = \beta_i^* + \varepsilon_i, \tag{3.2.37}$$

where $\beta_i^* = \hat{\beta}_i + \hat{\lambda}$, and $\varepsilon_1, \cdots, \varepsilon_m$ are noises. We assume that $\varepsilon_1, \cdots, \varepsilon_m$ are i.i.d. random variables with common mean 0 and the unknown variance σ^2.

Thus

$$\tau_{ij} = \hat{\tau}_{ij} + \frac{\varepsilon_i - \varepsilon_j}{c}, \tag{3.2.38}$$

where τ_{ij} denotes a noisy observation of $\hat{\tau}_{ij}$.

The noisy observation equations associated the equations (3.2.27) are as follows:

$$\|\mathbf{s} - \boldsymbol{a}_i\|^2 = (\beta_i^*)^2 + 2\beta_i^*\varepsilon_i + \varepsilon_i^2, i = 1, 2, \cdots, m. \tag{3.2.39}$$

This implies that

$$\|\mathbf{s} - \boldsymbol{a}_m\|^2 - \|\mathbf{s} - \boldsymbol{a}_i\|^2 = (\beta_m^*)^2 - (\beta_i^*)^2 + \zeta_i, i = 1, 2, \cdots, m - 1,$$

where

$$\zeta_i = 2\beta_m^*\varepsilon_m - 2\beta_i^*\varepsilon_i + \varepsilon_m^2 - \varepsilon_i^2, \quad i = 1, 2, \cdots, m - 1.$$

That is

$$\sum_{j=1}^{d} 2(a_{ij} - a_{mj})s_j = \sum_{j=1}^{d}(a_{ij}^2 - a_{mj}^2) + (\beta_m^*)^2 - (\beta_i^*)^2 + \zeta_i, i = 1, 2, \cdots, m - 1. \tag{3.2.40}$$

Then the equations (3.2.40) can be written as

$$\boldsymbol{\zeta} = \mathbf{Y} - \boldsymbol{B}\mathbf{s}, \tag{3.2.41}$$

where

$$\mathbf{Y} = \boldsymbol{C} + \hat{\lambda}\boldsymbol{D} + \boldsymbol{\zeta},$$

from equation(3.2.29) and

$$\boldsymbol{\zeta} = (\zeta_1, \cdots, \zeta_{m-1})^T.$$

Lemma 3.2.2 *If $E(\varepsilon_1^4) < \infty$, then the covariance matrix of $\boldsymbol{\zeta}$:*

$$\mathrm{Cov}\,(\boldsymbol{\zeta}, \boldsymbol{\zeta}) = \Sigma = (\sigma_{ij})_{(m-1)\times(m-1)}, \tag{3.2.42}$$

where

$$\sigma_{ij} = \begin{cases} 4(\beta_m^*)^2\sigma^2 + 4\beta_m^* v_3 + v_4 - \sigma^4, & \text{if } i \neq j, \\ 4((\beta_m^*)^2 + (\beta_i^*)^2)\sigma^2 + 4(\beta_m^* + \beta_i^*)v_3 + 2v_4 - 2\sigma^4, & \text{if } i = j, \end{cases}$$

Proof. For any i, j,

$$\begin{aligned} \zeta_i\zeta_j =& 4(\beta_m^*)^2\varepsilon_m^2 - 4\beta_m^*\beta_j^*\varepsilon_m\varepsilon_j + 2\beta_m^*\varepsilon_m^3 - 2\beta_m^*\varepsilon_m\varepsilon_j^2 \\ & - 4\beta_m^*\beta_i^*\varepsilon_m\varepsilon_i + 4\beta_i^*\beta_j^*\varepsilon_i\varepsilon_j - 2\beta_i^*\varepsilon_m^2\varepsilon_i + 2\beta_i^*\varepsilon_i\varepsilon_j^2 \\ & + 2\beta_m^*\varepsilon_m^3 - 2\beta_j^*\varepsilon_m^2\varepsilon_j + \varepsilon_m^4 - \varepsilon_m^2\varepsilon_j^2 \\ & - 2\beta_m^*\varepsilon_m\varepsilon_i^2 + 2\beta_j^*\varepsilon_i^2\varepsilon_j - \varepsilon_i^2\varepsilon_m^2 + \varepsilon_i^2\varepsilon_j^2. \end{aligned}$$

Then

$$\begin{aligned} & \mathrm{Cov}(\zeta_i, \zeta_j) \\ =& E\,(\zeta_i\zeta_j) \\ =& 4(\beta_m^*)^2E(\varepsilon_m^2) + 2\beta_m^*E(\varepsilon_m^3) + 4\beta_i^*\beta_j^*E(\varepsilon_i\varepsilon_j) + 2\beta_i^*(\varepsilon_i\varepsilon_j^2) \\ & + 2\beta_m^*E(\varepsilon_m^3) + E(\varepsilon_m^4) - E(\varepsilon_m^2)E(\varepsilon_j^2) + 2\beta_j^*E(\varepsilon_i^2\varepsilon_j) - E(\varepsilon_i^2)E(\varepsilon_m^2) + E(\varepsilon_i^2\varepsilon_j^2) \\ =& \begin{cases} 4(\beta_m^*)^2\sigma^2 + 4\beta_m^* v_3 + v_4 - \sigma^4, & \text{if } i \neq j, \\ 4((\beta_m^*)^2 + (\beta_i^*)^2)\sigma^2 + 4(\beta_m^* + \beta_i^*)v_3 + 2v_4 - 2\sigma^4, & \text{if } i = j, \end{cases} \end{aligned}$$

and $v_3 = E(\varepsilon_1^3)$, and $v_4 = E(\varepsilon_1^4)$. ■

The least square estimator $\tilde{\mathbf{s}}$ of $\hat{\mathbf{s}}$ is defined by

$$\arg\min_{s} \|\boldsymbol{B}\mathbf{s} - \mathbf{Y}\|^2. \tag{3.2.43}$$

Then

$$\tilde{\mathbf{s}} = (\boldsymbol{B}^T\boldsymbol{B})^{-1}\boldsymbol{B}^T\mathbf{Y} = \hat{\mathbf{s}} + (\boldsymbol{B}^T\boldsymbol{B})^{-1}\boldsymbol{B}^T\boldsymbol{\zeta}. \tag{3.2.44}$$

Theorem 3.2.2 *(1). $\tilde{\mathbf{s}}$ is a unbiased estimator of $\hat{\mathbf{s}}$, i.e.,*

$$E\left(\tilde{\mathbf{s}}\right) = \hat{\mathbf{s}}. \tag{3.2.45}$$

(2). If $E(\varepsilon_1^4) < \infty$, then the covariance matrix of $\tilde{\mathbf{s}}$:

$$\mathrm{Cov}\left(\tilde{\mathbf{s}}, \tilde{\mathbf{s}}\right) = (\boldsymbol{B}^T\boldsymbol{B})^{-1}\boldsymbol{B}^T\Sigma((\boldsymbol{B}^T\boldsymbol{B})^{-1}\boldsymbol{B}^T)^T, \tag{3.2.46}$$

Proof. (3.2.45) is obvious since

$$E(\zeta_i) = 0, i = 1, \cdots, m-1.$$

Next, let us show (3.2.46).

$$\begin{aligned}\mathrm{Cov}\left(\tilde{\mathbf{s}}, \tilde{\mathbf{s}}\right) =& E\left((\tilde{\mathbf{s}} - E(\tilde{\mathbf{s}}))(\tilde{\mathbf{s}} - E(\tilde{\mathbf{s}}))^T\right) \\ =& (\boldsymbol{B}^T\boldsymbol{B})^{-1}\boldsymbol{B}^T E\left(\boldsymbol{\zeta}\boldsymbol{\zeta}^T\right)((\boldsymbol{B}^T\boldsymbol{B})^{-1}\boldsymbol{B}^T)^T \\ =& (\boldsymbol{B}^T\boldsymbol{B})^{-1}\boldsymbol{B}^T\Sigma((\boldsymbol{B}^T\boldsymbol{B})^{-1}\boldsymbol{B}^T)^T.\end{aligned}$$

■

3.3 Bi-level method for source localization problem

The solution of (3.2.6) is not what we want, in this section we consider bi-level method to solve it. In the constrains $\begin{pmatrix} 1 & \beta_i \\ \beta_i & \alpha_i \end{pmatrix} \succeq 0$, we have $\alpha_i \geq \beta_i^2$. Since

$\alpha_i = \beta_i^2$ is the solution that we want, we need β_i to be bigger. Therefore we transfer our relaxation problem to be:

$$\begin{cases} \min \quad \sum_{i=1}^{m} \alpha_i - \sum_{i=1}^{m} c\gamma_i \\ \text{s.t.} \quad \beta_i - \beta_j = c_0 \hat{\tau}_{ij}, \quad i, j = 1, \cdots, m, \\ \qquad \begin{pmatrix} \boldsymbol{a}_i^T & -1 \end{pmatrix} \boldsymbol{Z} \begin{pmatrix} \boldsymbol{a}_i \\ -1 \end{pmatrix} = \alpha_i, \quad i = 1, \cdots, m, \\ \qquad \begin{pmatrix} 1 & \beta_i \\ \beta_i & \alpha_i \end{pmatrix} \succeq 0, \quad i = 1, \cdots, m, \\ \qquad \boldsymbol{Z}_{1:d,1:d} = I_d, \boldsymbol{Z} \succeq 0, \\ \qquad \beta_i \geq \gamma_i. \end{cases} \tag{3.3.47}$$

Then we need to find c such that the sum of $\alpha_i - \beta_i^2$ is minimized. That is

$$\min_c f(c) = \sum_{i=1}^{m} (\alpha_i - \beta_i^2) \tag{3.3.48}$$

This method do the best to let α_i close to β_i^2.

3.4 Mixed SDP-SOCP relaxation model for source localization combined with sensor network localization problem

3.4.1 Mixed SDP-SOCP relaxation model

In the previous section, we present the mixed SDP-SOCP relaxation model for source localization problem. In this section, we want solve the source localization problem combined with sensor network localization problem.

The standard sensor network localization problem takes edge distance information and attempts to determine the location of the sensors. For example, given the sensor-to-sensor edge set N_r with distances d_{ij} for all $(i,j) \in N_r$ and the sensor-to-anchor edge set N_a with distances $\hat{d}_{kj}$ for all $(k,j) \in N_a$, and given the anchor locations $\{\boldsymbol{a}_1, \dots, \boldsymbol{a}_m\}$ in $\mathbb{R}^d$, we want to determine the sensor locations $\{\boldsymbol{r}_1, \boldsymbol{r}_2, \dots, \boldsymbol{r}_n\}$ in $\mathbb{R}^d$ that satisfy

$$\begin{cases} \|\boldsymbol{r}_i - \boldsymbol{r}_j\|^2 = d_{ij}^2, & \forall (i,j) \in N_r, \\ \|\boldsymbol{a}_k - \boldsymbol{r}_j\|^2 = \hat{d}_{kj}^2, & \forall (k,j) \in N_a, \end{cases} \tag{3.4.49}$$

Let $\boldsymbol{R} = [\boldsymbol{r}_1, \boldsymbol{r}_2, \dots, \boldsymbol{r}_n]$ be the $\mathbb{R}^{d\times n}$ matrix determined by $\boldsymbol{r}_1, \boldsymbol{r}_2, \dots, \boldsymbol{r}_n$. Then, we have

$$\begin{cases} \|\boldsymbol{r}_i - \boldsymbol{r}_j\|^2 = \boldsymbol{e}_{ij}^T \boldsymbol{R}^T \boldsymbol{R} \boldsymbol{e}_{ij}, \quad \forall (i,j) \in N_r, \\ \|\boldsymbol{a}_k - \boldsymbol{r}_j\|^2 = \begin{pmatrix} \boldsymbol{a}_k^T & \boldsymbol{e}_j^T \end{pmatrix} \begin{pmatrix} \boldsymbol{I} \\ \boldsymbol{R}^T \end{pmatrix} [\boldsymbol{I} \ \boldsymbol{R}] \begin{pmatrix} \boldsymbol{a}_k \\ \boldsymbol{e}_j \end{pmatrix}, \quad \forall (k,j) \in N_a, \end{cases} \tag{3.4.50}$$

where $\boldsymbol{e}_{ij}$ is the vector with 1 at the i-th position, -1 at the j-th position and 0 elsewhere, $\boldsymbol{e}_j$ is the vector with -1 at the j-th position and 0 elsewhere. Let $\boldsymbol{Y} = \boldsymbol{R}^T\boldsymbol{R}$. Then (3.4.49) is equivalent to find a symmetric matrix $\boldsymbol{Y} \in \mathbb{R}^{n\times n}$ and a matrix $\boldsymbol{R} \in \mathbb{R}^{d\times n}$ such that the following equations are satisfied.

$$\begin{cases} \boldsymbol{e}_{ij}^T Y \boldsymbol{e}_{ij} = d_{ij}^2, \quad \forall (i,j) \in N_r, \\ \begin{pmatrix} \boldsymbol{a}_k^T & \boldsymbol{e}_j^T \end{pmatrix} \begin{pmatrix} \boldsymbol{I} & \boldsymbol{R} \\ \boldsymbol{R}^T & \boldsymbol{Y} \end{pmatrix} \begin{pmatrix} \boldsymbol{a}_k \\ \boldsymbol{e}_j \end{pmatrix} = \hat{d}_{kj}^2, \quad \forall (k,j) \in N_a, \\ \boldsymbol{Y} = \boldsymbol{R}^T \boldsymbol{R}. \end{cases} \tag{3.4.51}$$

An effective method for solving this problem is to relax $\boldsymbol{Y} = \boldsymbol{R}^T\boldsymbol{R}$ to $\boldsymbol{Y} \succeq$

$\boldsymbol{R}^T\boldsymbol{R}$ which is equivalent to

$$\boldsymbol{Z} := \begin{pmatrix} \boldsymbol{I} & \boldsymbol{R} \\ \boldsymbol{R}^T & \boldsymbol{Y} \end{pmatrix} \succeq 0.$$

Then, the relaxed version of the problem (3.4.51) can be represented as a standard semi-definite programming model.

$$\begin{cases} \min \ \ 0 \bullet \boldsymbol{Z} \\ \text{subject to } (\mathbf{0}^T \ \boldsymbol{e}_{ij}^T)\boldsymbol{Z}\begin{pmatrix} \mathbf{0} \\ \boldsymbol{e}_{ij} \end{pmatrix} = d_{ij}^2, \quad \forall (i,j) \in N_r, \\ \left(\boldsymbol{a}_k^T \ \boldsymbol{e}_j^T\right) \boldsymbol{Z} \begin{pmatrix} \boldsymbol{a}_k \\ \boldsymbol{e}_j \end{pmatrix} = \hat{d}_{kj}^2, \quad \forall (k,j) \in N_a, \\ \boldsymbol{Z}_{1:d,1:d} = I_d, \quad \boldsymbol{Z} \succeq 0. \end{cases} \tag{3.4.52}$$

For our source localization combined with sensor network localization problem, we assume there are some anchor microphones $\boldsymbol{a}_i = (a_{i1}, a_{i2}, \cdots, a_{id})^T$, $i = 1, \cdots, m$ whose locations are known and some sensor microphones $\boldsymbol{r}_i = (r_{i1}, r_{i2}, \cdots, r_{id})^T$, $i = 1, \cdots, n$ whose locations are unknown. Given a true source location $\mathbf{s}$, we also can derive the true time difference of arrival(TDOA) as the previous section

$$\begin{aligned} T(\{\boldsymbol{r}_j, \boldsymbol{r}_k\}, \mathbf{s}) &= \frac{\|\mathbf{s} - \boldsymbol{r}_j\| - \|\mathbf{s} - \boldsymbol{r}_k\|}{c_0}, \quad (j,k) \in T_r, \\ T(\{\boldsymbol{r}_j, \boldsymbol{a}_h\}, \mathbf{s}) &= \frac{\|\mathbf{s} - \boldsymbol{r}_j\| - \|\mathbf{s} - \boldsymbol{a}_h\|}{c_0}, \quad (j,h) \in T_a, \end{aligned} \tag{3.4.53}$$

where c_0 is the speed of sound in the air. The estimated TDOA will be given by $\hat{\tau}_{jk}$ using the signals received at the two sensor microphones and $\hat{\tau}_{jh}$ using the signals received at one sensor microphone and one anchor microphone. Also we have the information about sensor-to-sensor edge set N_r with distances d_{ij} for all $(i,j) \in N_r$ and the sensor-to-anchor edge set N_a with distances $\hat{d}_{kj}$ for all $(k,j) \in N_a$.

$$\begin{cases} \|\boldsymbol{r}_i - \boldsymbol{r}_j\|^2 = d_{ij}^2, \quad \forall (i,j) \in N_r, \\ \|\boldsymbol{a}_k - \boldsymbol{r}_j\|^2 = \hat{d}_{kj}^2, \quad \forall (k,j) \in N_a, \end{cases} \tag{3.4.54}$$

We want to determine the sensor locations $\{\boldsymbol{r}_1, \boldsymbol{r}_2, \dots, \boldsymbol{r}_n\}$ in $\mathbb{R}^d$ and the source location $\hat{\mathbf{s}}$ that satisfy time difference constraints plus distance constraints:

$$\begin{cases} \|\mathbf{s} - \boldsymbol{r}_j\| - \|\mathbf{s} - \boldsymbol{r}_k\| = c_0\hat{\tau}_{jk}, & (j,k) \in T_r, \\ \|\mathbf{s} - \boldsymbol{r}_j\| - \|\mathbf{s} - \boldsymbol{a}_h\| = c_0\hat{\tau}_{jh}, & (j,h) \in T_a, \\ \|\boldsymbol{r}_i - \boldsymbol{r}_j\|^2 = d_{ij}^2, \quad \forall (i,j) \in N_r, \\ \|\boldsymbol{r}_j - \boldsymbol{a}_k\|^2 = \hat{d}_{kj}^2, \quad \forall (k,j) \in N_a, \end{cases} \tag{3.4.55}$$

Denote such sensor-sensor edges by N_{tr} and sensor-anchor edges by N_{ta} in $T_r \cup T_a$, i.e., $N_{tr} = \cup_{(j,k)\in T_r}\{(s,j),(s,k)\}$, $N_{ta} = \cup_{(j,h)\in T_a}\{(s,j),(s,h)\}$. We assume that $N_{tr} \cap N_r = \emptyset$ and $N_{ta} \cap N_a = \emptyset$. (3.4.55) is equivalent to find a symmetric matrix $\boldsymbol{Y} \in \mathbb{R}^{n\times n}$ and a matrix $\boldsymbol{R} \in \mathbb{R}^{d\times n}$ such that the following equations are satisfied.

$$\begin{cases} \beta_{sj} - \beta_{sk} = c_0\hat{\tau}_{jk}, \quad \forall (j,k) \in T_r, \\ \beta_{sj} - \beta_{sh} = c_0\hat{\tau}_{jh}, \quad \forall (j,h) \in T_a, \\ \beta_{sj} = \|\mathbf{s} - \boldsymbol{r}_j\|, \quad \forall (s,j) \in N_{tr}, \\ \beta_{sh} = \|\mathbf{s} - \boldsymbol{a}_h\|, \quad \forall (s,h) \in N_{ta}, \\ \boldsymbol{e}_{1,j+1}^T \boldsymbol{Y} \boldsymbol{e}_{1,j+1} = \alpha_{sj}, \beta_{sj} \geq 0 \quad \forall (s,j) \in N_{tr}, \\ \begin{pmatrix} \boldsymbol{a}_h^T & \boldsymbol{e}_1^T \end{pmatrix} \begin{pmatrix} \boldsymbol{I} & \boldsymbol{R} \\ \boldsymbol{R}^T & \boldsymbol{Y} \end{pmatrix} \begin{pmatrix} \boldsymbol{a}_h \\ \boldsymbol{e}_1 \end{pmatrix} = \alpha_{sh}, \beta_{sh} \geq 0 \quad \forall (s,h) \in N_{ta}, \\ \alpha_{sj} = \beta_{sj}^2, \quad \forall (s,j) \in N_{tr}, \\ \alpha_{sh} = \beta_{sh}^2, \quad \forall (s,h) \in N_{ta}, \\ \boldsymbol{e}_{i+1,j+1}^T \boldsymbol{Y} \boldsymbol{e}_{i+1,j+1} = d_{ij}^2, \quad \forall (i,j) \in N_r, \\ \begin{pmatrix} \boldsymbol{a}_h^T & \boldsymbol{e}_{i+1}^T \end{pmatrix} \begin{pmatrix} \boldsymbol{I} & \boldsymbol{R} \\ \boldsymbol{R}^T & \boldsymbol{Y} \end{pmatrix} \begin{pmatrix} \boldsymbol{a}_h \\ \boldsymbol{e}_{i+1} \end{pmatrix} = \hat{d}_{ih}^2, \quad \forall (i,h) \in N_a, \\ \boldsymbol{Y} = \boldsymbol{R}^T \boldsymbol{R}. \end{cases} \tag{3.4.56}$$

It is equivalent to the following problem

$$\begin{cases} \beta_{sj} - \beta_{sk} = c_0 \hat{\tau}_{jk}, \quad \forall (j,k) \in T_r, \\ \beta_{sj} - \beta_{sh} = c_0 \hat{\tau}_{jh}, \quad \forall (j,h) \in T_a, \\ \beta_{sj} = \|\mathbf{s} - \boldsymbol{r}_j\|, \quad \forall (s,j) \in N_{tr}, \\ \beta_{sh} = \|\mathbf{s} - \boldsymbol{a}_h\|, \quad \forall (s,h) \in N_{ta}, \\ \boldsymbol{e}_{1,j+1}^T \boldsymbol{Y} \boldsymbol{e}_{1,j+1} = \alpha_{sj}, \beta_{sj} \geq 0 \quad \forall (s,j) \in N_{tr}, \\ \left(\boldsymbol{a}_h^T \; \boldsymbol{e}_1^T \right) \begin{pmatrix} \boldsymbol{I} & \boldsymbol{R} \\ \boldsymbol{R}^T & \boldsymbol{Y} \end{pmatrix} \begin{pmatrix} \boldsymbol{a}_h \\ \boldsymbol{e}_1 \end{pmatrix} = \alpha_{sh}, \beta_{sh} \geq 0 \quad \forall (s,h) \in N_{ta}, \\ \alpha_{sj} \geq \beta_{sj}^2, \quad \forall (s,j) \in N_{tr}, \\ \alpha_{sj} \leq \beta_{sj}^2, \quad \forall (s,j) \in N_{tr}, \\ \alpha_{sh} \geq \beta_{sh}^2, \quad \forall (s,h) \in N_{ta}, \\ \alpha_{sh} \leq \beta_{sh}^2, \quad \forall (s,h) \in N_{ta}, \\ \boldsymbol{e}_{i+1,j+1}^T \boldsymbol{Y} \boldsymbol{e}_{i+1,j+1} = d_{ij}^2, \quad \forall (i,j) \in N_r, \\ \left(\boldsymbol{a}_h^T \; \boldsymbol{e}_{i+1}^T \right) \begin{pmatrix} \boldsymbol{I} & \boldsymbol{R} \\ \boldsymbol{R}^T & \boldsymbol{Y} \end{pmatrix} \begin{pmatrix} \boldsymbol{a}_h \\ \boldsymbol{e}_{i+1} \end{pmatrix} = \hat{d}_{ih}^2, \quad \forall (i,h) \in N_a, \\ \boldsymbol{Y} = \boldsymbol{R}^T \boldsymbol{R}. \end{cases} \tag{3.4.57}$$

The same as the previous technique, the inequality constraints $\alpha_{sj} \geq \beta_{sj}^2$ is equivalent to $\begin{pmatrix} 1 & \beta_{sj} \\ \beta_{sj} & \alpha_{sj} \end{pmatrix} \succeq 0$, and $\alpha_{sj} \leq \beta_{sj}^2$ is equivalent to $\|\mathbf{s} - \boldsymbol{a}_i\| \leq \beta_{is}$.

Thus we relax the equality constraints $\|\mathbf{s} - \boldsymbol{r}_j\| = \beta_{sj}$ to " $\leq$ " inequality constraints, which yields a second order cone problem. Also relax $\boldsymbol{Y} = \boldsymbol{R}^T \boldsymbol{R}$ to $\boldsymbol{Y} \succeq \boldsymbol{R}^T \boldsymbol{R}$ which is equivalent to

$$\boldsymbol{Z} := \begin{pmatrix} \boldsymbol{I} & \boldsymbol{R} \\ \boldsymbol{R}^T & \boldsymbol{Y} \end{pmatrix} \succeq 0.$$

Then, the relaxed version of the problem (3.4.57) can be represented as a mixed SDP-SOCP model.

$$
\left\{
\begin{aligned}
&\min \quad \sum_{(s,j)\in N_{tr}} \alpha_{sj} + \sum_{(s,h)\in N_{ta}} \alpha_{sh} \\
&\text{subject to } \beta_{sj} - \beta_{sk} = c_0 \hat{\tau}_{jk}, \quad \forall (j,k) \in T_r, \\
&\qquad \beta_{sj} - \beta_{sh} = c_0 \hat{\tau}_{jh}, \quad \forall (j,h) \in T_a, \\
&\qquad \beta_{sj} \geq \|\mathbf{s} - \boldsymbol{r}_j\|, \quad \forall (s,j) \in N_{tr}, \\
&\qquad \beta_{sh} \geq \|\mathbf{s} - \boldsymbol{a}_h\|, \quad \forall (s,h) \in N_{ta}, \\
&\qquad (\mathbf{0}^T \ \boldsymbol{e}_{1,j+1}^T)\boldsymbol{Z} \begin{pmatrix} \mathbf{0} \\ \boldsymbol{e}_{1,j+1} \end{pmatrix} = \alpha_{sj}, \beta_{sj} \geq 0 \quad \forall (s,j) \in N_{tr}, \\
&\qquad \left(\boldsymbol{a}_h^T \ \boldsymbol{e}_1^T\right) \boldsymbol{Z} \begin{pmatrix} \boldsymbol{a}_h \\ \boldsymbol{e}_1 \end{pmatrix} = \alpha_{sh}, \beta_{sh} \geq 0 \quad \forall (s,h) \in N_{ta}, \\
&\qquad \begin{pmatrix} 1 & \beta_{sj} \\ \beta_{sj} & \alpha_{sj} \end{pmatrix} \succeq 0, \forall (s,j) \in N_{tr}, \\
&\qquad \begin{pmatrix} 1 & \beta_{sh} \\ \beta_{sh} & \alpha_{sh} \end{pmatrix} \succeq 0, \forall (s,h) \in N_{ta}, \\
&\qquad (\mathbf{0}^T \ \boldsymbol{e}_{i+1,j+1}^T)\boldsymbol{Z} \begin{pmatrix} \mathbf{0} \\ \boldsymbol{e}_{i+1,j+1} \end{pmatrix} = d_{ij}^2, \quad \forall (i,j) \in N_r, \\
&\qquad \left(\boldsymbol{a}_h^T \ \boldsymbol{e}_{i+1}^T\right) \boldsymbol{Z} \begin{pmatrix} \boldsymbol{a}_h \\ \boldsymbol{e}_{i+1} \end{pmatrix} = \hat{d}_{ih}^2, \quad \forall (i,h) \in N_a, \\
&\qquad \boldsymbol{Z}_{1:d,1:d} = I_d, \quad \boldsymbol{Z} \succeq 0. \\
&\qquad \boldsymbol{r}_j = \boldsymbol{Z}_{1:d,d+1+j}
\end{aligned}
\right. \tag{3.4.58}
$$

3.4.2 Statistical analysis

Statistical analysis for least square model

Assume we have M pairs of microphones $\boldsymbol{r}_j$ and $\boldsymbol{r}_k$, $(j,k) \in T_r$, whose locations are unknown, and N pairs of microphones $\boldsymbol{a}_h$ and $\boldsymbol{r}_j$, $(h,j) \in T_a$, whose locations are known and unknown respectively. Given a true source location $\mathbf{s}$, we can derive the true time difference of arrival(TDOA)

$$T(\{\boldsymbol{r}_j, \boldsymbol{r}_k\}, \mathrm{s}) = \frac{\| \mathrm{s} - \boldsymbol{r}_j \| - \| \mathrm{s} - \boldsymbol{r}_k \|}{c},$$

and

$$T(\{\boldsymbol{a}_h, \boldsymbol{r}_j\}, \mathrm{s}) = \frac{\| \mathrm{s} - \boldsymbol{a}_h \| - \| \mathrm{s} - \boldsymbol{r}_j \|}{c},$$

where c is the speed of sound in the air. Then the estimate of these true TDOA will be given by $\hat{\tau}_{jk}$ and $\hat{\tau}_{hj}$ respectively using the signals received at the two microphones.

We need to estimate the source location $\hat{\mathrm{s}}$. If the TDOA estimates for each microphone pair are assumed to be independently corrupted by zero-mean additive white Gaussian noise of equal variance, then the estimator can be found by minimizing the least squares error criterion

$$V(\hat{\mathrm{s}}) = \sum_{(j,k)\in T_r} (\hat{\tau}_{jk} - T(\{\hat{\boldsymbol{r}}_j, \hat{\boldsymbol{r}}_k\}, \hat{\mathrm{s}}))^2 + \sum_{(h,j)\in T_a} (\hat{\tau}_{hj} - T(\{\boldsymbol{a}_h, \hat{\boldsymbol{r}}_j\}, \hat{\mathrm{s}}))^2 \tag{3.4.59}$$

Then the source location estimate if found from

$$\hat{\mathrm{s}}_{LS} = \arg\min_{\hat{\mathrm{s}}} V(\hat{\mathrm{s}})$$

$T(\{\hat{\boldsymbol{r}}_j, \hat{\boldsymbol{r}}_k\}, \hat{\mathrm{s}})$ can be expressed as their first-order Taylor series expansion

$$\begin{aligned}
&T(\{\hat{\boldsymbol{r}}_j, \hat{\boldsymbol{r}}_k\}, \hat{\mathrm{s}})\\
&= \frac{\| \hat{\mathrm{s}} - \hat{\boldsymbol{r}}_j \| - \| \hat{\mathrm{s}} - \hat{\boldsymbol{r}}_k \|}{c}\\
&= \frac{\| \mathrm{s} - \boldsymbol{r}_j + \triangle\mathrm{s} - \triangle\boldsymbol{r}_j \| - \| \mathrm{s} - \boldsymbol{r}_k + \triangle\mathrm{s} - \triangle\boldsymbol{r}_k \|}{c}\\
&\approx \frac{\| \mathrm{s} - \boldsymbol{r}_j \| + \frac{\mathrm{s}-\boldsymbol{r}_j}{\|\mathrm{s}-\boldsymbol{r}_j\|} \cdot (\triangle\mathrm{s} - \triangle\boldsymbol{r}_j) - (\| \mathrm{s} - \boldsymbol{r}_k \| + \frac{\mathrm{s}-\boldsymbol{r}_k}{\|\mathrm{s}-\boldsymbol{r}_k\|} \cdot (\triangle\mathrm{s} - \triangle\boldsymbol{r}_k))}{c}\\
&= T(\{\boldsymbol{r}_j, \boldsymbol{r}_k\}, \mathrm{s}) + \frac{1}{c}\left(\frac{(\mathrm{s} - \boldsymbol{r}_j)}{\| \mathrm{s} - \boldsymbol{r}_j \|} \cdot (\triangle\mathrm{s} - \triangle\boldsymbol{r}_j) - \frac{(\mathrm{s} - \boldsymbol{r}_k)}{\| \mathrm{s} - \boldsymbol{r}_k \|} \cdot (\triangle\mathrm{s} - \triangle\boldsymbol{r}_k) \right).
\end{aligned}$$

Thus

$$\begin{aligned}
&T(\{\hat{\boldsymbol{r}}_j, \hat{\boldsymbol{r}}_k\}, \hat{\mathrm{s}}) - T(\{\boldsymbol{r}_j, \boldsymbol{r}_k\}, \mathrm{s})\\
\approx&\frac{1}{c}\left(\frac{(\mathrm{s} - \boldsymbol{r}_j)}{\| \mathrm{s} - \boldsymbol{r}_j \|} - \frac{(\mathrm{s} - \boldsymbol{r}_k)}{\| \mathrm{s} - \boldsymbol{r}_k \|} \right) \cdot \triangle\mathrm{s} - \frac{(\mathrm{s} - \boldsymbol{r}_j)}{c\,\| \mathrm{s} - \boldsymbol{r}_j \|} \cdot \triangle\boldsymbol{r}_j + \frac{(\mathrm{s} - \boldsymbol{r}_k)}{c\,\| \mathrm{s} - \boldsymbol{r}_k \|} \cdot \triangle\boldsymbol{r}_k
\end{aligned}$$

For $(j,k) \in T_r$, set

$$\begin{aligned}
\mathbf{h}_{jk} =& \frac{1}{c}\left(\frac{(\mathbf{s}-\boldsymbol{r}_j)}{\parallel \mathbf{s}-\boldsymbol{r}_j \parallel} - \frac{(\mathbf{s}-\boldsymbol{r}_k)}{\parallel \mathbf{s}-\boldsymbol{r}_k \parallel}\right),\\
\phi_{jk} =& \left(-\tfrac{(\mathbf{s}-\boldsymbol{r}_j)}{c\|\mathbf{s}-\boldsymbol{r}_j\|}, \tfrac{(\mathbf{s}-\boldsymbol{r}_k)}{c\|\mathbf{s}-\boldsymbol{r}_k\|}\right)^T,\\
\triangle\boldsymbol{r}_{jk} =& \left(\triangle\boldsymbol{r}_j, \triangle\boldsymbol{r}_k\right)^T
\end{aligned}$$

Then

$$T(\{\hat{\boldsymbol{r}}_j, \hat{\boldsymbol{r}}_k\}, \hat{\mathbf{s}}) - T(\{\boldsymbol{r}_j, \boldsymbol{r}_k\}, \mathbf{s}) \approx \mathbf{h}_{jk} \cdot \triangle\mathbf{s} + \phi_{jk}^T \triangle\boldsymbol{r}_{jk}, \quad (j,k) \in T_r.$$

Similarly, we can get

$$T(\{\boldsymbol{a}_h, \hat{\boldsymbol{r}}_j\}, \hat{\mathbf{s}}) - T(\{\boldsymbol{a}_h, \boldsymbol{r}_j\}, \mathbf{s}) \approx \mathbf{h}_{hj} \cdot \triangle\mathbf{s} + \phi_{hj} \cdot \triangle\boldsymbol{r}_{hj}, \quad (h,j) \in T_a,$$

where

$$\begin{aligned}
\mathbf{h}_{hj} =& \frac{1}{c}\left(\frac{(\mathbf{s}-\boldsymbol{a}_h)}{\parallel \mathbf{s}-\boldsymbol{a}_h \parallel} - \frac{(\mathbf{s}-\boldsymbol{r}_j)}{\parallel \mathbf{s}-\boldsymbol{r}_j \parallel}\right),\\
\phi_{hj} =& \frac{(\mathbf{s}-\boldsymbol{r}_j)}{c \parallel \mathbf{s}-\boldsymbol{r}_j \parallel},\\
\triangle\boldsymbol{r}_{hj} =& \triangle\boldsymbol{r}_j.
\end{aligned}$$

For convenience, according to the same order we arrange

$T(\{\hat{\boldsymbol{r}}_j, \hat{\boldsymbol{r}}_k\}, \hat{\mathbf{s}}) - T(\{\boldsymbol{r}_j, \boldsymbol{r}_k\}, \mathbf{s}),\ (j,k) \in T_r,\ \ T(\{\boldsymbol{a}_h, \hat{\boldsymbol{r}}_j\}, \hat{\mathbf{s}}) - T(\{\boldsymbol{a}_h, \boldsymbol{r}_j\}, \mathbf{s}),\ (h,j) \in T_a,$

$\hat{\tau}_{jk} - T(\{\hat{\boldsymbol{r}}_j, \hat{\boldsymbol{r}}_k\}, \hat{\mathbf{s}}),\ (j,k) \in T_r,\ \ \hat{\tau}_{hj} - T(\{\boldsymbol{a}_h, \hat{\boldsymbol{r}}_j\}, \hat{\mathbf{s}}),\ (h,j) \in T_a,$

$\triangle\boldsymbol{r}_{jk},\ (j,k) \in T_r,\ \ \triangle\boldsymbol{r}_{hj},\ (h,j) \in T_a,$

$\phi_{jk},\ (j,k) \in T_r,\ \ \phi_{hj},\ (h,j) \in T_a,$

$\mathbf{h}_{jk},\ (j,k) \in T_r,\ \ \mathbf{h}_{hj},\ (h,j) \in T_a.$

respectively, into

$$\begin{aligned}
&\triangle\tau_{\hat{\mathbf{s}}} = (\triangle\tau_{\hat{s}}^1, \cdots, \triangle\tau_{\hat{s}}^{M+N})^T,\\
&\triangle\tau_{\hat{\tau}} = (\triangle\tau_{\hat{\tau}}^1, \cdots, \triangle\tau_{\hat{\tau}}^{M+N})^T,\\
&\triangle\boldsymbol{r} = \left(\triangle\boldsymbol{r}_1^T, \triangle\boldsymbol{r}_2^T, \cdots, \triangle\boldsymbol{r}_M^T, \triangle\boldsymbol{r}_{M+1}, \cdots, \triangle\boldsymbol{r}_{M+N}\right)^T,\\
&(\phi_1^T, \cdots, \phi_M^T, \phi_{M+1}, \cdots, \phi_{M+N})^T,\\
&\mathbf{H} = \left(\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_M, \mathbf{h}_{M+1}, \cdots, \mathbf{h}_{M+N}\right)^T.
\end{aligned}$$

Set

$$\mathbf{\Phi} = \begin{pmatrix} \phi_1^T & & & & & & \\ & \phi_2^T & & & & & \\ & & \ddots & & & & \\ & & & \phi_M^T & & & \\ & & & & \phi_{M+1} & & \\ & & & & & \ddots & \\ & & & & & & \phi_{M+N} \end{pmatrix},$$

Then

$$\triangle\tau_{\hat{\mathbf{s}}} = \mathbf{H}\triangle\mathbf{s} + \Phi\triangle\boldsymbol{r},$$

and

$$\begin{aligned} V(\hat{\mathrm{s}}) = & \sum_{(j,k)\in T_r} (\hat{\tau}_{jk} - T(\{\hat{\boldsymbol{r}}_j, \hat{\boldsymbol{r}}_k\}, \hat{\mathrm{s}}))^2 + \sum_{(h,j)\in T_a} (\hat{\tau}_{hj} - T(\{\boldsymbol{a}_h, \hat{\boldsymbol{r}}_j\}, \hat{\mathrm{s}}))^2 \\ = & \sum_{(j,k)\in T_r} (\hat{\tau}_{jk} - T(\{\boldsymbol{r}_j, \boldsymbol{r}_k\}, \mathrm{s}) + T(\{\boldsymbol{r}_j, \boldsymbol{r}_k\}, \mathrm{s}) - T(\{\hat{\boldsymbol{r}}_j, \hat{\boldsymbol{r}}_k\}, \hat{\mathrm{s}}))^2 \\ & + \sum_{(h,j)\in T_a} (\hat{\tau}_{hj} - T(\{\boldsymbol{a}_h, \boldsymbol{r}_j\}, \mathrm{s}) + T(\{\boldsymbol{a}_h, \boldsymbol{r}_j\}, \mathrm{s}) - T(\{\boldsymbol{a}_h, \hat{\boldsymbol{r}}_j\}, \hat{\mathrm{s}}))^2 \\ = & (\triangle\tau_{\hat{\tau}} - \triangle\tau_{\hat{s}})^T(\triangle\tau_{\hat{\tau}} - \triangle\tau_{\hat{s}}) \\ = & (\triangle\tau_{\hat{\tau}} - \Phi\triangle\boldsymbol{r} - \mathbf{H}\triangle\mathbf{s})^T(\triangle\tau_{\hat{\tau}} - \Phi\triangle\boldsymbol{r} - \mathbf{H}\triangle\mathbf{s}). \end{aligned}$$

It can be shown that the right side of the above equation can be minimized at

$$\triangle\mathbf{s}_{LS} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T(\triangle\tau_{\hat{\tau}} - \Phi\triangle\boldsymbol{r})$$

The covariance of $\triangle\mathbf{s}_{LS}$ is given by

$$\mathrm{cov}\{\triangle\mathbf{s}_{LS}\} = E[(\triangle\mathbf{s}_{LS} - E(\triangle\mathbf{s}_{LS}))(\triangle\mathbf{s}_{LS} - E(\triangle\mathbf{s}_{LS}))^T]$$

Since TDOA estimates are assumed to be corrupted with a zero-mean, uncorrelated noise, $E(\triangle \mathbf{s}_{LS}) = 0$. Then

$$\begin{aligned}
&\mathrm{cov}\{\triangle \mathbf{s}_{LS}\} \\
&= E[\triangle \mathbf{s}_{LS} \triangle \mathbf{s}_{LS}^T] \\
&= (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}' E\left(\triangle\tau_{\hat{\tau}}\triangle\tau_{\hat{\tau}}^T + \Phi\triangle\boldsymbol{r}\triangle\boldsymbol{r}^T\Phi\right)\mathbf{H}((\mathbf{H}^T\mathbf{H})^{-1})^T \\
&= (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}'\left(\mathrm{cov}\{\triangle\tau_{\hat{\tau}}\} + \mathrm{cov}\{\Phi\triangle\boldsymbol{r}\}\right)\mathbf{H}(\mathbf{H}^T\mathbf{H})^{-1}
\end{aligned} \tag{3.4.60}$$

Statistical analysis for least absolute value model

In this subsection we consider the least absolute deviation error:

$$E(\hat{\mathbf{s}}) = \sum_{(j,k)\in T_r} |\hat{\tau}_{jk} - T(\{\hat{\boldsymbol{r}}_j, \hat{\boldsymbol{r}}_k\}, \hat{\mathbf{s}})| + \sum_{(h,j)\in T_a} |\hat{\tau}_{hj} - T(\{\boldsymbol{a}_h, \hat{\boldsymbol{r}}_j\}, \hat{\mathbf{s}})|$$

From some estimates in previous subsection, $E(\hat{\mathbf{s}})$ can be rewritten as

$$E(\hat{\mathbf{s}}) = \left|\triangle\tau_{\hat{\tau}} - \Phi\triangle\boldsymbol{r} - \mathbf{H}\triangle\mathbf{s}\right|. \tag{3.4.61}$$

The least absolute deviation (LAD) estimator $\triangle\mathbf{s}_{LAD}$ can be found by minimizing the least absolute deviation error criterion

$$\triangle\mathbf{s}_{LAD} = \arg\min_{\triangle\mathbf{s}} \left|\triangle\tau_{\hat{\tau}} - \Phi\triangle\boldsymbol{r} - \mathbf{H}\triangle\mathbf{s}\right|.$$

Let y_i denote the i-th component of $\triangle\tau_{\hat{\tau}} - \Phi\triangle\boldsymbol{r}$. Set $\beta = \triangle\mathbf{s}$ and let z_i denote the i-th row of $\mathbf{H}$. Then, the minimization $\min_{\triangle\mathbf{s}} E(\hat{\mathbf{s}})$ of the right side of the above equation is equivalent to the least absolute deviation estimator $\tilde{\beta}$ of the linear model

$$y_i = z_i\beta + e_i, \quad i = 1, 2, \cdots, M+N,$$

where e_i, $i = 1, 2, \cdots, M+N$, are independent random variables with a common probability density function f. We assume that the median of e_i is 0 and f is a continuous, and positive in a neighborhood of 0.

Set $S = \sum_{i=1}^{M+N} z_i^T z_i = \mathbf{H}^T\mathbf{H}$, $x_i = z_i S^{-1/2}$. Then $\sum_{i=1}^{M+N} x_i^T x_i = I_{M+N}$ and (see Theorem 1 in Babu [1989]),

$$2f(0)S^{1/2}\tilde{\beta} \approx \sum_{i=1}^{M+N} x_i^T \mathrm{sign} e_i = \sum_{i=1}^{M+N} x_i^T \mathrm{sign}(y_i - z_i\beta).$$

That is, the least absolute deviation estimator $\triangle \mathbf{s}_{LAD}$ of $\triangle \mathbf{s}$:

$$\triangle \mathbf{s}_{LAD} \approx \frac{1}{2f(0)} \sum_{i=1}^{M+N} (\mathbf{H}^T\mathbf{H})^{-1} z_i^T \text{sign} e_i.$$

Since the median of e_i is 0, then $P(e_i \le 0) = P(e_i \ge 0) = \frac{1}{2}$, $E(\text{sign}e_i) = -P(e_i < 0) + P(e_i > 0) = 0$ and $E[(\text{sign}e_i)^2] = 1$, which implies that

$$\begin{aligned}
\text{cov}(\triangle \mathbf{s}_{LAD}, \triangle \mathbf{s}_{LAD}) &= E(\triangle \mathbf{s}_{LAD}(\triangle \mathbf{s}_{LAD})^T) \\
&\approx \frac{1}{4f^2(0)} E(\sum_{i=1}^{M+N} (\mathbf{H}^T\mathbf{H})^{-1} z_i^T z_j (\mathbf{H}^T\mathbf{H})^{-1} \text{sign}e_i \text{sign}e_j) \\
&= \frac{1}{4f^2(0)} (\mathbf{H}^T\mathbf{H})^{-1} \sum_{i=1}^{M+N} z_i z_j (\mathbf{H}^T\mathbf{H})^{-1} \\
&= \frac{1}{4f^2(0)} (\mathbf{H}^T\mathbf{H})^{-1}.
\end{aligned}$$

In particular, if $e_i \sim N(0, \sigma^2)$, then $f(0) = \frac{1}{\sqrt{2\pi}\sigma}$, and

$$\text{cov}(\triangle \mathbf{s}_{LAD}, \triangle \mathbf{s}_{LAD}) \approx \frac{\pi\sigma^2}{2} (\mathbf{H}^T\mathbf{H})^{-1}.$$

Chapter 4

The solution region of the mixed SDP-SOCP model associated with source localization problem

4.1 Introduction

In the last chapter, we have presented the the mixed SDP-SOCP relaxation model (3.2.25) of (3.2.2). Some methods to obtain the exact solution for the source localization have been proposed. The estimator properties for the true source location under noise has been studied. It is obvious that the solution of (3.2.2) is a feasible solution for (3.2.25). For the SOCP relaxation part, by Proposition 5.1 (a) in (Tseng [2007]), we know that the optimal source location s must lies in the convex hull of some microphones a_i. That means when the source is located in the convex hull of microphone array, it can be correctly localized by the above SDP-SOCP relaxation model (3.2.25). However, when the source is located outside of the convex hull of microphone array, the optimal solution of the SDP-SOCP relaxation problem (3.2.25) may has some errors. The following question is natural and important:

(P). When the number of microphone is fixed and the error of time difference of arrival estimates goes to zero, what is the condition such that the solution for (3.2.25) is the solution of (3.2.2)?

In this chapter, we shall answer this question. We study the problem in two-dimensional case. We obtain a representation for the solution of the mixed SDP-SOCP model and the characterization such that the mixed SDP-SOCP model has an exact relaxation. The characterization shows that the source localization with some time-difference information can be solved exactly by the mixed SDP-SOCP relaxation model in a larger region than the triangle region determined by three points. The representation theorem and its proof are presented in Section 2 for the solution of the mixed SDP-SOCP model. The characterization theorem and its proof are given in Section 3.

4.2 A representation theorem for the solution of the mixed SDP-SOCP

In this section, we study some properties of the solution possessed by the mixed SDP-SOCP in $\mathbb{R}^2$, i.e., $d = 2$ and $m = 3$. For standardization, since most array configurations can be represented by a union of triangular array, we shall simplify the theorem by deriving the properties for a triangular array. The main result is Theorem 4.2.2 which gives a representation for the solution of the mixed SDP-SOCP and which plays a key role in the proof of the characterization theorem (Theorem 4.3.1 in next section), which is important in deriving the geometry of the localizable region.

Let $\alpha = (\alpha_1, \alpha_2, \alpha_3)^T$, $\beta = (\beta_1, \beta_2, \beta_3)^T$, $\mathbf{s} = (s_1, s_2)^T$. Let $\boldsymbol{a}_i = (a_{i1}, a_{i2})^T$, $i = 1, 2, 3$, be three points in $\mathbb{R}^2$. Then the mixed SDP-SOCP (3.2.25) can be written the equivalent form:

$$\begin{cases} \min & \alpha_1 + \alpha_2 + \alpha_3 \\ \text{s.t.} & \beta_1 - \beta_2 = c_0 \hat{\tau}_{12,} \\ & \beta_1 - \beta_3 = c_0 \hat{\tau}_{13}, \\ & \|\mathbf{s} - \boldsymbol{a}_i\|^2 + (y - \|\mathbf{s}\|^2) = \alpha_i, \quad i = 1, 2, 3, \\ & y \geq \|\mathbf{s}\|^2, \quad \alpha_i \geq \beta_i^2, \quad \beta_i \geq \|\mathbf{s} - \boldsymbol{a}_i\|, \quad i = 1, 2, 3. \end{cases} \tag{4.2.1}$$

For the convenience, we use the notation $(\mathbf{s}, y, \beta, \alpha)$ to denote a feasible solution for (4.2.1).

Let $\mathbf{s}^* = (s_1^*, s_2^*)^T$ satisfy

$$\|\mathbf{s}^* - \boldsymbol{a}_1\| - \|\mathbf{s}^* - \boldsymbol{a}_2\| = c_0 \tau_{12}, \ \|\mathbf{s}^* - \boldsymbol{a}_1\| - \|\mathbf{s}^* - \boldsymbol{a}_3\| = c_0 \tau_{13}.$$

Denote by $\beta_i^* = \|\mathbf{s}^* - \boldsymbol{a}_i\|$, $i = 1, 2, 3$. Set $\beta^* = (\beta_1^*, \beta_2^*, \beta_3^*)^T$, $\alpha^* = ((\beta_1^*)^2, (\beta_2^*)^2, (\beta_3^*)^2)^T$, and $y^* = \|\mathbf{s}^*\|^2$. Then $(\mathbf{s}^*, y^*, \beta^*, \alpha^*)$ is a feasible solution of (4.2.1).

Theorem 4.2.1 *If $\mathbf{s}^*$ is in the triangle region $\Delta_{a_1 a_2 a_3}$ determined by the three points $\boldsymbol{a}_1$, $\boldsymbol{a}_2$ and $\boldsymbol{a}_3$, then $(\mathbf{s}^*, y^*, \beta^*, \alpha^*)$ is an optimal solution of (4.2.1)*

Proof. Let $(\hat{\mathbf{s}}, \hat{y}, \hat{\beta}, \hat{\alpha})$ be an optimal solution of (4.2.1), then there exist a $\lambda \geq 0$ such that

$$\hat{\beta}_i = \beta_i^* - \lambda, \quad i = 1, 2, 3,$$

Thus

$$\hat{\mathbf{s}} \in \cap_{i=1}^3 B(\boldsymbol{a}_i, \beta_i^* - \lambda) \subset \cap_{i=1}^3 B(\boldsymbol{a}_i, \beta_i^*).$$

But if $\mathbf{s}^* \in \Delta_{a_1 a_2 a_3}$, then

$$\cap_{i=1}^3 B(\boldsymbol{a}_i, \beta_i^*) = \{\mathbf{s} : \|\mathbf{s} - \boldsymbol{a}_i\| \leq \beta_i^* = \|\mathbf{s}^* - \boldsymbol{a}_i\|, i = 1, 2, 3\} = \{\mathbf{s}^*\}.$$

Therefore $\hat{\mathbf{s}}$=$\mathbf{s}^*$. ■

Lemma 4.2.1 *Let $(\hat{\mathbf{s}}, \hat{y}, \hat{\beta}, \hat{\alpha})$ be an optimal solution of (4.2.1). Then $\|\hat{\mathbf{s}} - \boldsymbol{a}_i\| < \hat{\beta}_i$, $i = 1, 2, 3$ can not be satisfied at the same time.*

Proof. If for all $i = 1,2,3$, $\|\hat{\mathbf{s}} - \boldsymbol{a}_i\| < \hat{\beta}_i$. Then it is clear that $\hat{y} > \|\hat{\mathbf{s}}\|^2$. Let $\delta = \min_{i=1,2,3}\{\hat{\beta}_i - \|\hat{\mathbf{s}} - \boldsymbol{a}_i\|\}$, $\tilde{\beta}_i = \hat{\beta}_i - \delta/2$. Given $\varepsilon > 0$ such that $\hat{y} - \|\hat{\mathbf{s}}\|^2 \geq \varepsilon$ and $(\hat{\beta}_i - \delta/2)^2 \leq \hat{\alpha}_i - \varepsilon,\ i = 1,2,3$. Let $\tilde{\mathbf{s}} = \hat{\mathbf{s}}$, $\tilde{y} = \hat{y} - \varepsilon$, $\tilde{\alpha}_i = \hat{\alpha}_i - \varepsilon$, then $(\tilde{\mathbf{s}}, \tilde{y}, \tilde{\beta}, \tilde{\alpha})$ is also a feasible solution of (4.2.1) and $\tilde{\alpha}_1 + \tilde{\alpha}_2 + \tilde{\alpha}_3 < \hat{\alpha}_1 + \hat{\alpha}_2 + \hat{\alpha}_3$. Thus, $(\hat{\mathbf{s}}, \hat{y}, \hat{\beta}, \hat{\alpha})$ is not optimal. ■

Lemma 4.2.2 *Let* $(\hat{\mathbf{s}}, \hat{y}, \hat{\beta}, \hat{\alpha})$ *be an optimal solution of (4.2.1). Then* $\hat{\beta}_i^2 < \hat{\alpha}_i,\ i = 1,2,3$ *can not be satisfied at the same time.*

Proof. If for all $i = 1,2,3$, $\hat{\beta}_i^2 < \hat{\alpha}_i$. Then it is clear that $\hat{y} > \|\hat{\mathbf{s}}\|^2$. Choose $\delta > 0$, such that $\hat{\alpha}_i - \delta \geq \hat{\beta}_i^2,\ i = 1,2,3$ and $\hat{y} - \|\hat{\mathbf{s}}\|^2 \geq \delta$. Let $\tilde{\mathbf{s}} = \hat{\mathbf{s}}$, $\tilde{y} = \hat{y} - \delta$, $\tilde{\beta}_i = \hat{\beta}_i$, $\tilde{\alpha}_i = \hat{\alpha}_i - \delta$, $i = 1,2,3$. Then $(\tilde{\mathbf{s}}, \tilde{y}, \tilde{\beta}, \tilde{\alpha})$ is also a feasible solution of (4.2.1) and $\tilde{\alpha}_1 + \tilde{\alpha}_2 + \tilde{\alpha}_3 < \hat{\alpha}_1 + \hat{\alpha}_2 + \hat{\alpha}_3$. Therefore, $(\hat{\mathbf{s}}, \hat{y}, \hat{\beta}, \hat{\alpha})$ is not optimal. ■

For simplicity of notations, let us consider the case: $\boldsymbol{a}_1 = (a,0)^T$, $\boldsymbol{a}_2 = (b,0)^T$ and $\boldsymbol{a}_3 = (0,c)^T$ where $a \leq 0$, $b > 0$, $c < 0$. Let Ω be the region enclosed by lines $u_2 = 0$, $u_1 - a = -\frac{a}{c}u_2$ and $u_1 - b = -\frac{b}{c}u_2$, i.e.,

$$\Omega = \left\{ \boldsymbol{u} = (u_1, u_2)^T; u_2 \geq 0, u_1 - a > -\frac{a}{c}u_2 \text{ and } u_1 - b < -\frac{b}{c}u_2, \right\}.$$

Since $(\mathbf{s}^*, y^*, \beta^*, \alpha^*)$ is an optimal solution of (4.2.1) if $\mathbf{s}^*$ is in the triangle region $\Delta_{a_1a_2a_3}$ determined by the three points $\boldsymbol{a}_1$, $\boldsymbol{a}_2$ and $\boldsymbol{a}_3$, we only consider the case: $\mathbf{s}^* \in \Omega$ with $s_2^* > 0$.

For $\lambda \in [0, \min_{1\leq i\leq 3} \beta_i^*]$, then we have

$$B(\boldsymbol{a}_1, \beta_1^* - \lambda) \cap B(\boldsymbol{a}_2, \beta_2^* - \lambda) \subset B(\boldsymbol{a}_3, \beta_3^* - \lambda).$$

where $B(\boldsymbol{a}, r) = \{\boldsymbol{u}, |\boldsymbol{u} - \boldsymbol{a}| \leq r\}$. In particular, $\beta_3^* - \lambda > -c$.

Let $(\hat{\mathbf{s}}, \hat{y}, \hat{\beta}, \hat{\alpha})$ be an optimal solution of (4.2.1). Then there exists $\lambda \in [0, \min_{1\leq i\leq 3} \beta_i^*]$, such that

$$\hat{\beta}_i = \beta_i^* - \lambda, \quad i = 1,2,3;$$

and

$$\hat{\mathbf{s}} \in \cap_{i=1}^{3} B(\boldsymbol{a}_i, \beta_i^* - \lambda) = B(\boldsymbol{a}_1, \beta_1^* - \lambda) \cap B(\boldsymbol{a}_2, \beta_2^* - \lambda).$$

Let $\mathbf{s}^\lambda = (s_1^\lambda, s_2^\lambda)^T$ denote the point such that $s_2^\lambda \geq 0$ and $\|\mathbf{s}^\lambda - \boldsymbol{a}_i\| = \beta_i^* - \lambda$ for $i = 1, 2$. When $\lambda > 0$, $\|\hat{\mathbf{s}} - \boldsymbol{a}_3\|^2 < (\beta_3^* - \lambda)^2$, and if $a \leq \hat{s}_1 \leq b$, then

$$\hat{s}_2 < s_2^\lambda < s_2^*.$$

By Lemma 4.2.1 and Lemma 4.2.2, the one of the following seven cases :

(C1):

$$\begin{cases} \|\hat{\mathbf{s}} - \boldsymbol{a}_1\|^2 = (\beta_1^* - \lambda)^2, \\ \|\hat{\mathbf{s}} - \boldsymbol{a}_2\|^2 = (\beta_2^* - \lambda)^2, \\ \|\hat{\mathbf{s}} - \boldsymbol{a}_3\|^2 \leq (\beta_3^* - \lambda)^2 = \hat{\alpha}_3; \end{cases} \tag{4.2.2}$$

(C2):

$$\begin{cases} \|\hat{\mathbf{s}} - \boldsymbol{a}_1\|^2 = (\beta_1^* - \lambda)^2, \\ (\beta_2^* - \lambda)^2 - \|\hat{\mathbf{s}} - \boldsymbol{a}_2\|^2 = (\beta_3^* - \lambda)^2 - \|\hat{\mathbf{s}} - \boldsymbol{a}_3\|^2 > 0; \end{cases} \tag{4.2.3}$$

(C3):

$$\begin{cases} \|\hat{\mathbf{s}} - \boldsymbol{a}_2\|^2 = (\beta_2^* - \lambda)^2, \\ (\beta_1^* - \lambda)^2 - \|\hat{\mathbf{s}} - \boldsymbol{a}_1\|^2 = (\beta_3^* - \lambda)^2 - \|\hat{\mathbf{s}} - \boldsymbol{a}_3\|^2 > 0; \end{cases} \tag{4.2.4}$$

(C4): $\|\hat{\mathbf{s}} - \boldsymbol{a}_1\|^2 = (\beta_1^* - \lambda)^2$, $\|\hat{\mathbf{s}} - \boldsymbol{a}_2\|^2 < (\beta_2^* - \lambda)^2 = \hat{\alpha}_2$, $\|\hat{\mathbf{s}} - \boldsymbol{a}_3\|^2 \leq (\beta_3^* - \lambda)^2$, and $(\beta_2^* - \lambda)^2 - \|\hat{\mathbf{s}} - \boldsymbol{a}_2\|^2 > (\beta_3^* - \lambda)^2 - \|\hat{\mathbf{s}} - \boldsymbol{a}_3\|^2$;

(C5): $\|\hat{\mathbf{s}} - \boldsymbol{a}_1\|^2 = (\beta_1^* - \lambda)^2$, $\|\hat{\mathbf{s}} - \boldsymbol{a}_2\|^2 < (\beta_2^* - \lambda)^2$, $\|\hat{\mathbf{s}} - \boldsymbol{a}_3\|^2 \leq (\beta_3^* - \lambda)^2 = \hat{\alpha}_3$, and $(\beta_2^* - \lambda)^2 - \|\hat{\mathbf{s}} - \boldsymbol{a}_2\|^2 < (\beta_3^* - \lambda)^2 - \|\hat{\mathbf{s}} - \boldsymbol{a}_3\|^2$;

(C6): $\|\hat{\mathbf{s}} - \boldsymbol{a}_1\|^2 < (\beta_1^* - \lambda)^2 = \hat{\alpha}_1$, $\|\hat{\mathbf{s}} - \boldsymbol{a}_2\|^2 = (\beta_2^* - \lambda)^2$, $\|\hat{\mathbf{s}} - \boldsymbol{a}_3\|^2 \leq (\beta_3^* - \lambda)^2$, and $(\beta_1^* - \lambda)^2 - \|\hat{\mathbf{s}} - \boldsymbol{a}_1\|^2 > (\beta_3^* - \lambda)^2 - \|\hat{\mathbf{s}} - \boldsymbol{a}_3\|^2$;

(C7): $\|\hat{\mathbf{s}} - \boldsymbol{a}_1\|^2 < (\beta_1^* - \lambda)^2$, $\|\hat{\mathbf{s}} - \boldsymbol{a}_2\|^2 = (\beta_2^* - \lambda)^2$, $\|\hat{\mathbf{s}} - \boldsymbol{a}_3\|^2 \leq (\beta_3^* - \lambda)^2 = \hat{\alpha}_3$,and $(\beta_1^* - \lambda)^2 - \|\hat{\mathbf{s}} - \boldsymbol{a}_1\|^2 < (\beta_3^* - \lambda)^2 - \|\hat{\mathbf{s}} - \boldsymbol{a}_3\|^2$.

We want to prove that the cases *(C4)*, *(C5)*, *(C6)* and *(C7)* can not appear. The following theorem is a main result in this section.

Theorem 4.2.2 *(Representation theorem). Assume that* $-c \geq |a+b|$. *Let* $(\hat{\mathbf{s}}, \hat{y}, \hat{\beta}, \hat{\alpha})$ *be an optimal solution of (4.2.1). Then* $\hat{s}_2 \geq 0$ *and there exists* $\lambda \geq 0$ *such that* $\hat{\beta} = \beta^* - \lambda$, *and one of (C1), (C2) and (C3) holds. Furthermore, if* $\hat{s}_1 \in [a, b]$, *then (C1) must be true.*

Proof. The proof will be completed by the following five lemmas.

■

Lemma 4.2.3 *Let* $(\hat{\mathbf{s}}, \hat{y}, \beta^* - \lambda, \hat{\alpha})$ *be a feasible solution of (4.2.1). If* $(\hat{\mathbf{s}}, \hat{y}, \beta^* - \lambda, \hat{\alpha})$ *satisfies (C4) or (C6), then* $(\hat{\mathbf{s}}, \hat{y}, \beta^* - \lambda, \hat{\alpha})$ *is not an optimal solution of (4.2.1).*

Proof. We only consider the case *(C4)* since the case *(C6)* is similar. Assume that $(\hat{\mathbf{s}}, \hat{y}, \beta^* - \lambda, \hat{\alpha})$ satisfies *(C4)*. If $\hat{\mathbf{s}}$ is below of x-axis, let $\hat{\mathbf{s}}^*$ denote the reflection of $\hat{\mathbf{s}}$ with respect to x-axis. Then $\|\hat{\mathbf{s}} - \boldsymbol{a}_2\| = \|\hat{\mathbf{s}}^* - \boldsymbol{a}_2\|$, $\|\hat{\mathbf{s}} - \boldsymbol{a}_1\| = \|\hat{\mathbf{s}}^* - \boldsymbol{a}_1\|$, and

$$\begin{aligned}(\beta_3^* - \lambda)^2 - \|\hat{\mathbf{s}}^* - \boldsymbol{a}_3\|^2 \leq &(\beta_3^* - \lambda)^2 - \|\hat{\mathbf{s}} - \boldsymbol{a}_3\|^2 \\ &< (\beta_2^* - \lambda)^2 - \|\hat{\mathbf{s}} - \boldsymbol{a}_2\|^2 = (\beta_2^* - \lambda)^2 - \|\hat{\mathbf{s}}^* - \boldsymbol{a}_2\|^2.\end{aligned}$$

Set

$$g(\lambda, \boldsymbol{u}) = (\beta_3^* - \lambda)^2 - \|\boldsymbol{u} - \boldsymbol{a}_3\|^2 - \left((\beta_2^* - \lambda)^2 - \|\boldsymbol{u} - \boldsymbol{a}_2\|^2\right), \ \boldsymbol{u} \in \cap_{i=1}^3 B(\boldsymbol{a}_i, \beta_i^* - \lambda).$$

Then $g(\lambda, \mathbf{s}^\lambda) > 0$, $g(\lambda, \hat{\mathbf{s}}) < 0$ and $g(\lambda, \hat{\mathbf{s}}^*) < 0$, thus there exists $\tilde{\mathbf{s}}$ with $s_1^\lambda < \tilde{s}_1 < \hat{s}_1{}^*$, $\tilde{s}_2 > 0$, and $\|\tilde{\mathbf{s}} - \boldsymbol{a}_1\|^2 = (\beta_1^* - \lambda)^2$ and $\|\tilde{\mathbf{s}} - \boldsymbol{a}_2\|^2 < (\beta_2^* - \lambda)^2 = \hat{\alpha}_2$, such that

$$g(\lambda, \tilde{\mathbf{s}}) = 0.$$

Set $\tilde{y} = \|\tilde{\mathbf{s}}\|^2 + (\beta_2^* - \lambda)^2 - \|\tilde{\mathbf{s}} - \boldsymbol{a}_2\|^2$, $\tilde{\beta} = \beta^* - \lambda$ and $\tilde{\alpha}_1 = (\beta_1^* - \lambda)^2 + (\beta_2^* - \lambda)^2 - \|\tilde{\mathbf{s}} - \boldsymbol{a}_2\|^2$, $\tilde{\alpha}_2 = \tilde{\beta}_2^2$, $\tilde{\alpha}_3 = \tilde{\beta}_3^2$. Then $(\tilde{\mathbf{s}}, \tilde{y}, \tilde{\beta}, \tilde{\alpha})$ is a feasible solution of (4.2.1), and $\tilde{\alpha}_1 < (\beta_1^* - \lambda)^2 + (\beta_2^* - \lambda)^2 - \|\hat{\mathbf{s}} - \boldsymbol{a}_2\|^2$, and so

$$\tilde{\alpha}_1 + \tilde{\alpha}_2 + \tilde{\alpha}_3 < \hat{\alpha}_1 + \hat{\alpha}_2 + \hat{\alpha}_3.$$

Thus, $(\hat{\mathbf{s}}, \hat{y}, \beta^* - \lambda, \hat{\alpha})$ is not optimal.

■

Lemma 4.2.4 *Assume that* $-c \geq |a+b|$. *Let* $(\hat{\mathbf{s}}, \hat{y}, \beta^* - \lambda, \hat{\alpha})$ *be a feasible solution of (4.2.1). If* $\hat{s}_2 < 0$ *or* $\hat{s}_1 \in [a, b]$, *and* $(\hat{\mathbf{s}}, \hat{y}, \beta^* - \lambda, \hat{\alpha})$ *satisfies one of (C2) and (C3)*

then $(\hat{\mathbf{s}}, \hat{y}, \beta^* - \lambda, \hat{\alpha})$ *is not an optimal solution of (4.2.1).*

Proof. We only consider the condition (*C2*). If $\hat{\mathbf{s}}$ is below x-axis, denote by $\hat{\mathbf{s}}^*$ the reflection of $\hat{\mathbf{s}}$ with respect to x-axis. Then

$$(\beta_3^* - \lambda)^2 - \|\hat{\mathbf{s}}^* - \boldsymbol{a}_3\|^2 < (\beta_3^* - \lambda)^2 - \|\hat{\mathbf{s}} - \boldsymbol{a}_3\|^2.$$

If $(\beta_3^* - \lambda)^2 - \|\hat{\mathbf{s}}^* - \boldsymbol{a}_3\|^2 \geq (\beta_2^* - \lambda)^2 - \|\hat{\mathbf{s}}^* - \boldsymbol{a}_2\|^2$, set $\tilde{y} = \|\hat{\mathbf{s}}^*\|^2 + (\beta_3^* - \lambda)^2 - \|\hat{\mathbf{s}}^* - \boldsymbol{a}_3\|^2$, $\tilde{\beta} = \beta^* - \lambda$ and $\tilde{\alpha}_1 = (\beta_1^* - \lambda)^2 + (\beta_3^* - \lambda)^2 - \|\hat{\mathbf{s}}^* - \boldsymbol{a}_3\|^2$, $\tilde{\alpha}_2 = \|\hat{\mathbf{s}}^* - \boldsymbol{a}_2\|^2 + (\beta_3^* - \lambda)^2 - \|\hat{\mathbf{s}}^* - \boldsymbol{a}_3\|^2$, $\tilde{\alpha}_3 = \tilde{\beta}_3^2$. Then $(\hat{\mathbf{s}}^*, \tilde{y}, \tilde{\beta}, \tilde{\alpha})$ is a feasible solution of (4.2.1), and $\tilde{\alpha}_i < \hat{\alpha}_i$ for $i = 1, 2$, and so

$$\tilde{\alpha}_1 + \tilde{\alpha}_2 + \tilde{\alpha}_3 < \hat{\alpha}_1 + \hat{\alpha}_2 + \hat{\alpha}_3.$$

Thus, $(\hat{\mathbf{s}}, \hat{y}, \beta^* - \lambda, \hat{\alpha})$ is not optimal.

If $(\beta_3^* - \lambda)^2 - \|\hat{\mathbf{s}}^* - \boldsymbol{a}_3\|^2 < (\beta_2^* - \lambda)^2 - \|\hat{\mathbf{s}}^* - \boldsymbol{a}_2\|^2$, then there exists $\tilde{\mathbf{s}}$ with $\hat{s}_2 < \tilde{s}_2 < \hat{s}_2^*$, $\|\tilde{\mathbf{s}} - \boldsymbol{a}_1\|^2 = (\beta_1^* - \lambda)^2$ and $\|\tilde{\mathbf{s}} - \boldsymbol{a}_2\|^2 < (\beta_2^* - \lambda)^2$, such that $g(\lambda, \tilde{\mathbf{s}}) = 0$. Set $\tilde{y} = \|\tilde{\mathbf{s}}\|^2 + (\beta_2^* - \lambda)^2 - \|\tilde{\mathbf{s}} - \boldsymbol{a}_2\|^2$, $\tilde{\beta} = \beta^* - \lambda$ and $\tilde{\alpha}_1 = (\beta_1^* - \lambda)^2 + (\beta_2^* - \lambda)^2 - \|\tilde{\mathbf{s}} - \boldsymbol{a}_2\|^2$, $\tilde{\alpha}_2 = \tilde{\beta}_2^2$, $\tilde{\alpha}_3 = \tilde{\beta}_3^2$. Then $(\tilde{\mathbf{s}}, \tilde{y}, \tilde{\beta}, \tilde{\alpha})$ is a feasible solution of (4.2.1), and $\tilde{\alpha}_1 < (\beta_1^* - \lambda)^2 + (\hat{\beta}_2 - \lambda)^2 - \|\hat{\mathbf{s}} - \boldsymbol{a}_2\|^2$, and so

$$\tilde{\alpha}_1 + \tilde{\alpha}_2 + \tilde{\alpha}_3 < \hat{\alpha}_1 + \hat{\alpha}_2 + \hat{\alpha}_3,$$

which implies that $(\hat{\mathbf{s}}, \hat{y}, \beta^* - \lambda, \hat{\alpha})$ is also not optimal.

Therefore, $\hat{\mathbf{s}}$ must be above x-axis, i.e., $\hat{s}_2 \geq 0$, and $s_1^\lambda < \hat{s}_1$.

The equation $(\beta_3^* - \lambda)^2 - \|\boldsymbol{u} - \boldsymbol{a}_3\|^2 = (\beta_2^* - \lambda)^2 - \|\boldsymbol{u} - \boldsymbol{a}_2\|^2$ can be written by

$$2cu_2 = 2bu_1 + (\beta_2^* - \lambda)^2 - (\beta_3^* - \lambda)^2 - b^2 + c^2,$$

which is a straight line with a negative slope.

Assume that the straight line intersects the circle $\{\boldsymbol{u}; \|u - \boldsymbol{a}_1\| = \beta_1^* - \lambda\}$ at $\hat{\mathbf{s}}^{(1)} = (\hat{s}_1^{(1)}, \hat{s}_2^{(1)})^T$ and $\hat{\mathbf{s}}^{(2)} = (\hat{s}_1^{(2)}, \hat{s}_2^{(2)})^T$, where $\hat{s}_2^{(1)} \leq \hat{s}_2^{(2)}$.

Without loss of generality, assume that $\hat{s}_2^{(1)} < \hat{s}_2^{(2)}$, then $\hat{s}_2^{(2)} > 0$, and $\hat{\mathbf{s}} = \hat{\mathbf{s}}^{(1)}$ or $\hat{\mathbf{s}} = \hat{\mathbf{s}}^{(2)}$.

If $\hat{\mathbf{s}} = \hat{\mathbf{s}}^{(1)}$, for $\varepsilon > 0$, set

$$u_1^\epsilon = \hat{s}_1 - \varepsilon, \quad u_2^\varepsilon = \frac{b}{c}u_1^\varepsilon + \frac{(\beta_2^* - \lambda)^2 - (\beta_3^* - \lambda)^2 - b^2 + c^2}{2c} = \hat{s}_2 - \frac{b}{c}\varepsilon.$$

Then when ε is small enough, $\boldsymbol{u}^\varepsilon = (u_1^\varepsilon, u_2^\varepsilon)^T \in B(\boldsymbol{a}_1, \beta_1^* - \lambda) \cap B(\boldsymbol{a}_2, \beta_2^* - \lambda)$,

$$\|\boldsymbol{u}^\varepsilon - \boldsymbol{a}_1\|^2 + (\beta_2^* - \lambda)^2 - \|\boldsymbol{u}^\varepsilon - \boldsymbol{a}_2\|^2 > (\beta_1^* - \lambda)^2,$$

and

$$(\beta_3^* - \lambda)^2 - \|\boldsymbol{u}^\varepsilon - \boldsymbol{a}_3\|^2 = (\beta_2^* - \lambda)^2 - \|\boldsymbol{u}^\varepsilon - \boldsymbol{a}_2\|^2 < (\beta_2^* - \lambda)^2 - \|\hat{\mathbf{s}} - \boldsymbol{a}_2\|^2.$$

Set $\tilde{y} = \|\boldsymbol{u}^\varepsilon\|^2 + (\beta_2^* - \lambda)^2 - \|\boldsymbol{u}^\varepsilon - \boldsymbol{a}_2\|^2$, $\tilde{\beta} = \beta^* - \lambda$ and $\tilde{\alpha}_1 = \|\boldsymbol{u}^\varepsilon - \boldsymbol{a}_1\|^2 + (\beta_2^* - \lambda)^2 - \|\boldsymbol{u}^\varepsilon - \boldsymbol{a}_2\|^2$, $\tilde{\alpha}_2 = \tilde{\beta}_2^2$, $\tilde{\alpha}_3 = \tilde{\beta}_3^2$. Then $(\boldsymbol{u}^\varepsilon, \tilde{y}, \tilde{\beta}, \tilde{\alpha})$ is a feasible solution of (4.2.1), and $\tilde{\alpha}_1 < (\beta_1^* - \lambda)^2 + (\beta_2^* - \lambda)^2 - \|\hat{\mathbf{s}} - \boldsymbol{a}_2\|^2$, and so

$$\tilde{\alpha}_1 + \tilde{\alpha}_2 + \tilde{\alpha}_3 < \hat{\alpha}_1 + \hat{\alpha}_2 + \hat{\alpha}_3.$$

Therefore, $(\hat{\mathbf{s}}, \hat{y}, \beta, \hat{\alpha})$ is not optimal.

If $\hat{\mathbf{s}} = \hat{\mathbf{s}}^{(2)}$ and $\hat{s}_1 \geq a$, for $\varepsilon > 0$, set

$$u_1^\varepsilon = \hat{s}_1 - \varepsilon, \quad u_2^\varepsilon = \hat{s}_2 - \varepsilon.$$

Then

$$\|\boldsymbol{u}^\varepsilon - \boldsymbol{a}_1\|^2 = (\beta_1^* - \lambda)^2 - 2\varepsilon(|\hat{s}_1 - a| + \hat{s}_2) + O(\varepsilon^2),$$

and so, for ε small enough, $\boldsymbol{u}^\varepsilon \in B(\boldsymbol{a}_1, \beta_1^* - \lambda)$. Set

$$\lambda^\varepsilon = \lambda + (\beta_1^* - \|\boldsymbol{u}^\varepsilon - \boldsymbol{a}_1\|) \frac{\beta_1^* - \lambda}{|\hat{s}_1 - a| + \hat{s}_2}.$$

Then

$$\lambda^\varepsilon = \lambda + \varepsilon + O(\varepsilon^2),$$

and

$$\frac{b}{c}u_1^\varepsilon + \frac{(\beta_2^* - \lambda^\varepsilon)^2 - (\beta_3^* - \lambda^\varepsilon)^2 - b^2 + c^2}{2c} + O(\varepsilon^2) = \hat{s}_2 - \frac{\beta_2^* - \beta_3^* + b}{c}\varepsilon.$$

Since $|\beta_2^* - \beta_3^*| < \sqrt{b^2 + c^2} < b - c$, we have $\beta_2^* - \beta_3^* + b > c$, i.e., $\frac{\beta_2^* - \beta_3^* + b}{-c} > -1$ which yields

$$u_2^\varepsilon < \frac{b}{c}u_1^\varepsilon + \frac{(\beta_2^* - \lambda^\varepsilon)^2 - (\beta_3^* - \lambda^\varepsilon)^2 - b^2 + c^2}{2c}.$$

Noting that $\hat{s}_2 > 0$ and $s_1^\lambda < \hat{s}_1$, we see that $(\beta_2^* - \lambda)^2 > \|\hat{\mathbf{s}} - \boldsymbol{a}_2\|^2$. Then we can choose $\varepsilon_0 > 0$ small enough such that for all $\varepsilon \in (0, \varepsilon_0)$, $\boldsymbol{u}^\varepsilon \in B(\boldsymbol{a}_1, \beta_1^* - \lambda^\varepsilon) \cap B(\boldsymbol{a}_2, \beta_2^* - \lambda^\varepsilon)$,

$$\|\boldsymbol{u}^\varepsilon - \boldsymbol{a}_1\|^2 + (\beta_3^* - \lambda^\varepsilon)^2 - \|\boldsymbol{u}^\varepsilon - \boldsymbol{a}_3\|^2 > (\beta_1^* - \lambda^\varepsilon)^2,$$

and

$$u_2^\varepsilon < \frac{b}{c}u_1^\varepsilon + \frac{(\beta_2^* - \lambda^\varepsilon)^2 - (\beta_3^* - \lambda^\varepsilon)^2 - b^2 + c^2}{2c}.$$

Noting that

$$(\beta_3^* - \lambda^\varepsilon)^2 - \|\boldsymbol{u} - \boldsymbol{a}_3\|^2 > (\beta_2^* - \lambda^\varepsilon)^2 - \|\boldsymbol{u} - \boldsymbol{a}_2\|^2,$$

if and only if

$$u_2 < \frac{b}{c}u_1 + \frac{(\beta_2^* - \lambda^\varepsilon)^2 - (\beta_3^* - \lambda^\varepsilon)^2 - b^2 + c^2}{2c},$$

we have that

$$(\beta_3^* - \lambda^\varepsilon)^2 - \|\boldsymbol{u}^\varepsilon - \boldsymbol{a}_3\|^2 > (\beta_2^* - \lambda^\varepsilon)^2 - \|\boldsymbol{u}^\varepsilon - \boldsymbol{a}_2\|^2.$$

Set $\tilde{y} = \|\boldsymbol{u}^\varepsilon\|^2 + (\beta_3^* - \lambda^\varepsilon)^2 - \|\boldsymbol{u}^\varepsilon - \boldsymbol{a}_3\|^2$, $\tilde{\beta} = \beta^* - \lambda^\varepsilon$ and $\tilde{\alpha}_1 = \|\boldsymbol{u}^\varepsilon - \boldsymbol{a}_1\|^2 + (\beta_3^* - \lambda^\varepsilon)^2 - \|\boldsymbol{u}^\varepsilon - \boldsymbol{a}_3\|^2$, $\tilde{\alpha}_2 = \|\boldsymbol{u}^\varepsilon - \boldsymbol{a}_2\|^2 + (\beta_3^* - \lambda^\varepsilon)^2 - \|\boldsymbol{u}^\varepsilon - \boldsymbol{a}_3\|^2$, $\tilde{\alpha}_3 = \tilde{\beta}_3^2$. Then $(\boldsymbol{u}^\varepsilon, \tilde{y}, \tilde{\beta}, \tilde{\alpha})$ is a feasible solution of (4.2.1). Since

$$\|\boldsymbol{u}^\varepsilon - \boldsymbol{a}_1\|^2 - \|\hat{\mathbf{s}} - \boldsymbol{a}_1\|^2 = -2\varepsilon\left((\hat{s}_1 - a) + \hat{s}_2\right) + O(\varepsilon^2),$$

$$\|\boldsymbol{u}^\varepsilon - \boldsymbol{a}_2\|^2 - \|\hat{\mathbf{s}} - \boldsymbol{a}_2\|^2 = -2\varepsilon\left((\hat{s}_1 - b) + \hat{s}_2\right) + O(\varepsilon^2),$$

$$\|\boldsymbol{u}^\varepsilon - \boldsymbol{a}_3\|^2 - \|\hat{\mathbf{s}} - \boldsymbol{a}_3\|^2 = -2\varepsilon\left(\hat{s}_1 + (\hat{s}_2 - c)\right) + O(\varepsilon^2),$$

and

$$(\beta_3^* - \lambda^\varepsilon)^2 - (\beta_3^* - \lambda)^2 = -2\varepsilon(\beta_3^* - \lambda) + O(\varepsilon^2),$$

we have that

$$\begin{aligned}
&\tilde{\alpha}_1 + \tilde{\alpha}_2 + \tilde{\alpha}_3 - (\hat{\alpha}_1 + \hat{\alpha}_2 + \hat{\alpha}_3)\\
=&\|\boldsymbol{u}^\varepsilon - \boldsymbol{a}_1\|^2 - \|\hat{\mathbf{s}} - \boldsymbol{a}_1\|^2 + \|\boldsymbol{u}^\varepsilon - \boldsymbol{a}_2\|^2 - \|\hat{\mathbf{s}} - \boldsymbol{a}_2\|^2 - 2\left(\|\boldsymbol{u}^\varepsilon - \boldsymbol{a}_3\|^2 - \|\hat{\mathbf{s}} - \boldsymbol{a}_3\|^2\right)\\
&+ 3\left((\beta_3^* - \lambda^\varepsilon)^2 - (\beta_3^* - \lambda)^2\right)\\
=&-4c\varepsilon + 2\varepsilon(a+b) - 6\varepsilon(\beta_3^* - \lambda) + O(\varepsilon^2).
\end{aligned}$$

Therefore, under the condition $-c \geq |a+b|$, when ε is small enough, $\tilde{\alpha}_1 + \tilde{\alpha}_2 + \tilde{\alpha}_3 < \hat{\alpha}_1 + \hat{\alpha}_2 + \hat{\alpha}_3$, and so, $(\hat{\mathbf{s}}, \hat{y}, \beta^* - \lambda, \hat{\alpha})$ is also not optimal. This is a contradiction.

■

Lemma 4.2.5 *Assume that* $-3c \geq |a+b|$. *Let* $(\hat{\mathbf{s}}, \hat{y}, \beta^* - \lambda, \hat{\alpha})$ *be a feasible solution of (4.2.1). If* $\hat{s}_2 = 0$, *and one of (C5) and (C7), then* $(\hat{\mathbf{s}}, \hat{y}, \beta^* - \lambda, \hat{\alpha})$ *is not an optimal solution of (4.2.1).*

Proof. We only consider the condition (*C5*). Suppose that $\hat{s}_2 = 0$, then $\hat{s}_1 = \beta_1^* - \lambda + a$. If $(\beta_3^* - \lambda)^2 - \|\hat{\mathbf{s}} - \boldsymbol{a}_3\|^2 > (\beta_2^* - \lambda)^2 - \|\hat{\mathbf{s}} - \boldsymbol{a}_2\|^2$, then

$$\begin{aligned}
&\hat{\alpha}_1 + \hat{\alpha}_2 + \hat{\alpha}_3\\
=&\|\hat{\mathbf{s}} - \boldsymbol{a}_1\|^2 + \|\hat{\mathbf{s}} - \boldsymbol{a}_2\|^2 + 2((\beta_3^* - \lambda)^2 - \|\hat{\mathbf{s}} - \boldsymbol{a}_3\|^2) + (\beta_3^* - \lambda)^2\\
=&(\beta_1^* - \lambda)^2 + (\beta_1^* - \lambda + a - b)^2 + 2((\beta_3^* - \lambda)^2 - (\beta_1^* - \lambda + a)^2 - c^2) + (\beta_3^* - \lambda)^2.
\end{aligned}$$

Choose $\varepsilon_0 > 0$ such that for any $\varepsilon \in [0, \varepsilon_0)$,

$$(\beta_3^* - \lambda^\varepsilon)^2 - \|\hat{\mathbf{s}}^\varepsilon - \boldsymbol{a}_3\|^2 > (\beta_2^* - \lambda^\varepsilon)^2 - \|\hat{\mathbf{s}}^\varepsilon - \boldsymbol{a}_2\|^2,$$

where $\lambda^\varepsilon = \lambda + \varepsilon$ and $\hat{\mathbf{s}}^\varepsilon = (\hat{s}_1 - \varepsilon, 0)^T$. Set $\tilde{y} = \|\hat{\mathbf{s}}^\varepsilon\|^2 + (\beta_3^* - \lambda^\varepsilon)^2 - \|\hat{\mathbf{s}}^\varepsilon - \boldsymbol{a}_3\|^2$, $\tilde{\beta} = \beta^* - \lambda^\varepsilon$ and $\tilde{\alpha}_1 = \|\hat{\mathbf{s}}^\varepsilon - \boldsymbol{a}_1\|^2 + (\beta_3^* - \lambda^\varepsilon)^2 - \|\hat{\mathbf{s}}^\varepsilon - \boldsymbol{a}_3\|^2$, $\tilde{\alpha}_2 = \|\hat{\mathbf{s}}^\varepsilon - \boldsymbol{a}_2\|^2 + (\beta_3^* - \lambda^\varepsilon)^2 - \|\hat{\mathbf{s}}^\varepsilon - \boldsymbol{a}_3\|^2$, $\tilde{\alpha}_3 = \tilde{\beta}_3^2$. Then

$$\begin{aligned}
&\tilde{\alpha}_1 + \tilde{\alpha}_2 + \tilde{\alpha}_3 \\
=&\|\hat{\mathbf{s}} - \boldsymbol{a}_1\|^2 + \|\hat{\mathbf{s}} - \boldsymbol{a}_2\|^2 + 2((\beta_3^* - \lambda)^2 - \|\hat{\mathbf{s}} - \boldsymbol{a}_3\|^2) + (\beta_3^* - \lambda)^2 \\
&- 2\varepsilon(\beta_1^* - \lambda) + \varepsilon^2 - 2\varepsilon(\beta_1^* - \lambda + a - b) + \varepsilon^2 - 6\varepsilon(\beta_3^* - \lambda) + 3\epsilon^2 \\
&+ 4\varepsilon(\beta_1^* - \lambda + a) - 2\varepsilon^2 \\
=&\hat{\alpha}_1 + \hat{\alpha}_2 + \hat{\alpha}_3 + 2\varepsilon\left(a + b - 3(\beta_3^* - \lambda)\right) + 3\varepsilon^2
\end{aligned}$$

Therefore, noting that $\beta_3^* - \lambda > -c$, under the condition $-3c \geq |a + b|$, when ε is small enough, $\tilde{\alpha}_1 + \tilde{\alpha}_2 + \tilde{\alpha}_3 < \hat{\alpha}_1 + \hat{\alpha}_2 + \hat{\alpha}_3$. Therefore, $(\hat{\mathbf{s}}, \hat{y}, \beta^* - \lambda, \hat{\alpha})$ is not optimal.

■

Lemma 4.2.6 *Assume that* $-c \geq |a+b|$. *Let* $(\hat{\mathbf{s}}, \hat{y}, \beta^* - \lambda, \hat{\alpha})$ *be a feasible solution of (4.2.1). If one of (C5) and (C7), then* $(\hat{\mathbf{s}}, \hat{y}, \beta^* - \lambda, \hat{\alpha})$ *is not an optimal solution of (4.2.1).*

Proof. We only consider the condition (*C5*). Suppose that $(\hat{\mathbf{s}}, \hat{y}, \beta^* - \lambda, \hat{\alpha})$ is an optimal solution of (4.2.1). Then by Lemma 4.2.5, $\hat{s}_2 > 0$, and $\hat{\mathbf{s}}$ is above x-axis by a similar argument in the proof of Lemma 4.2.4. Thus, $\hat{\mathbf{s}}$ is above x-axis with $\tilde{s}_2 > 0$ and $s_1^\lambda < \hat{s}_1$.

For $\varepsilon > 0$, set

$$u_1^\varepsilon = \hat{s}_1 - \frac{\varepsilon(\hat{s}_1 - a)}{|\hat{s}_1 - a|} \quad u_2^\varepsilon = \hat{s}_2 - \varepsilon.$$

Then

$$\|\boldsymbol{u}^\varepsilon - \boldsymbol{a}_1\|^2 = (\beta_1^* - \lambda)^2 - 2\varepsilon\left(|\hat{s}_1 - a| + \hat{s}_2\right) + 2\varepsilon^2 + O(\varepsilon^2),$$

and so, for ε small enough, $\boldsymbol{u}^\varepsilon \in B(\boldsymbol{a}_1, \beta_1^* - \lambda)$. Set

$$\lambda^\varepsilon = \lambda + (\beta_1^* - \|\boldsymbol{u}^\varepsilon - \boldsymbol{a}_1\|)\,\frac{\beta_1^* - \lambda}{|\hat{s}_1 - a| + \hat{s}_2}.$$

Then

$$\lambda^\varepsilon = \lambda + \varepsilon + O(\varepsilon^2),$$

Noting that $\tilde{s}_2 > 0$ and $s_1^\lambda < \hat{s}_1$, we see that $(\beta_2^* - \lambda)^2 > \|\hat{\mathbf{s}} - \boldsymbol{a}_2\|^2$. Then we can choose $\varepsilon_0 > 0$ small enough such that for all $\varepsilon \in (0, \varepsilon_0)$, $\boldsymbol{u}^\varepsilon \in B(\boldsymbol{a}_2, \beta_2^* - \lambda^\varepsilon)$.

Set $\tilde{y} = \|\boldsymbol{u}^\varepsilon\|^2 + (\beta_3^* - \lambda^\varepsilon)^2 - \|\boldsymbol{u}^\varepsilon - \boldsymbol{a}_3\|^2$, $\tilde{\beta} = \beta^* - \lambda^\varepsilon$ and $\tilde{\alpha}_1 = \|\boldsymbol{u}^\varepsilon - \boldsymbol{a}_1\|^2 + (\beta_3^* - \lambda^\varepsilon)^2 - \|\boldsymbol{u}^\varepsilon - \boldsymbol{a}_3\|^2$, $\tilde{\alpha}_2 = \|\boldsymbol{u}^\varepsilon - \boldsymbol{a}_2\|^2 + (\beta_3^* - \lambda^\varepsilon)^2 - \|\boldsymbol{u}^\varepsilon - \boldsymbol{a}_3\|^2$, $\tilde{\alpha}_3 = \tilde{\beta}_3^2$. Then $(\boldsymbol{u}^\varepsilon, \tilde{y}, \tilde{\beta}, \tilde{\alpha})$ is a feasible solution of (4.2.1). Since

$$\|\boldsymbol{u}^\varepsilon - \boldsymbol{a}_1\|^2 - \|\hat{\mathbf{s}} - \boldsymbol{a}_1\|^2 = -2\varepsilon\left(\frac{(\hat{s}_1 - a)}{|\hat{s}_1 - a|}(\hat{s}_1 - a) + \hat{s}_2\right) + O(\varepsilon^2),$$

$$\|\boldsymbol{u}^\varepsilon - \boldsymbol{a}_2\|^2 - \|\hat{\mathbf{s}} - \boldsymbol{a}_2\|^2 = -2\varepsilon\left(\frac{(\hat{s}_1 - a)}{|\hat{s}_1 - a|}(\hat{s}_1 - b) + \hat{s}_2\right) + O(\varepsilon^2),$$

$$\|\boldsymbol{u}^\varepsilon - \boldsymbol{a}_3\|^2 - \|\hat{\mathbf{s}} - \boldsymbol{a}_3\|^2 = -2\varepsilon\left(\frac{(\hat{s}_1 - a)}{|\hat{s}_1 - a|}\hat{s}_1 + (\hat{s}_2 - c)\right) + O(\varepsilon^2),$$

and

$$(\beta_3^* - \lambda^\varepsilon)^2 - (\beta_3^* - \lambda)^2 = -2\varepsilon(\beta_3^* - \lambda) + O(\varepsilon^2),$$

we have that

$$\begin{aligned}
&\tilde{\alpha}_1 + \tilde{\alpha}_2 + \tilde{\alpha}_3 - (\hat{\alpha}_1 + \hat{\alpha}_2 + \hat{\alpha}_3)\\
=&\|\boldsymbol{u}^\varepsilon - \boldsymbol{a}_1\|^2 - \|\hat{\mathbf{s}} - \boldsymbol{a}_1\|^2 + \|\boldsymbol{u}^\varepsilon - \boldsymbol{a}_2\|^2 - \|\hat{\mathbf{s}} - \boldsymbol{a}_2\|^2 - 2\left(\|\boldsymbol{u}^\varepsilon - \boldsymbol{a}_3\|^2 - \|\hat{\mathbf{s}} - \boldsymbol{a}_3\|^2\right)\\
&+ 3\left((\beta_3^* - \lambda^\varepsilon)^2 - (\beta_3^* - \lambda)^2\right)\\
=&-4c\varepsilon + 2(a+b)\frac{\varepsilon(\hat{s}_1 - a)}{|\hat{s}_1 - a|} - 6\varepsilon(\beta_3^* - \lambda) + O(\varepsilon^2).
\end{aligned}$$

Therefore, under the condition $-c \geq |a+b|$, when ε is small enough, $\tilde{\alpha}_1 + \tilde{\alpha}_2 + \tilde{\alpha}_3 < \hat{\alpha}_1 + \hat{\alpha}_2 + \hat{\alpha}_3$, and so, $(\hat{\mathbf{s}}, \hat{y}, \beta^* - \lambda, \hat{\alpha})$ is also not optimal. This is a contradiction.

■

Lemma 4.2.7 *Let $(\hat{\mathbf{s}}, \hat{y}, \beta^* - \lambda, \hat{\alpha})$ be an optimal solution of (4.2.1). If $(\hat{\mathbf{s}}, \hat{y}, \beta^* - \lambda, \hat{\alpha})$ satisfies (C1), then $\hat{s}_2 > 0$.*

Proof. If $\hat{s}_2 < 0$, denote by $\hat{\mathbf{s}}^*$ the reflection of $\hat{\mathbf{s}}$ with respect to x-axis. Then

$$\|\hat{\mathbf{s}}^* - \boldsymbol{a}_1\|^2 = (\beta_1^* - \lambda)^2, \quad \|\hat{\mathbf{s}}^* - \boldsymbol{a}_2\|^2 = (\beta_2^* - \lambda)^2, \quad \|\hat{\mathbf{s}}^* - \boldsymbol{a}_3\|^2 \leq (\beta_3^* - \lambda)^2$$

and

$$(\beta_3^* - \lambda)^2 - \|\hat{\mathbf{s}}^* - \boldsymbol{a}_3\|^2 < (\beta_3^* - \lambda)^2 - \|\hat{\mathbf{s}} - \boldsymbol{a}_3\|^2.$$

Set $\tilde{y} = \|\hat{\mathbf{s}}^*\|^2 + (\beta_3^* - \lambda)^2 - \|\hat{\mathbf{s}}^* - \boldsymbol{a}_3\|^2$, $\tilde{\beta} = \beta^* - \lambda$ and $\tilde{\alpha}_1 = (\beta_1^* - \lambda)^2 + (\beta_3^* - \lambda)^2 - \|\hat{\mathbf{s}}^* - \boldsymbol{a}_3\|^2$, $\tilde{\alpha}_2 = \|\hat{\mathbf{s}}^* - \boldsymbol{a}_2\|^2 + (\beta_3^* - \lambda)^2 - \|\hat{\mathbf{s}}^* - \boldsymbol{a}_3\|^2$, $\tilde{\alpha}_3 = \tilde{\beta}_3^2$. Then $(\hat{\mathbf{s}}^*, \tilde{y}, \tilde{\beta}, \tilde{\alpha})$ is a feasible solution of (4.2.1), and $\tilde{\alpha}_i < \hat{\alpha}_i$ for $i = 1, 2$, and so

$$\tilde{\alpha}_1 + \tilde{\alpha}_2 + \tilde{\alpha}_3 < \hat{\alpha}_1 + \hat{\alpha}_2 + \hat{\alpha}_3.$$

Thus, $(\hat{\mathbf{s}}, \hat{y}, \beta^* - \lambda, \hat{\alpha})$ is not optimal.

If $\hat{s}_2 = 0$, then $\hat{s}_1 = \beta_1^* - \lambda + a$ and $\hat{s}_1 = \beta_2^* - \lambda - b$. Therefore, $b - a = \beta_2^* - \beta_1^*$. This is a contradiction of that $b - a, \beta_1^*, \beta_2^*$ are three edges of the triangle $\triangle_{\boldsymbol{a}_1\boldsymbol{a}_2\mathbf{s}^*}$.
■

4.3 A characterization theorem

In this section, based on the representation theorem derived in the previous section, we continue to study the characteristics of the solution to the mixed SDP-SOCP model. In particular, we would like to know the geometric structure for the localizable region, where the source can be located exactly when there is no error in the delay estimates. This gives a standard to compare the performance of different convex relaxation models.

Technically, we want to find a condition such that $(\mathbf{s}^*, y^*, \beta^*, \alpha^*)$ is an optimal solution to (4.2.1) where $s_1^* \in [a, b]$. For simplicity of notations, let us first consider $\boldsymbol{a}_1 = (a, 0)^T$, $\boldsymbol{a}_2 = (b, 0)^T$ and $\boldsymbol{a}_3 = (0, c)^T$ where $a < 0$, $b > 0$, $c < 0$.

Let $(\hat{\mathbf{s}}, \hat{y}, \hat{\beta}, \hat{\alpha})$ be an optimal solution of (4.2.1) and assume that $\hat{s}_1 \in [a, b]$. Then by Theorem 4.2.2, $\hat{s}_2 > 0$ and $\hat{\mathbf{s}}$ is a solutions of the equation (4.2.2).

Define

$$D = \{\lambda \in (0, \min\{\beta_1^*, \beta_2^*, \beta_3^*\}]\,;\ \text{such that (4.2.2) has a solution}\}\,. \tag{4.3.5}$$

Now, assume $\hat{\mathbf{s}}$ to be a solution of the equation (4.2.2). Define

$$M(\lambda) :=(\beta_1^* - \lambda)^2 + (\beta_3^* - \lambda)^2 + \|\hat{\mathbf{s}} - \boldsymbol{a}_2\|^2 + 2((\beta_3^* - \lambda)^2 - \|\hat{\mathbf{s}} - \boldsymbol{a}_3\|^2), \lambda \in D.$$

It follows from Theorem 4.2.2 that $(\mathbf{s}^*, y^*, \beta^*, \alpha^*)$ is an optimal solution of (4.2.1) if and only if

$$g(\lambda) := M(\lambda) - ((\beta_1^*)^2 + (\beta_2^*)^2 + (\beta_3^*)^2) > 0 \text{ for any } \lambda \in D. \tag{4.3.6}$$

Set

$$U = \left\{\mathbf{s}^* = (s_1^*, s_2^*)^T; g(\lambda) > 0 \text{ for all } \lambda \in D\right\}. \tag{4.3.7}$$

The following theorem gives a characterization such that the mixed SDP-SOCP (4.2.1) has an exact relaxation under the condition $-c \geq |a+b|$.

Theorem 4.3.1 *(Characterization theorem). Assume that $-c \geq |a+b|$. Let $(\hat{\mathbf{s}}, \hat{y}, \hat{\beta}, \hat{\alpha})$ be an optimal solution of (4.2.1) and assume that $\hat{s}_1 \in [a, b]$. If*

$$\begin{aligned}\Psi(\mathbf{s}^*) := &-\left((\beta_1^*)^2 - (\beta_3^*)^2 + \frac{a((\beta_1^*)^2 - (\beta_2^*)^2)}{b-a} + ab + c^2\right)\\ &\left(\beta_1^* - \beta_2^* - 3\beta_3^* + \frac{2a(\beta_1^* - \beta_2^*)}{b-a}\right)\\ &- 2c^2\left(-2\,\beta_1^* + \frac{((\beta_1^*)^2 - (\beta_2^*)^2 + (b-a)^2)(\beta_1^* - \beta_2^*)}{(b-a)^2}\right) > 0.\end{aligned} \tag{4.3.8}$$

then $\hat{\mathbf{s}} = \mathbf{s}^*$, i.e., the mixed SDP-SOCP (4.2.1) has an exact relaxation.

Conversely, if the mixed SDP-SOCP (4.2.1) has an exact relaxation, then $\Psi(\mathbf{s}^*) \geq 0$.

Proof. By $\|\hat{\mathbf{s}} - \boldsymbol{a}_1\|^2 = (\beta_1^* - \lambda)^2$ and $\|\hat{\mathbf{s}} - \boldsymbol{a}_2\|^2 = (\beta_2^* - \lambda)^2$, we have that

$$(b-a)\,(2\hat{s}_1 - (a+b)) =(\beta_1^* - \lambda)^2 - (\beta_2^* - \lambda)^2 = (\beta_1^* - \beta_2^*)\,(\beta_1^* + \beta_2^* - 2\lambda)\,,$$

which yields

$$\hat{s}_1 = \frac{(\beta_1^* - \beta_2^*)}{2(b-a)}(\beta_1^* + \beta_2^* - 2\lambda) + \frac{a+b}{2}.$$

Therefore,

$$\hat{s}_2^2 = (\beta_1^* - \lambda)^2 - \left(\frac{(\beta_1^* - \beta_2^*)}{2(b-a)}(\beta_1^* + \beta_2^* - 2\lambda) + \frac{b-a}{2}\right)^2.$$

Noting that $\hat{s}_2 > 0$, we obtain

$$\hat{s}_2 = \sqrt{(\beta_1^* - \lambda)^2 - \left(\frac{(\beta_1^* - \beta_2^*)}{2(b-a)}(\beta_1^* + \beta_2^* - 2\lambda) + \frac{b-a}{2}\right)^2}.$$

By some algebraic manipulations, we have

$$\begin{aligned}&(\beta_1^* - \lambda)^2 - \left(\frac{(\beta_1^* - \beta_2^*)}{2(b-a)}(\beta_1^* + \beta_2^* - 2\lambda) + \frac{b-a}{2}\right)^2\\ =&\frac{1}{(b-a)^2}\left((b-a)^2 - (\beta_1^* - \beta_2^*)^2\right)\left(\lambda - \frac{\beta_1^* + \beta_2^* - (b-a)}{2}\right)\left(\lambda - \frac{\beta_1^* + \beta_2^* + (b-a)}{2}\right)\end{aligned}$$

Noting that the sum of the lengths of any two sides of a triangle is always greater than the length of the third one, we see that

$$0 < \frac{\beta_1^* + \beta_2^* - (b-a)}{2} < \min\{\beta_1^*, \beta_2^*\}.$$

Let $\hat{\mathbf{s}}_0$ denote the point of intersection of the lines $\boldsymbol{a}_3\mathbf{s}^*$ and $\boldsymbol{a}_1\boldsymbol{a}_2$. Then we also have that

$$\beta_1^* - \|\boldsymbol{a}_1 - \hat{\mathbf{s}}_0\| < \beta_3^*,\ \beta_2^* - \|\boldsymbol{a}_2 - \hat{\mathbf{s}}_0\| < \beta_3^*,\ \text{and}\ \|\boldsymbol{a}_1 - \hat{\mathbf{s}}_0\| + \|\boldsymbol{a}_2 - \hat{\mathbf{s}}_0\| = b - a.$$

Thus

$$\frac{\beta_1^* + \beta_2^* - (b-a)}{2} < \beta_3^*,$$

and so

$$0 < \frac{\beta_1^* + \beta_2^* - (b-a)}{2} < \min\{\beta_1^*, \beta_2^*, \beta_3^*\}.$$

Thus, when $0 < \lambda < \frac{\beta_1^* + \beta_2^* - (b-a)}{2}$,

$$(\beta_1^* - \lambda)^2 - \left(\frac{(\beta_1^* - \beta_2^*)}{2(b-a)}(\beta_1^* + \beta_2^* - 2\lambda) + \frac{b-a}{2}\right)^2 > 0;$$

and when $\frac{\beta_1^*+\beta_2^*-(b-a)}{2} < \lambda < \frac{\beta_1^*+\beta_2^*+(b-a)}{2}$,

$$(\beta_1^* - \lambda)^2 - \left(\frac{(\beta_1^* - \beta_2^*)}{2(b-a)} (\beta_1^* + \beta_2^* - 2\lambda) + \frac{b-a}{2} \right)^2 < 0.$$

Since $\min\{\beta_1^*, \beta_2^*, \beta_3^*\} < \frac{\beta_1^*+\beta_2^*+(b-a)}{2}$, from $0 \le \lambda < \min\{\beta_1^*, \beta_2^*, \beta_3^*\}$ and

$$(\beta_1^* - \lambda)^2 - \left(\frac{(\beta_1^* - \beta_2^*)}{2(b-a)} (\beta_1^* + \beta_2^* - 2\lambda) + \frac{b-a}{2} \right)^2 \ge 0,$$

we obtain $\lambda < \frac{\beta_1^*+\beta_2^*-(b-a)}{2}$, that is, $D = \left(0, \frac{\beta_1^*+\beta_2^*-(b-a)}{2}\right)$.

Now, by

$$\begin{aligned}
&(\beta_3^* - \lambda)^2 - \|\hat{\mathbf{s}} - a_3\|^2 \\
=&(\beta_3^* - \lambda)^2 - \hat{s}_1^2 - \hat{s}_2^2 + 2c\hat{s}_2 - c^2 \\
=&(\beta_3^* - \lambda)^2 - \left(\frac{(\beta_1^* - \beta_2^*)}{2(b-a)} (\beta_1^* + \beta_2^* - 2\lambda) + \frac{a+b}{2} \right)^2 \\
&- (\beta_1^* - \lambda)^2 + \left(\frac{(\beta_1^* - \beta_2^*)}{2(b-a)} (\beta_1^* + \beta_2^* - 2\lambda) + \frac{b-a}{2} \right)^2 - c^2 \\
&+ 2c\sqrt{(\beta_1^* - \lambda)^2 - \left(\frac{(\beta_1^* - \beta_2^*)}{2(b-a)} (\beta_1^* + \beta_2^* - 2\lambda) + \frac{b-a}{2} \right)^2} \\
=&-(\beta_1^* - \beta_3^*)(\beta_1^* + \beta_3^* - 2\lambda) - a\left(\frac{(\beta_1^* - \beta_2^*)}{b-a} (\beta_1^* + \beta_2^* - 2\lambda) + b \right) - c^2 \\
&+ 2c\sqrt{(\beta_1^* - \lambda)^2 - \left(\frac{(\beta_1^* - \beta_2^*)}{2(b-a)} (\beta_1^* + \beta_2^* - 2\lambda) + \frac{b-a}{2} \right)^2},
\end{aligned}$$

we obtain

$$\begin{aligned}
g(\lambda) =&3\lambda^2 - 2\lambda(\beta_1^* + \beta_2^* + \beta_3^*) - 2(\beta_1^* - \beta_3^*)(\beta_1^* + \beta_3^* - 2\lambda) \\
&- 2a\left(\frac{(\beta_1^* - \beta_2^*)}{b-a} (\beta_1^* + \beta_2^* - 2\lambda) + b \right) - 2c^2 \\
&+ 4c\sqrt{(\beta_1^* - \lambda)^2 - \left(\frac{(\beta_1^* - \beta_2^*)}{2(b-a)} (\beta_1^* + \beta_2^* - 2\lambda) + \frac{b-a}{2} \right)^2}.
\end{aligned}$$

Therefore

$$\{\mathbf{s}^* = (\hat{s}_1, \hat{s}_2); \tilde{g}(\lambda) > 0 \text{ for any } \lambda \in D\} = U,$$

where

$$
\begin{aligned}
\tilde{g}(\lambda) :=& \Bigg(3\lambda^2 - 2\lambda \left(\beta_1^* + \beta_2^* + \beta_3^*\right) - 2(\beta_1^* - \beta_3^*)(\beta_1^* + \beta_3^* - 2\lambda) \\
& - 2a \left(\frac{\beta_1^* - \beta_2^*}{b-a} \left(\beta_1^* + \beta_2^* - 2\lambda\right) + b \right) - 2c^2 \Bigg)^2 \\
& -16c^2 \left((\beta_1^* - \lambda)^2 - \left(\frac{\beta_1^* - \beta_2^*}{2(b-a)} \left(\beta_1^* + \beta_2^* - 2\lambda\right) + \frac{b-a}{2} \right)^2 \right).
\end{aligned}
$$

Notice that

$$
\tilde{g}(0) = 0, \quad \tilde{g}\left(\frac{\beta_1^* + \beta_2^* - (b-a)}{2} \right) \geq 0,
$$

and for any $\frac{\beta_1^*+\beta_2^*-(b-a)}{2} \leq \lambda \leq \frac{\beta_1^*+\beta_2^*+(b-a)}{2}$,

$$
(\beta_1^* - \lambda)^2 - \left(\frac{\beta_1^* - \beta_2^*}{2(b-a)} \left(\beta_1^* + \beta_2^* - 2\lambda\right) + \frac{b-a}{2} \right)^2 \leq 0,
$$

which implies that for any $\frac{\beta_1^*+\beta_2^*-(b-a)}{2} \leq \lambda \leq \frac{\beta_1^*+\beta_2^*+(b-a)}{2}$,

$$
\tilde{g}(\lambda) \geq 0.
$$

Therefore, we can write

$$
\tilde{g}(\lambda) = \lambda \left(9\lambda^3 + a_1\lambda^2 + a_2\lambda + a_3\right),
$$

where

$$
\begin{aligned}
a_1 :=& 12 \left(\beta_1^* - \beta_2^* - 3\beta_3^* + \frac{2a(\beta_1^* - \beta_2^*)}{b-a} \right) \\
=& 12 \left(-3\beta_3^* + \frac{(a+b)(\beta_1^* - \beta_2^*)}{b-a} \right) \\
\leq& 12 \left(-3\beta_3^* + |a+b| \right) \leq -24\beta_3^* < 0,
\end{aligned}
$$

$$
\begin{aligned}
a_2 :=& 4 \left(\beta_1^* - \beta_2^* - 3\beta_3^* + \frac{2a(\beta_1^* - \beta_2^*)}{b-a} \right)^2 - 12((\beta_1^*)^2 - (\beta_3^*)^2) - \frac{12a((\beta_1^*)^2 - (\beta_2^*)^2)}{b-a} \\
& + \frac{16(\beta_1^* - \beta_2^*)^2}{(b-a)^2} - 12ab - 28c^2,
\end{aligned}
$$

$$
\begin{aligned}
a_3 := 8\Psi(\mathbf{s}^*) =& -8 \left((\beta_1^*)^2 - (\beta_3^*)^2 + \frac{a((\beta_1^*)^2 - (\beta_2^*)^2)}{b-a} + ab + c^2 \right) \\
& \left(\beta_1^* - \beta_2^* - 3\beta_3^* + \frac{2a(\beta_1^* - \beta_2^*)}{b-a} \right) \\
& - 16c^2 \left(-2\,\beta_1^* + \frac{((\beta_1^*)^2 - (\beta_2^*)^2 + (b-a)^2)(\beta_1^* - \beta_2^*)}{(b-a)^2} \right).
\end{aligned}
$$

This yields that $(s^*, y^*, \beta^*, \alpha^*)$ is an optimal solution of (4.2.1) if and only if for any $\lambda \in D$

$$\tilde{f}(\lambda) := 9\lambda^3 + a_1\lambda^2 + a_2\lambda + a_3 > 0.$$

In particular, $a_3 = \tilde{f}(0) \geq 0$, and so, $a_3 \geq 0$ is a necessary condition.

Conversely, if $a_3 > 0$, then by $\lim_{\lambda \to -\infty} \tilde{f}(\lambda) = -\infty$, we see that there exists $\lambda_0 < 0$ such that

$$\tilde{f}\left(\lambda_0\right) = 0.$$

Let λ_1 and λ_2 denote other two roots of the equation $\tilde{f}(\lambda) = 0$. Then

$$\lambda_0 + \lambda_1 + \lambda_2 = -\frac{1}{9}a_1.$$

Therefore, by $-\frac{1}{9}a_1 \geq \frac{8}{3}\beta_3^* > \beta_1^* + \beta_2^* - (b-a)$,

$$\lambda_1 + \lambda_2 > \beta_1^* + \beta_2^* - (b-a),$$

which implies that one of λ_1 and λ_2 is bigger than $(\beta_1^* + \beta_2^* - (b-a))/2$. On the other hand, since

$$\tilde{f}\left(0\right) > 0, \text{ and } \tilde{f}\left(\frac{\beta_1^* + \beta_2^* - (b-a)}{2}\right) \geq 0,$$

if $\tilde{f}\left(\frac{\beta_1^*+\beta_2^*-(b-a)}{2}\right) > 0$, then the number of roots in $((0, \beta_1^* + \beta_2^* + b - a)/2)$ of $\tilde{f}(\lambda) = 0$ must be even number. Therefore, or $\tilde{f}\left(\frac{\beta_1^*+\beta_2^*-(b-a)}{2}\right) = 0$, or both λ_1 and λ_2 are $((\beta_1^* + \beta_2^* + b - a)/2, \infty)$, and so, the mixed SDP-SOCP (4.2.1) has an exact relaxation.

The proof is completed.

■

Since the distance between two points is invariant under translation and rotation, we can state a general form of Theorem 4.3.1 in the following theorem.

Theorem 4.3.2 *Let $\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_3$ be three point in $\mathbb{R}^2$ and let $\boldsymbol{r}$ denote the pedal point of the point $\boldsymbol{a}_3$ to the line segment from $\boldsymbol{a}_1$ to $\boldsymbol{a}_2$. Set*

$$a = -\|\boldsymbol{a}_1 - \boldsymbol{r}\|, b = \|\boldsymbol{a}_2 - \boldsymbol{r}\|, c = -\|\boldsymbol{a}_3 - \boldsymbol{r}\|,$$

and

$$\beta_i^* = \|\mathbf{s}^* - \boldsymbol{a}_i\|, \quad i = 1, 2, 3.$$

Assume that $\boldsymbol{r}$ is between $\boldsymbol{a}_1$ and $\boldsymbol{a}_2$, and $-c \geq |a+b|$.

Let $(\hat{\mathbf{s}}, \hat{y}, \hat{\beta}, \hat{\alpha})$ be an optimal solution of (4.2.1) and assume that $\hat{s}_1 \in [a, b]$. If $\Psi(\mathbf{s}^*) > 0$, then $\hat{\mathbf{s}} = \mathbf{s}^*$.

Conversely, if the mixed SDP-SOCP (4.2.1) has an exact relaxation, then $\Psi(\mathbf{s}^*) \geq 0$.

Corollary 4.3.1 *For any three point $\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_3$ with $\|\boldsymbol{a}_1 - \boldsymbol{a}_3\| = \|\boldsymbol{a}_2 - \boldsymbol{a}_3\|$, set*

$$b = \|\boldsymbol{a}_1 - \boldsymbol{a}_2\|/2, \quad c = -\sqrt{\|\boldsymbol{a}_2 - \boldsymbol{a}_3\|^2 - b^2}.$$

Let $(\hat{\mathbf{s}}, \hat{y}, \hat{\beta}, \hat{\alpha})$ be an optimal solution of (4.2.1) and assume that $\hat{s}_1 \in [a, b]$. If

$$3\left((\beta_1^*)^2 + (\beta_2^*)^2 - 2(\beta_3^*)^2 - 2b^2 + 2c^2\right)\beta_3^* - \frac{c^2}{b^2}\left(((\beta_1^*)^2 - (\beta_2^*)^2) - 4b^2(\beta_1^* + \beta_2^*)\right) > 0, \tag{4.3.9}$$

then the mixed SDP-SOCP (4.2.1) has an exact relaxation.

Conversely, if the mixed SDP-SOCP (4.2.1) has an exact relaxation, then

$$3\left((\beta_1^*)^2 + (\beta_2^*)^2 - 2(\beta_3^*)^2 - 2b^2 + 2c^2\right)\beta_3^* - \frac{c^2}{b^2}\left(((\beta_1^*)^2 - (\beta_2^*)^2) - 4b^2(\beta_1^* + \beta_2^*)\right) \geq 0, \tag{4.3.10}$$

Proof. Since under the condition $a = -b$, a_3 can be simplified by

$$\begin{aligned} a_3 &= 8\Psi(\mathbf{s}^*) \\ &= 12\left((\beta_1^*)^2 + (\beta_2^*)^2 - 2(\beta_3^*)^2 - 2b^2 + 2c^2\right)\beta_3^* \\ &\quad - \frac{4c^2}{b^2}\left(((\beta_1^*)^2 - (\beta_2^*)^2) - 4b^2(\beta_1^* + \beta_2^*)\right), \end{aligned} \tag{4.3.11}$$

Corollary 4.3.1 is a consequence of Theorem 4.3.2.

■

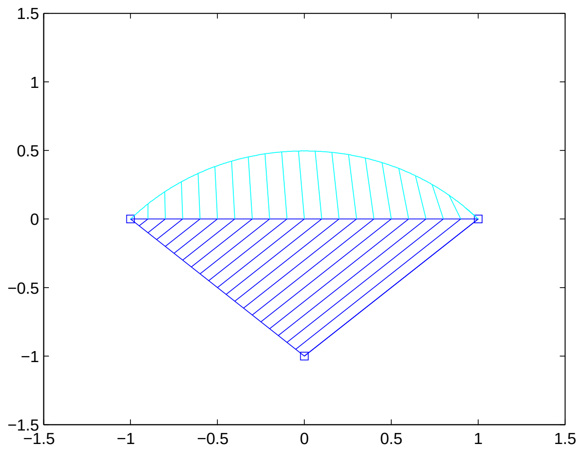

Figure 4.1: Exact localizable region

To illustrate the characterization theorem, we give a simple example here. Assume $a = -1$,$b = 1$,$c = -1$, by Theorem 4.3.1, we can find there is an extended region to the convex hull of the triangular array which has the exact localizable property (Figure 4.1). We call this region the localizable region of the model.

Chapter 5

Experiments for source localization problem

5.1 Introduction

In the last two chapters, we give the mixed SDP-SOCP relaxation model for source localization problem and show the exact solution region. In this chapter, some numerical results are presented. The chapter is organized as follows. In section 1, we give comparison between mixed SDP-SOCP relaxation model with other three relaxation models presented in chapter 2. The experiment for error correction algorithm is given in section 2. In section 3 and 4, some numerical results by real data and simulation are presented respectively using least square method, error correction method and bi-level method. In the last section, a example for mixed SDP-SOCP relaxation model for source localization combined with sensor network localization is given.

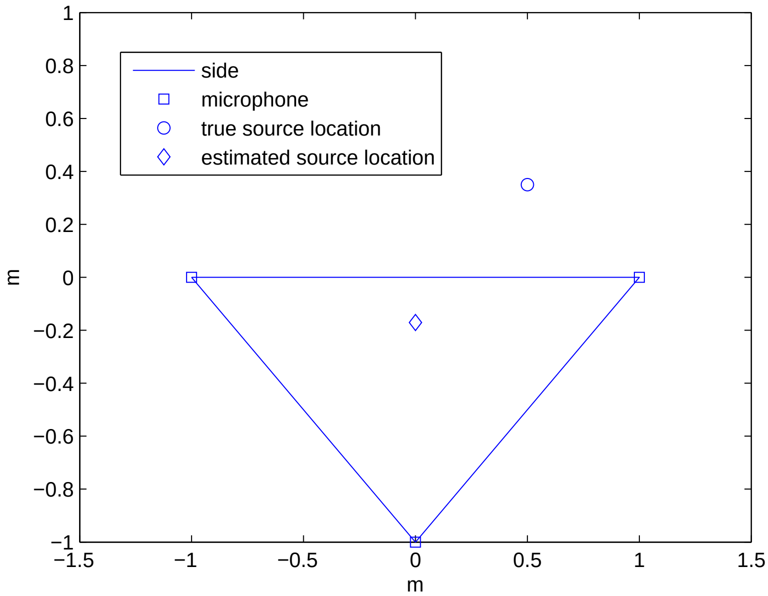

Figure 5.1: SDP relaxation model

5.2 Comparison between mixed SDP-SOCP relaxation model with other three models

In this section, some numerical results for comparing with four convex relaxation models in chapter 3 will be presented under the condition that there are no TODA errors. In solving those relaxation models, we use SDPT3 (Toh et al. [2006]) here rather than SDPA-M (Fujisawa et al. [2005]) used in the second chapter.

Assume there are three microphones located at the coordinates $\{(-1, 0), (0, -1), (1, 0)\}$ and the true source location is $(0.5, 0.35)$. The Figures 5.1, 5.2, 5.3 and 5.4 are using SDP relaxation model, SOCP relaxation model, YWL's model and mixed SDP-SOCP relaxation model respectively. From Table 5.1, we can see that the proposed mixed SDP-SOCP can indeed find the exact location subject to numerical errors, and is therefore more accurate than the other convex relaxation models.

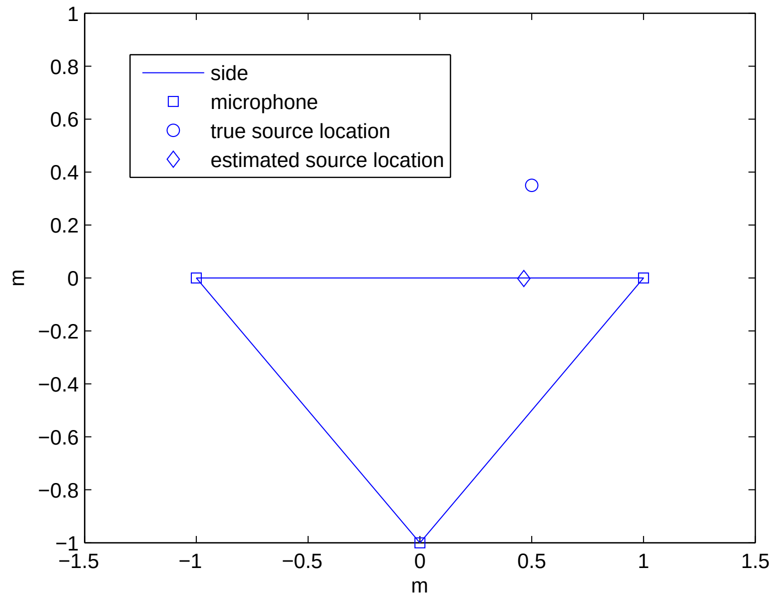

Figure 5.2: SOCP relaxation model

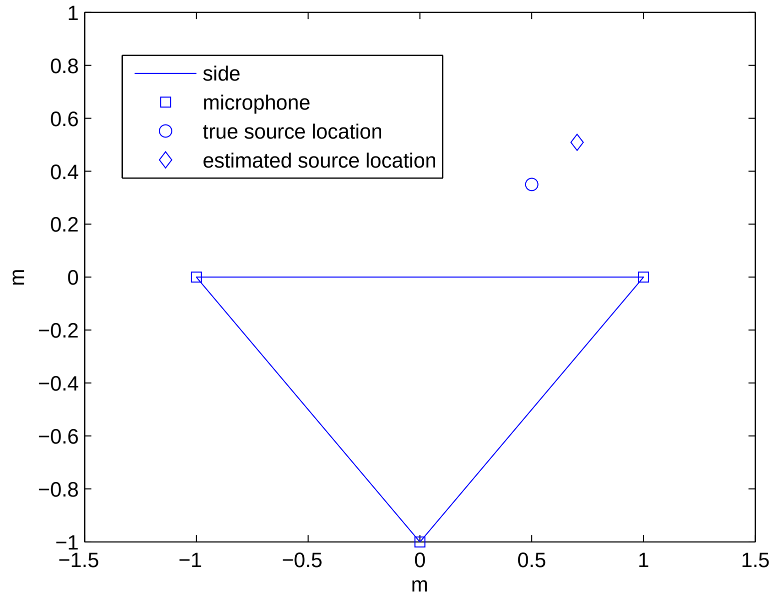

Figure 5.3: YWL's model

	SDP	SOCP	YWL	SDP-SOCP
ESL	$(0, -0.1707)$	$(0.4650, -0.0023)$	$(0.7033, 0.5092)$	$(0.5000, 0.3498)$
Error	0.7219	0.3540	0.2582	$1.6939e-04$

Table 5.1: Error comparison

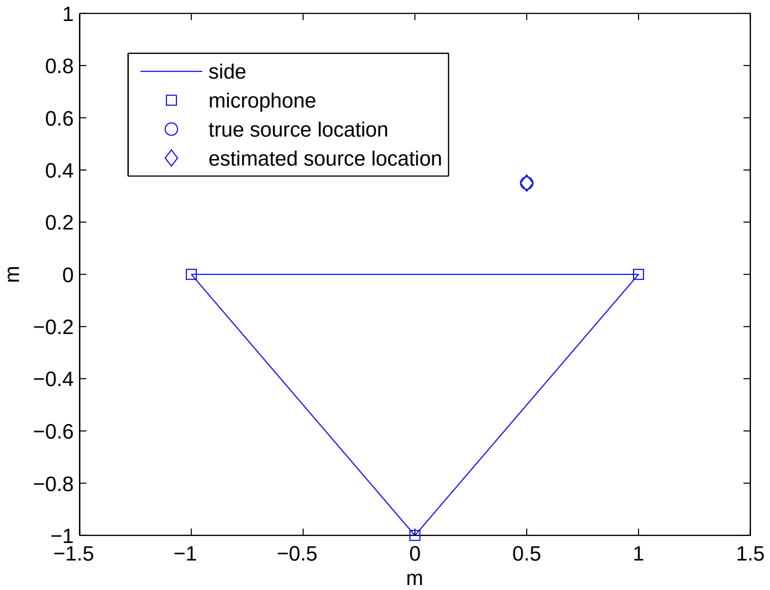

Figure 5.4: Mixed SDP-SOCP relaxation model

5.3 An error correction algorithm for source localization problem

Here, we also assume that there are three microphones located at the coordinates $\{(-1,0),(0,-1),(1,0)\}$, but the true source location is $(0.8, 0.5)$. Since the true source location is outside the region that our mixed SDP-SOCP relaxation model could correctly compute which we discussed in the previous chapter, we use the error correction algorithm in chapter 3. From Figure 5.5, the estimated source location solved by mixed SDP-SOCP relaxation model is $(0.7019, 0.2530)$ denoted by diamond point and the corresponding error is 0.2658. After using the error correction algorithm, the estimated source location become $(0.8000, 0.5000)$ which is the same as the true source location.

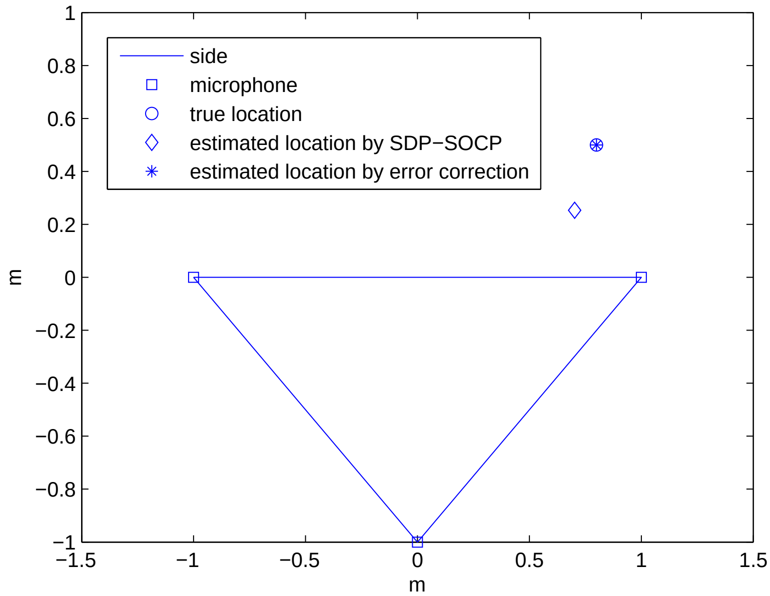

Figure 5.5: Error correction

5.4 Examples with real data

Equipment and setup

In collecting the experimental data, the microphone arrays used for the recording consist of 8 microphones, rectangular shape and L- shape in a room with 4 x 4.5 x 2.4m (L x W x H). The microphones are custom-built and are connected to OctaMic amplifier, which is then connected to ADI-648 by RME to Hammerfall DSP interface card in computer. All these equipments are from RME. The sound driver used is ASIO Hammerfall DSP. All the microphone has been calibrated before used. The speakers are connected using Delta 1010LT sound-card by M-Audio. Note that separate sound devices are used for recording and playback but their sampling frequency are set to be the same, which is 48000Hz. The speech signals used for the recordings are from NOIZEUS database, which consists of 30 male and female clean speeches. The noise used is babble noise from NOISEX-92 database. All signals are up-sampled to 48000Hz for playback. The recording setups are as shown in Figures 5.6 and 5.7. For each setup, Spkr 1 and Spkr 2 are

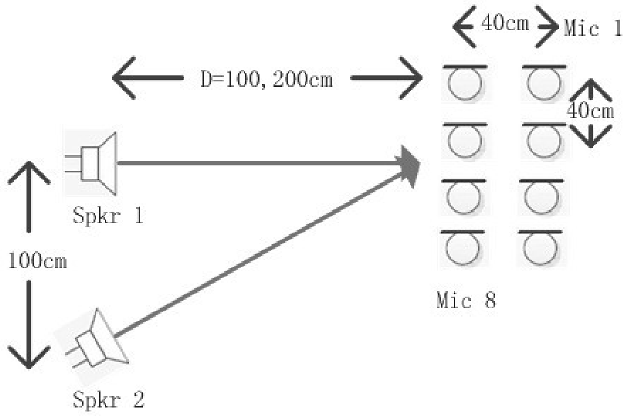

Figure 5.6: Rectangular shape array

positioned at two distance, d=100 and 200cm.

Mixed SDP-SOCP relaxation model with least square method

We continue to study the use of the mixed SDP-SOCP model for localization with real data. The SDPT3 (Toh et al. [2006]) will be employed to solve the formulated model (3.2.25) for $\hat{\mathbf{s}}$. By the characterization theorem, this relaxation model will not be exact when the source is out the localizable region even the time delay information is very accurate. Therefore, the location estimate can be further refined by using $\hat{\mathbf{s}}$ as a starting point for a further optimization using the least square iteration.

First we use this real data to compute the TDOA, and then solve the formulated mixed SDP-SOCP model (3.2.25) to get $\hat{\mathbf{s}}$. Figures 5.8, 5.9,5.10 and 5.11 are for rectangular shape microphone array, and Figures 5.12, 5.13,5.14 and 5.15 are for L shape microphone array, where the circle point is the true source location and the diamond point is the estimated source location computed from least square iteration with $\hat{\mathbf{s}}$ as a starting point. Clearly, the results look promising and the proposed method is very efficient computationally.

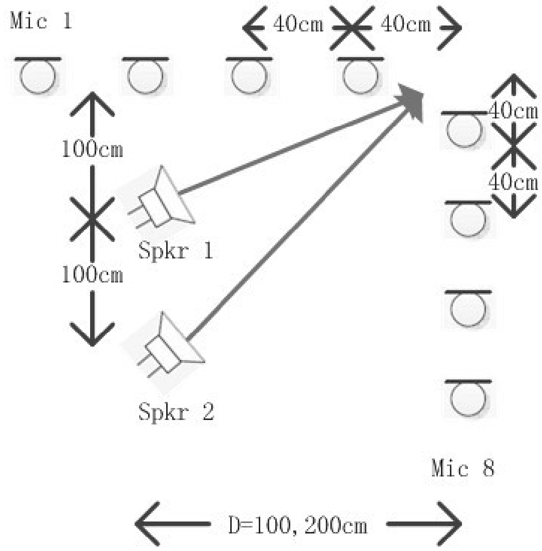

Figure 5.7: L shape array

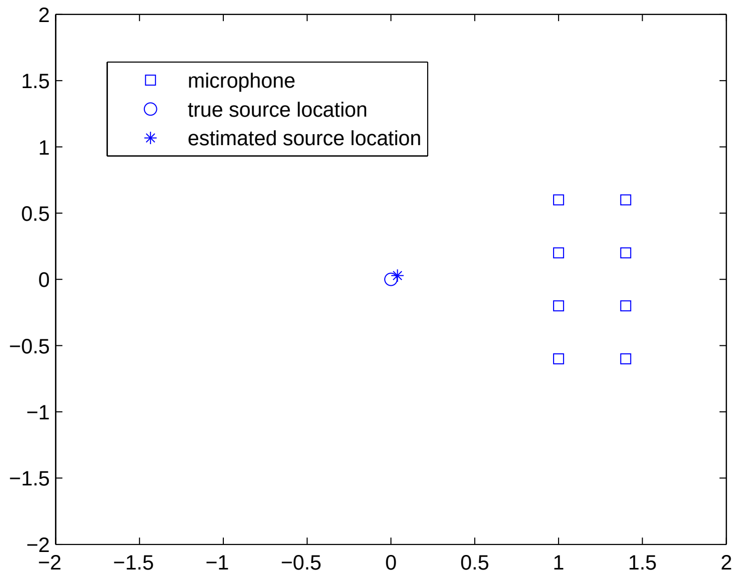

Figure 5.8: Speaker 1, D=100cm, SDP-SOCP-LS method

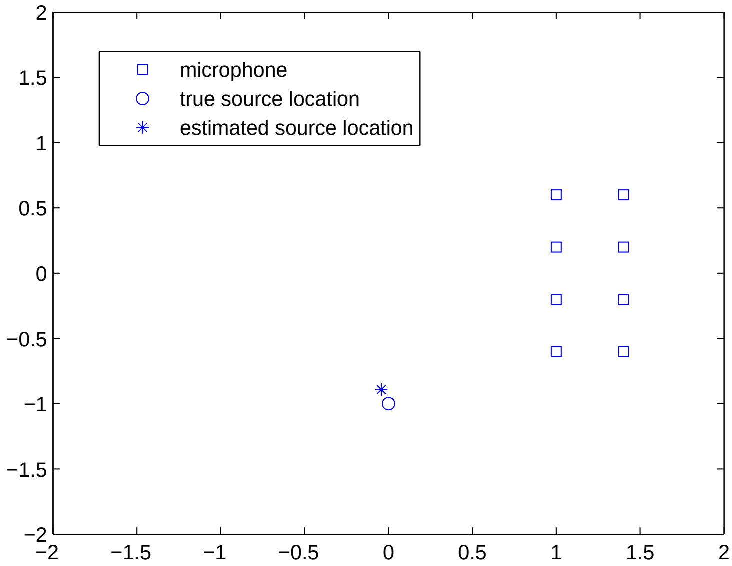

Figure 5.9: Speaker 2, D=100cm, SDP-SOCP-LS method

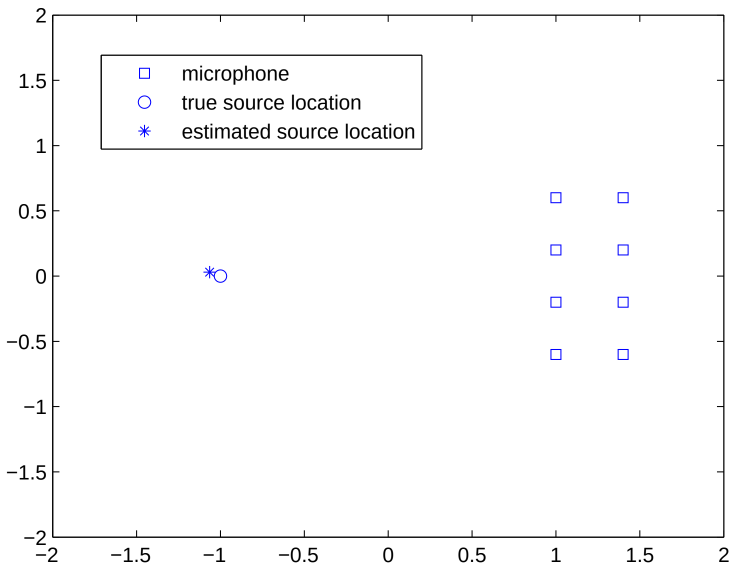

Figure 5.10: Speaker 1, D=200cm, SDP-SOCP-LS method

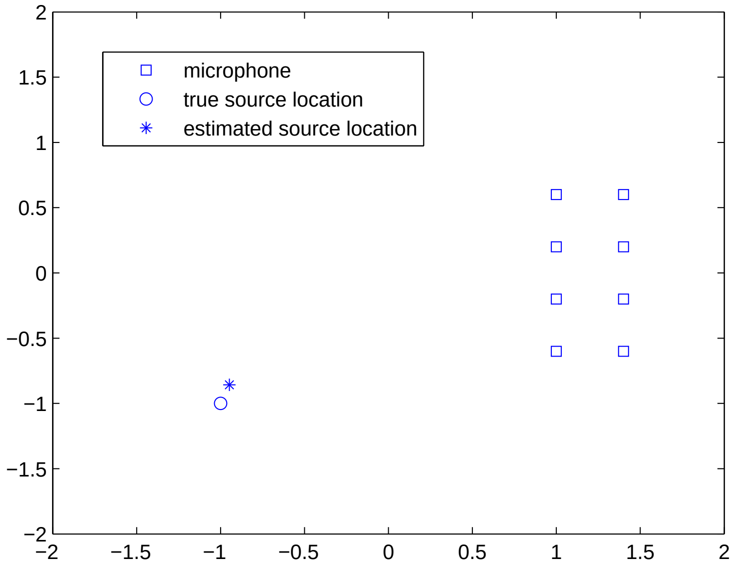

Figure 5.11: Speaker 2, D=200cm, SDP-SOCP-LS method

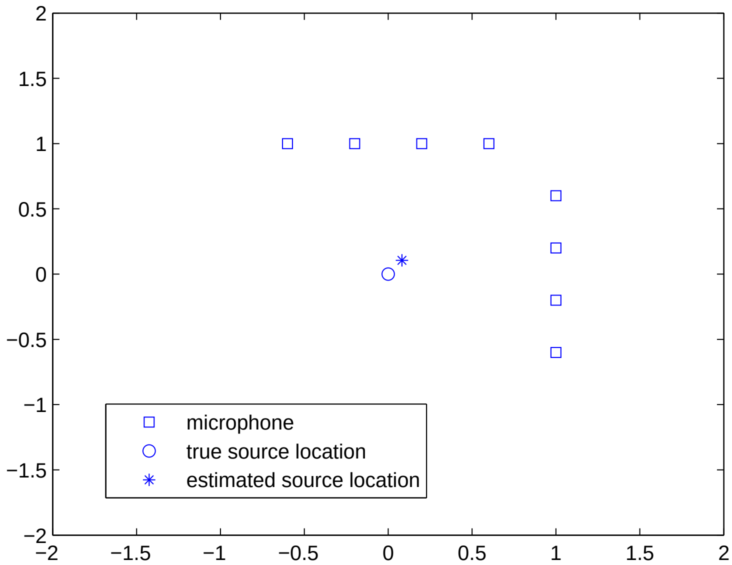

Figure 5.12: Speaker 1, D=100cm, SDP-SOCP-LS method

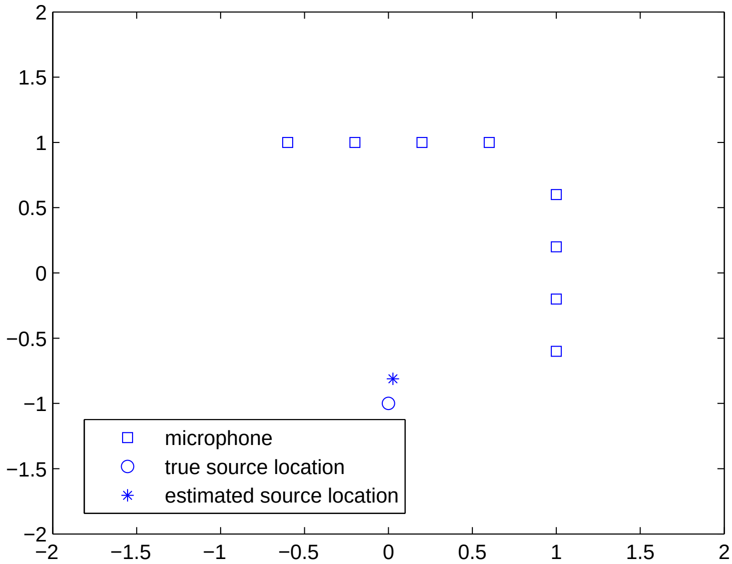

Figure 5.13: Speaker 2, D=100cm, SDP-SOCP-LS method

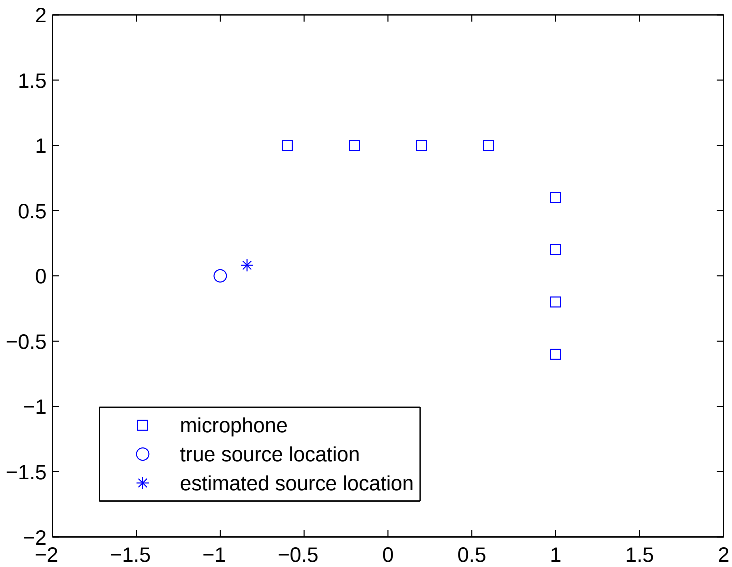

Figure 5.14: Speaker 1, D=200cm, SDP-SOCP-LS method

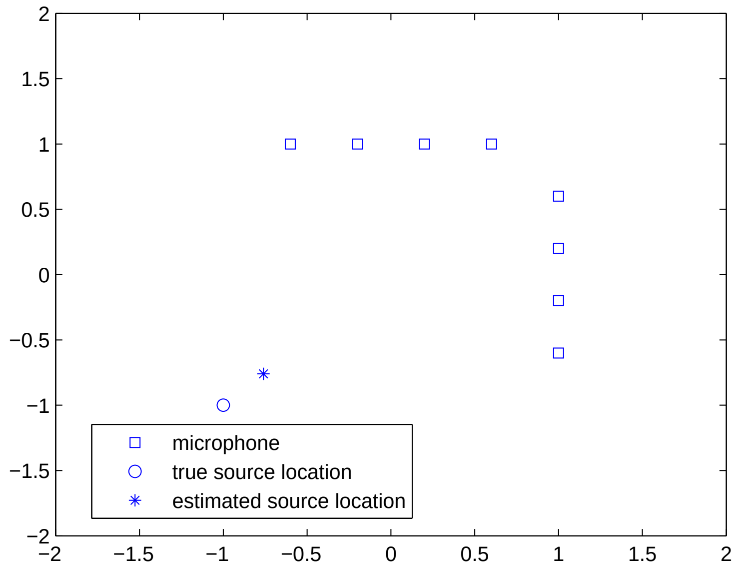

Figure 5.15: Speaker 2, D=200cm, SDP-SOCP-LS method

Mixed SDP-SOCP relaxation model with error correction method

Figures 5.16, 5.17,5.18 and 5.19 are for rectangular shape microphone array, and Figures 5.20, 5.21,5.22 and 5.23 are for L shape microphone array, where the circle point is the true source location and the star point is the estimated source location computed from error correction method.

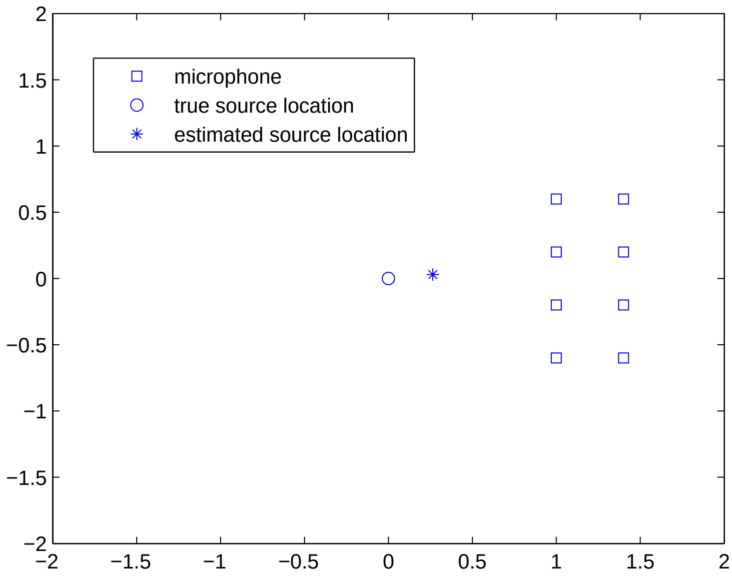

Figure 5.16: Speaker 1, D=100cm, SDP-SOCP-EC method

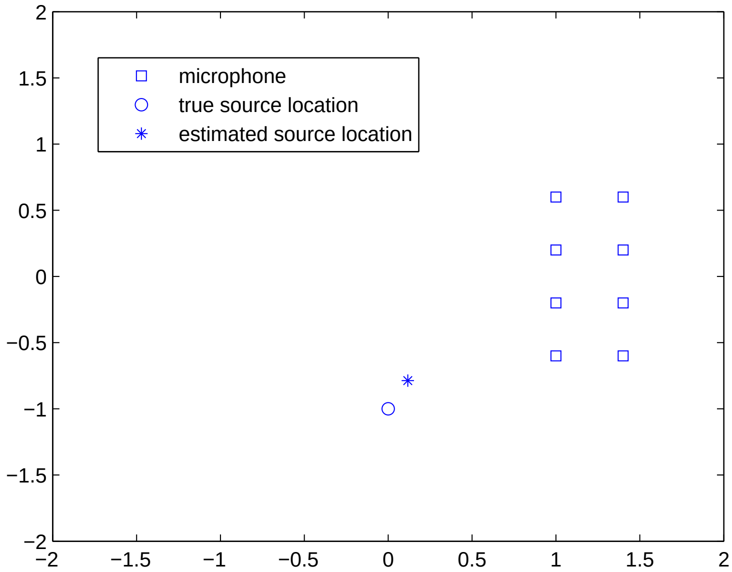

Figure 5.17: Speaker 2, D=100cm, SDP-SOCP-EC method

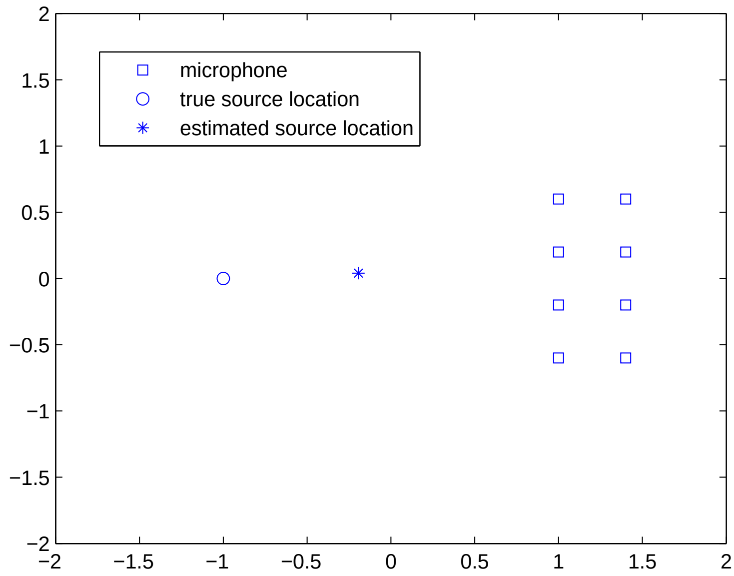

Figure 5.18: Speaker 1, D=200cm, SDP-SOCP-EC method

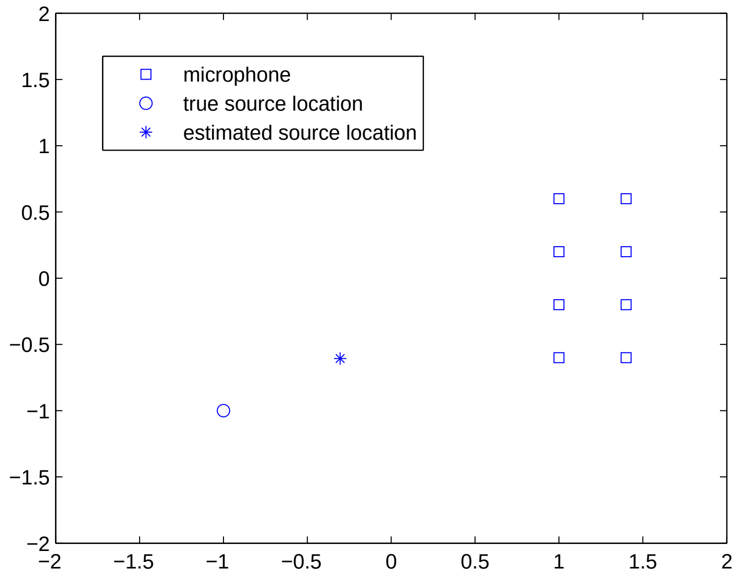

Figure 5.19: Speaker 2, D=200cm, SDP-SOCP-EC method

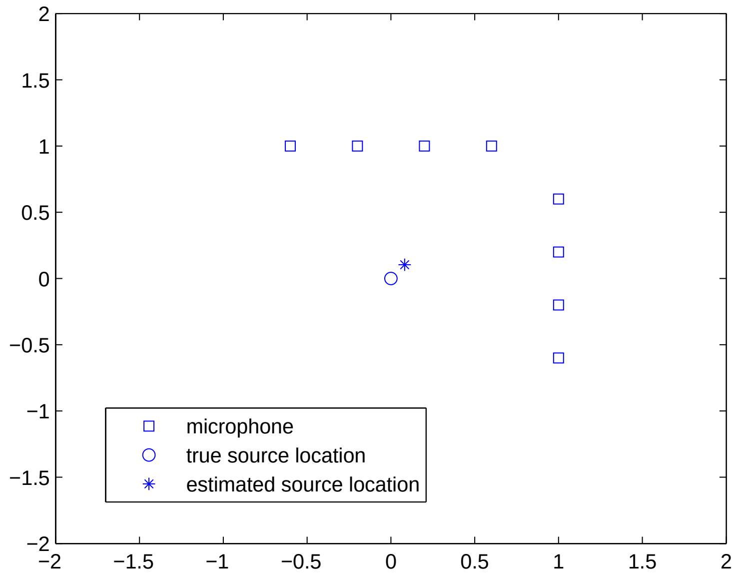

Figure 5.20: Speaker 1, D=100cm, SDP-SOCP-EC method

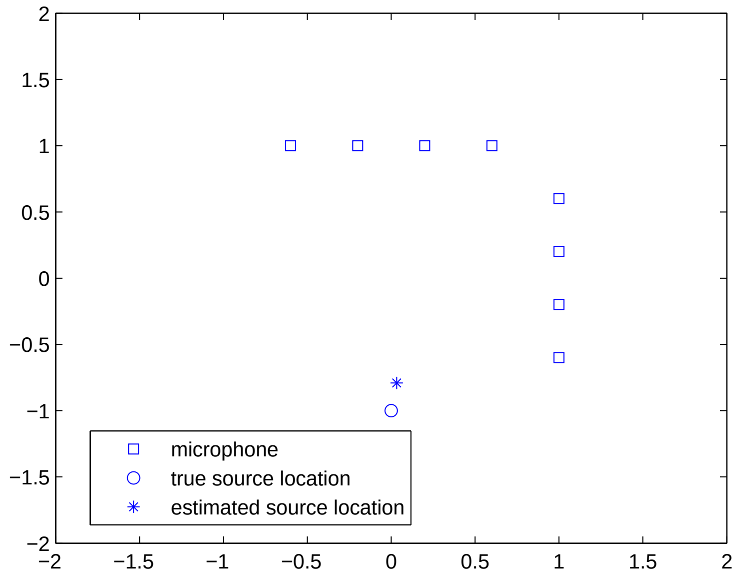

Figure 5.21: Speaker 2, D=100cm, SDP-SOCP-EC method

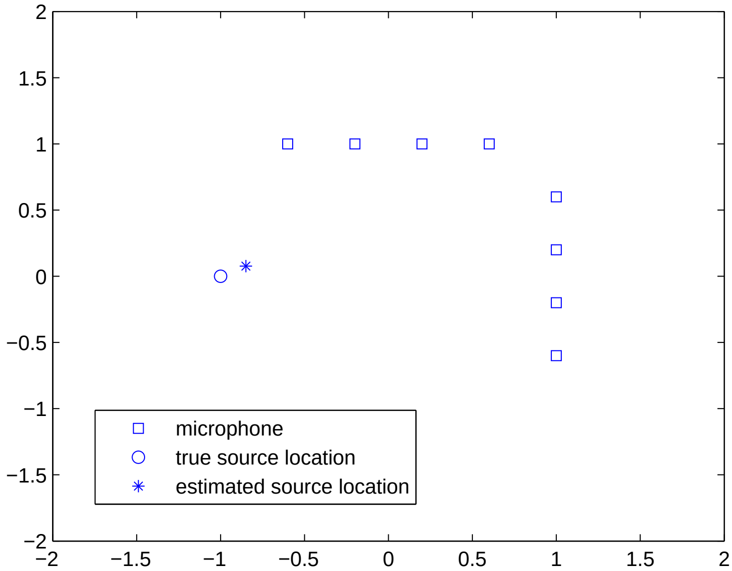

Figure 5.22: Speaker 1, D=200cm, SDP-SOCP-EC method

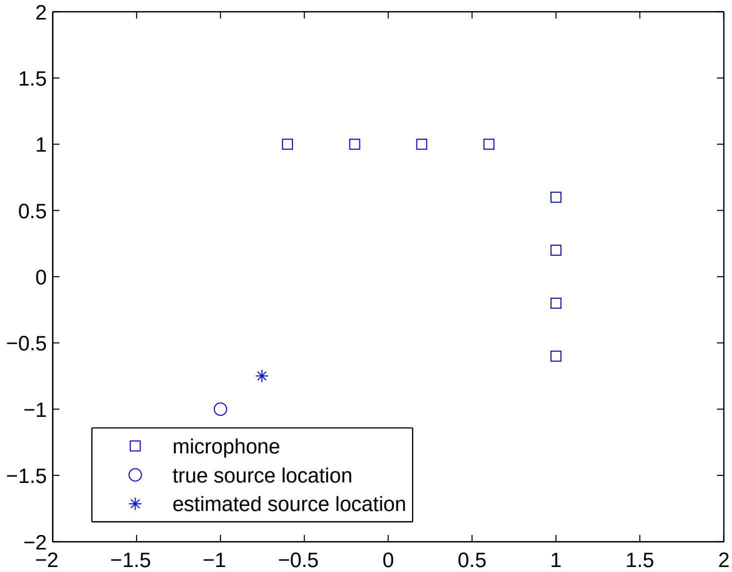

Figure 5.23: Speaker 2, D=200cm, SDP-SOCP-EC method

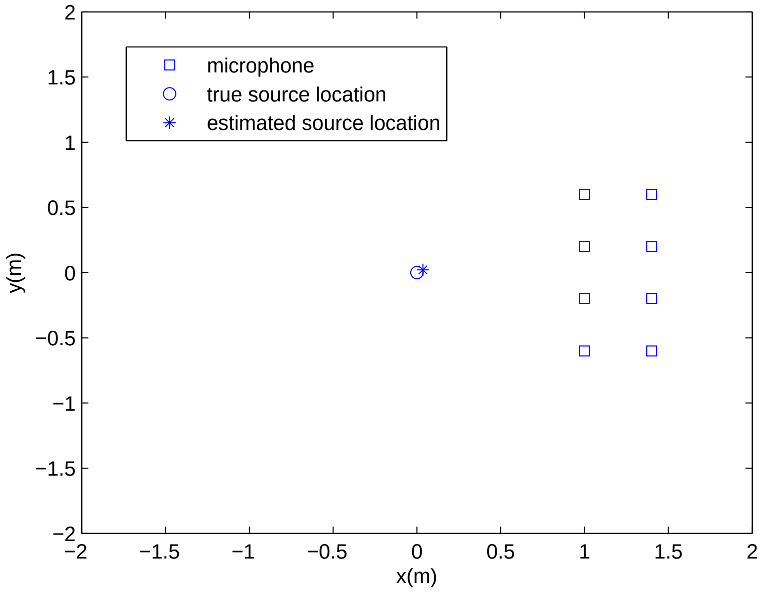

Figure 5.24: Speaker 1, D=100cm, Bi-level method

Bi-level method

Figures 5.24, 5.25,5.26 and 5.27 are for rectangular shape microphone array, and Figures 5.28, 5.29,5.30 and 5.31 are for L shape microphone array, where the circle point is the true source location and the star point is the estimated source location computed from bi-level method.

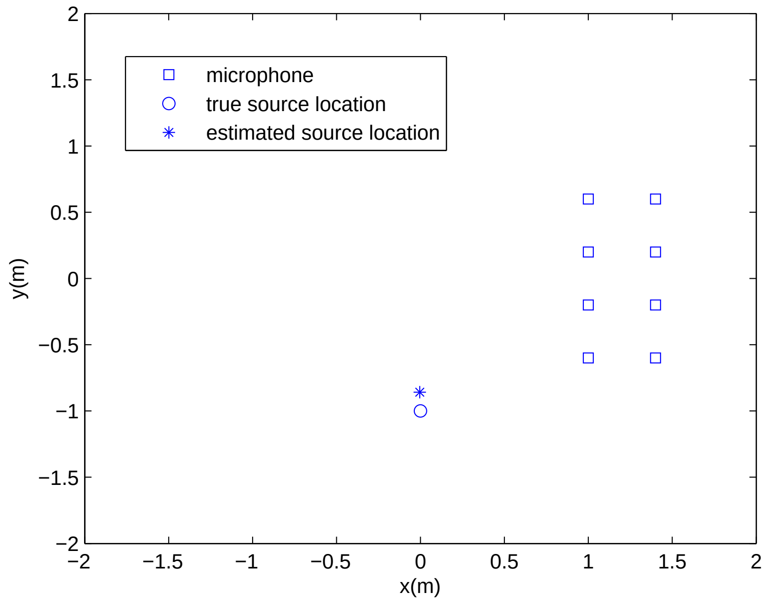

Figure 5.25: Speaker 2, D=100cm, Bi-level method

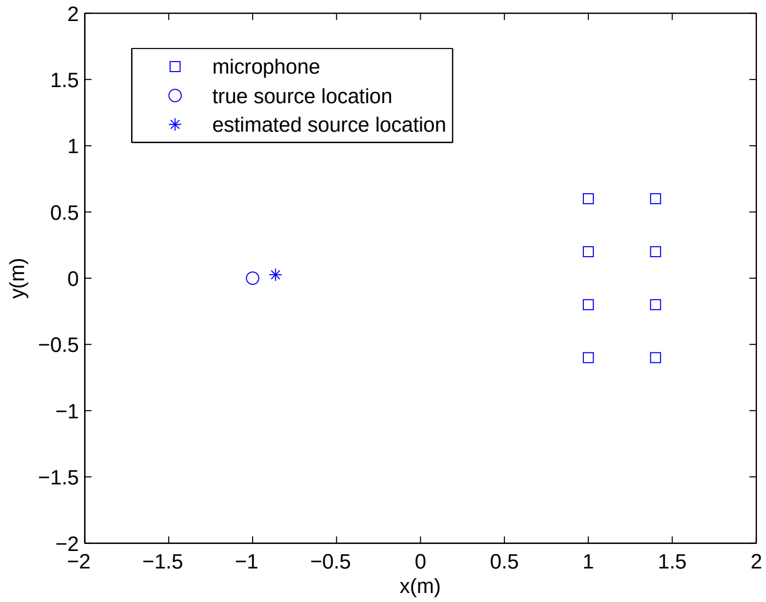

Figure 5.26: Speaker 1, D=200cm, Bi-level method

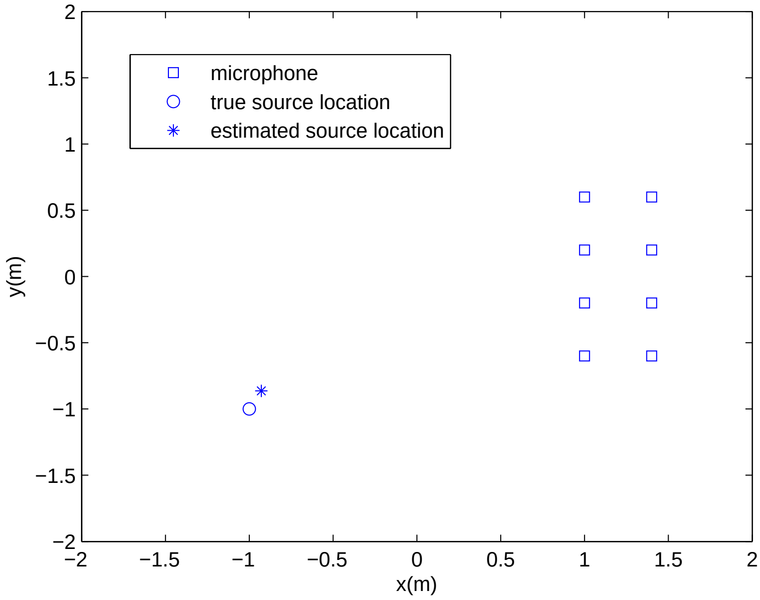

Figure 5.27: Speaker 2, D=200cm, Bi-level method

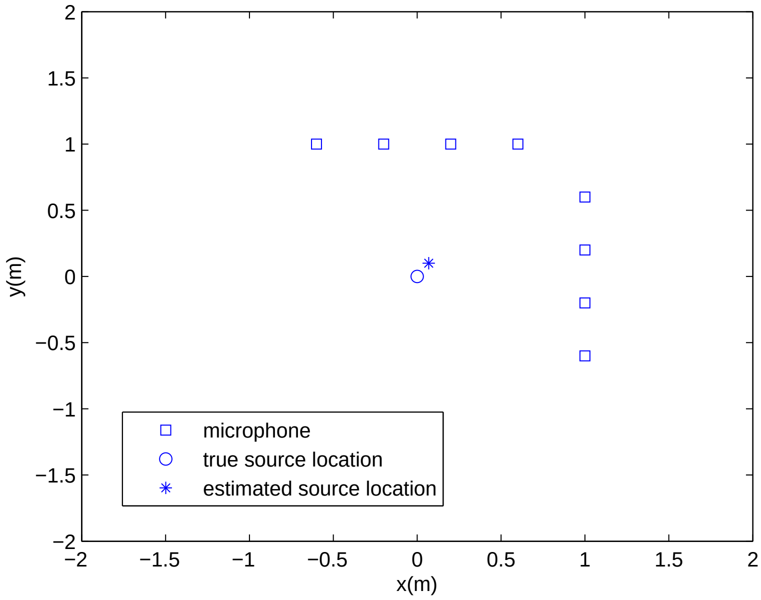

Figure 5.28: Speaker 1, D=100cm, Bi-level method

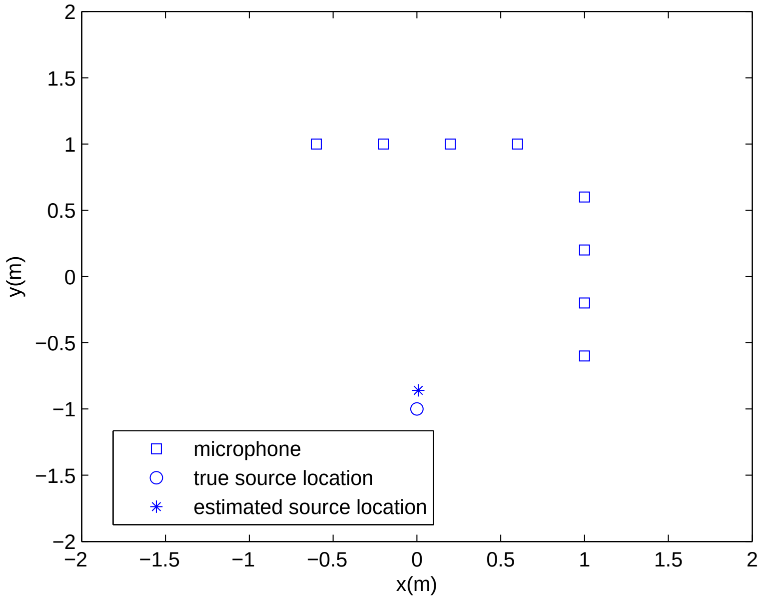

Figure 5.29: Speaker 2, D=100cm, Bi-level method

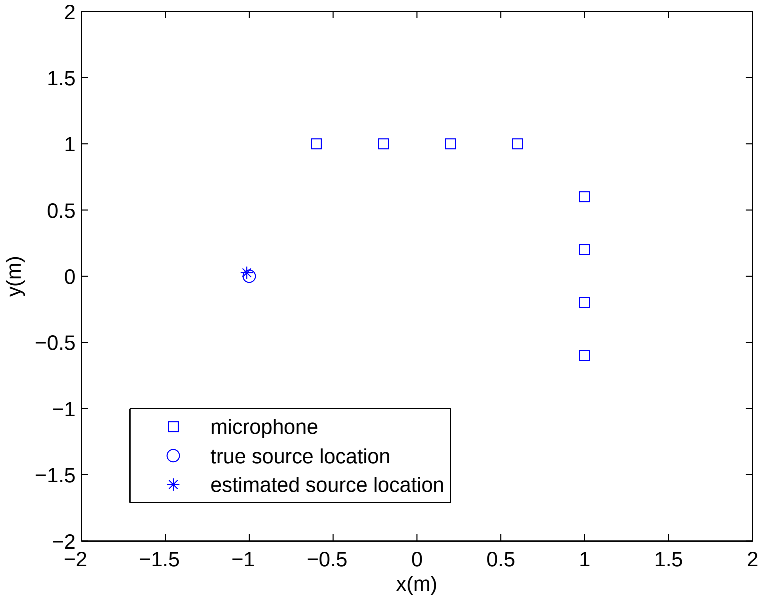

Figure 5.30: Speaker 1, D=200cm, Bi-level method

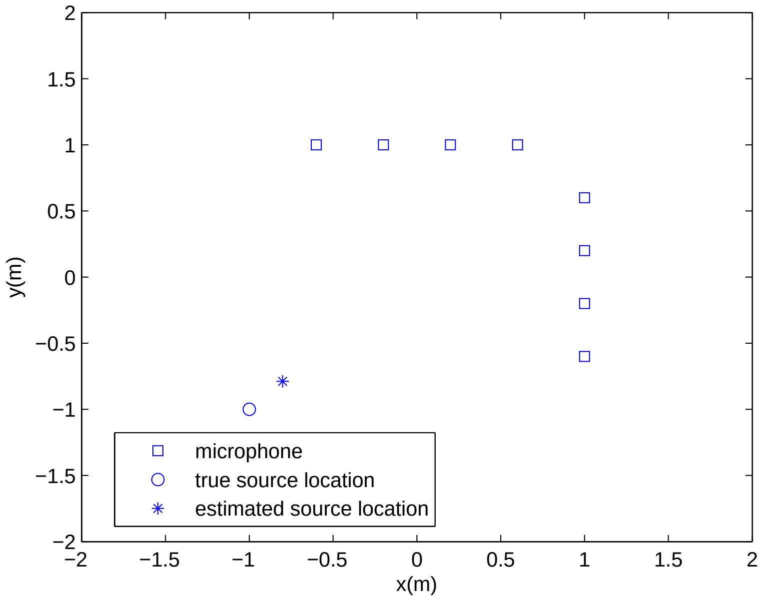

Figure 5.31: Speaker 2, D=200cm, Bi-level method

	SDP-SOCP-LS	SDP-SOCP-EC	Bi-level
speaker 1, D=100cm	0.0483	0.2664	0.0409
speaker 2, D=100cm	0.1167	0.2429	0.1424
speaker 1, D=200cm	0.0712	0.8071	0.1391
speaker 2, D=200cm	0.1522	0.7990	0.1540

Table 5.2: Error comparison with real data for rectangular shape microphone array

Tables 5.2 and 5.3 are the error comparisons for rectangular and L shape microphone array.

	SDP-SOCP-LS	SDP-SOCP-EC	Bi-level
speaker 1, D=100cm	0.1333	0.1320	0.1210
speaker 2, D=100cm	0.1906	0.2113	0.1403
speaker 1, D=200cm	0.1793	0.1686	0.0291
speaker 2, D=200cm	0.3393	0.3522	0.2908

Table 5.3: Error comparison with real data for L shape microphone array

5.5 Examples with simulation

In this section, The matlab code written by Eric A. Lehmann for the purpose of simulating reverberant audio data in small-room acoustics is used. It provide an implementation of the image-source method (ISM) described in (Lehmann and Johansson [2008]).

The image-source model (ISM) is a well-known technique that can be used in order to generate a synthetic room impulse response (RIR), i.e., a transfer function between a sound source and an acoustic sensor, in a given environment. Once such a RIR is available, a sample of audio data can be obtained by convolving the RIR with a given source signal. This provides a realistic sample of the sound signal that would effectively be recorded at the sensor in the considered environment.

This approach is of considerable importance as it provides a quick and easy way to generate a number of RIRs with varying environmental characteristics, such as different reverberation times, for instance. Consequently, the ISM technique has been used intensively in many application domains of room acoustics and signal processing.

Here we assume the room is 5 x 5 x 2.5m (L x W x H).

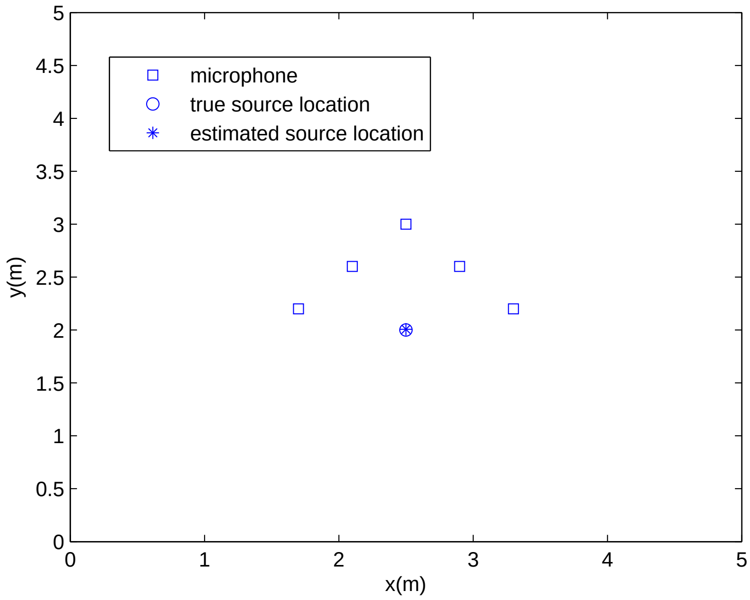

Figure 5.32: Example 1 with simulation, SDP-SOCP-EC method

Mixed SDP-SOCP relaxation model with error correction method

In this section, we use mixed SDP-SOCP relaxation model with error correction method to solve.

First, there are five microphones located at the coordinates $\{(1.7, 2.2), (2.1, 2.6), (2.5, 3), (2.9, 2.6), (3.3, 2.2)\}$ and the true source location is $(2.5, 2)$. From Fig 5.32, we can see that the estimated source location denoted by star point is almost at the same position as the circle point which is the true source location. The corresponding error is 0.0064.

In the next example, we move the source to the position $(2.5, 1)$ and microphones' positions are the same as previous example, see 5.33. Adding microphones to the right hand side or the left hand side regularly, See Figures 5.34, 5.35, 5.36 and 5.37. The corresponding errors are 0.1753, 0.1207, 0.1119, 0.0557 and 0.0209. We can see the error becomes small when adding microphone.

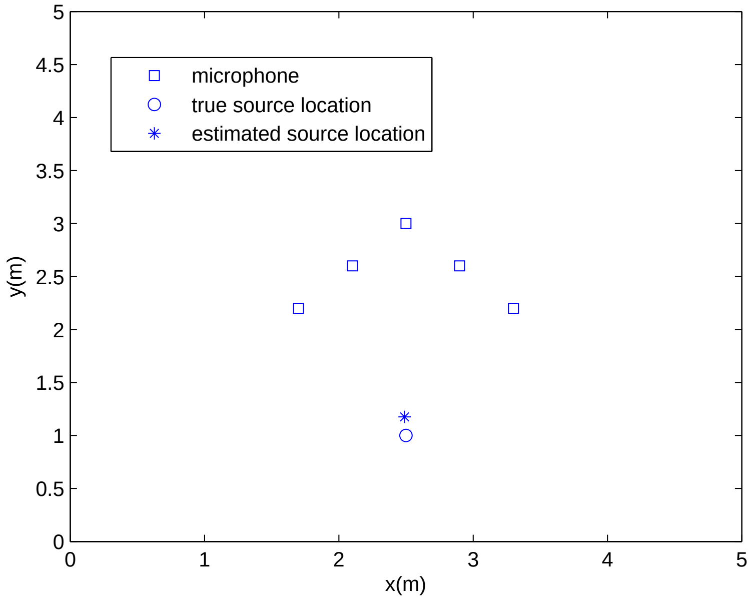

Figure 5.33: Example 2 with simulation, SDP-SOCP-EC method

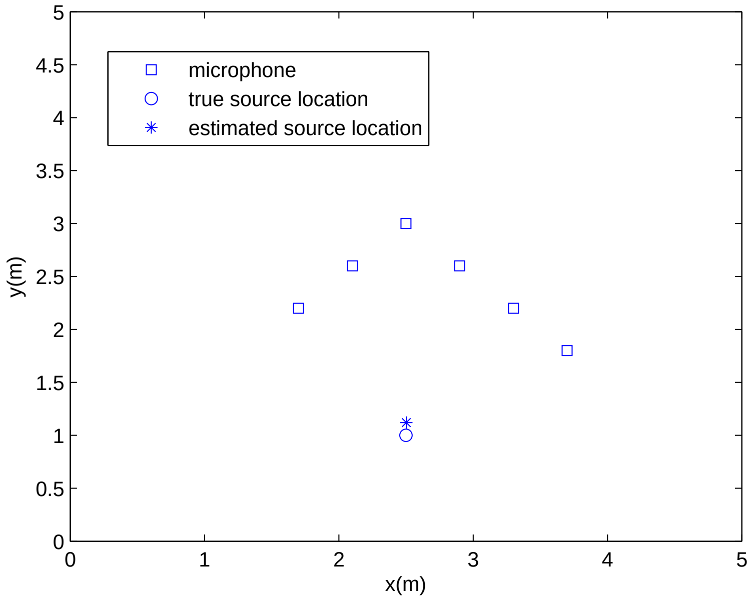

Figure 5.34: Example 2 with simulation, adding microphone (3.7,1.8), SDP-SOCP-EC method

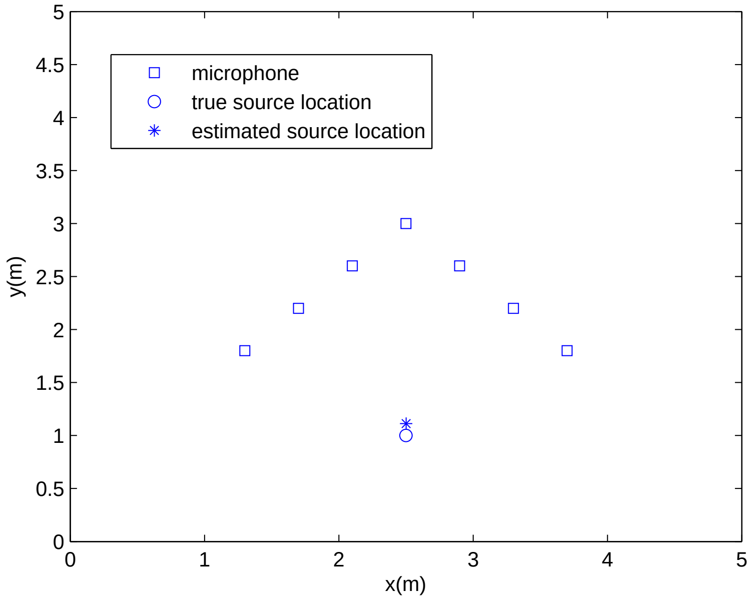

Figure 5.35: Example 2 with simulation, adding microphone (1.3,1.8), SDP-SOCP-EC method

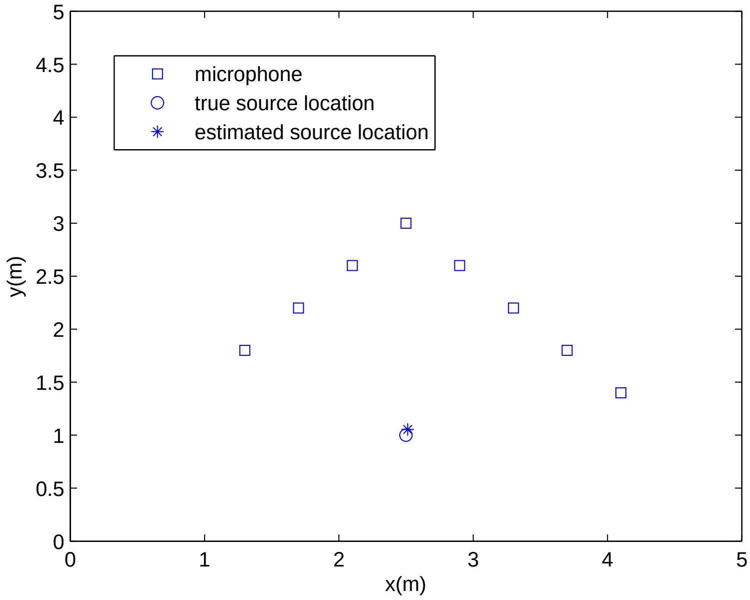

Figure 5.36: Example 2 with simulation, adding microphone (4.1,1.4), SDP-SOCP-EC method

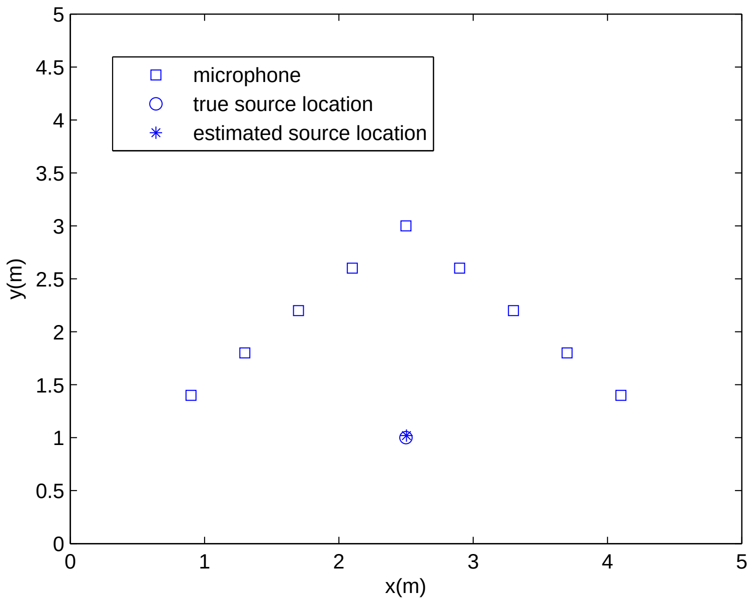

Figure 5.37: Example 2 with simulation, adding microphone (0.9,1.4), SDP-SOCP-EC method

Next, we use another shape of microphone. Assume there are six microphones located at the coordinates $\{(2.1, 3), (2.1, 2.5), (2.5, 3), (2.5, 2.5), (2.9, 3), (2.9, 2.5)\}$ and the true source location is $(2.5, 2)$. See Fig 5.38, where the circle point is the true source location and the star point is the estimated source location. The corresponding error is 0.0694.

In the next example, we move the source to the position $(2.5, 1.6)$ and microphones' positions are the same as previous example, see 5.39. Adding microphones to the right hand side or the left hand side regularly, See Figures 5.40 and 5.41. The corresponding errors are 0.3837, 0.2839 and 0.1453. We can see the error becomes small when adding microphone.

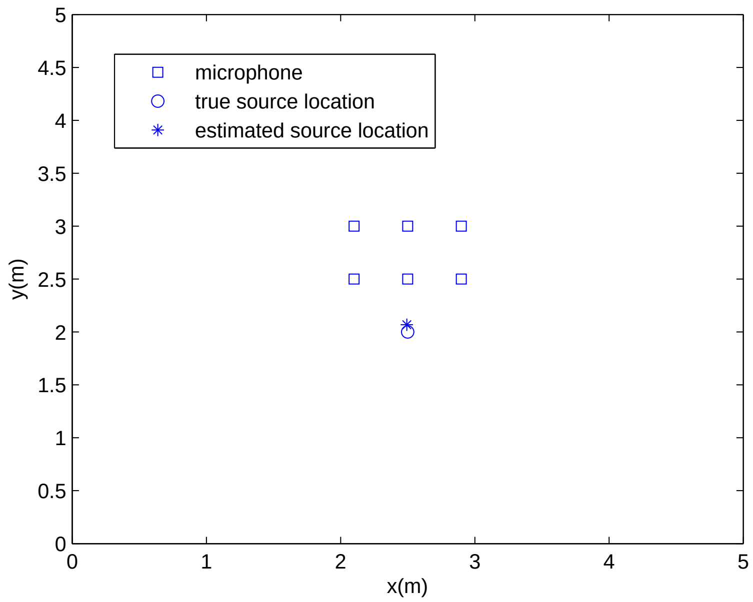

Figure 5.38: Example 3 with simulation, SDP-SOCP-EC method

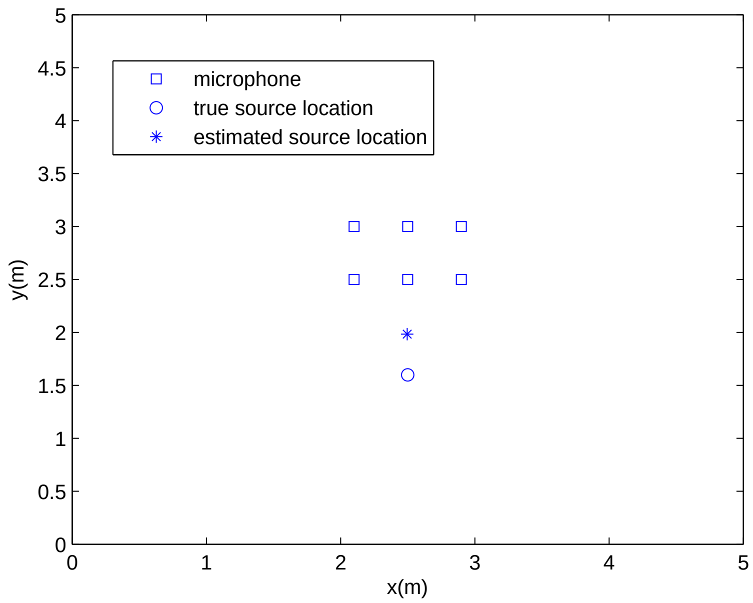

Figure 5.39: Example 4 with simulation, adding microphone (2.9,2.5), SDP-SOCP-EC method

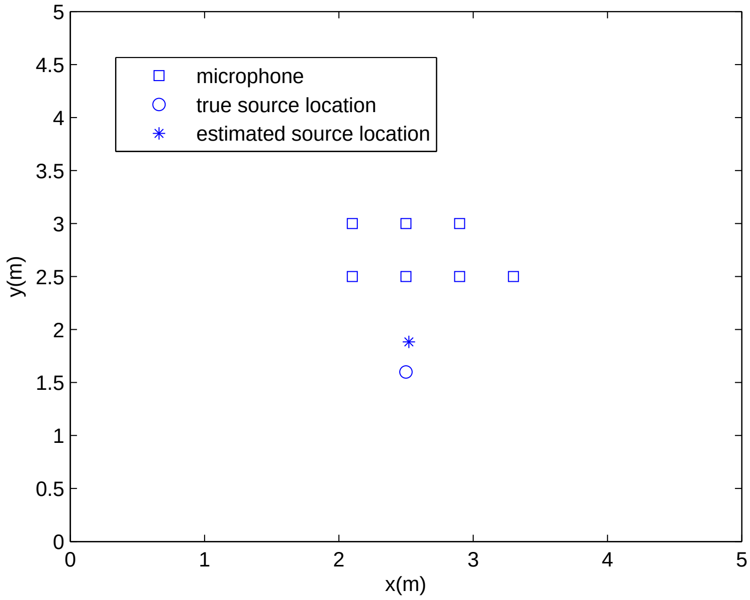

Figure 5.40: Example 4 with simulation, adding microphone (3.3,2.5), SDP-SOCP-EC method

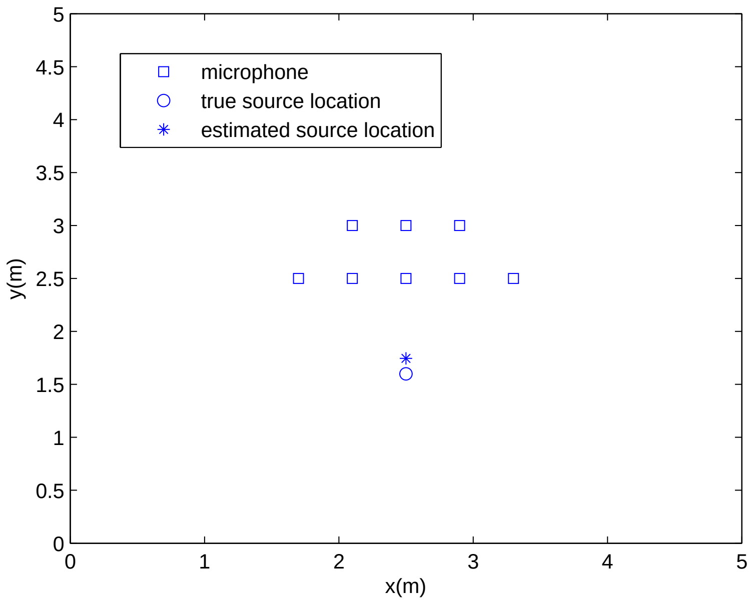

Figure 5.41: Example 4 with simulation, adding microphone (1.7,2.5), SDP-SOCP-EC method

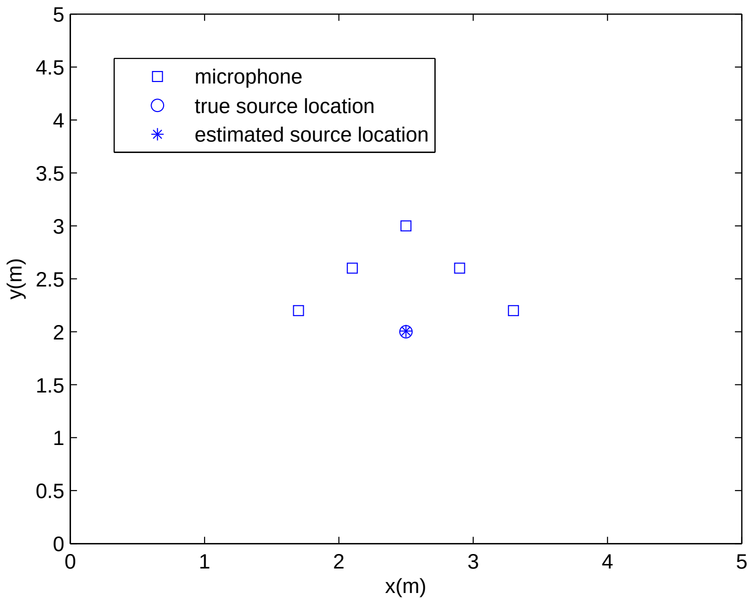

Figure 5.42: Example 5 with simulation, Bi-level method

Bi-level method

In this section, we use bi-level method to solve. The same as previous section, first assume there are five microphones located at the coordinates $\{(1.7, 2.2), (2.1, 2.6), (2.5, 3), (2.9, 2.6), (3.3, 2.2)\}$ and the true source location is $(2.5, 2)$. See Fig 5.42, the corresponding error is 0.0059.

In the next example, we move the source to the position $(2.5, 1)$ and microphones' positions are the same as previous example, see 5.43. Adding microphones to the right hand side or the left hand side regularly, See Figures 5.44, 5.45, 5.46 and 5.47. The corresponding errors are 0.7259, 0.0372, 0.0372, 0.0336 and 0.0336.

Next, we use another shape of microphone. Assume there are six microphones located at the coordinates $\{(2.1, 3), (2.1, 2.5), (2.5, 3), (2.5, 2.5), (2.9, 3), (2.9, 2.5)\}$ and the true source location is $(2.5, 2)$. See Fig 5.48, where the circle point is the true source location and the star point is the estimated source location. The corresponding error is 0.0072.

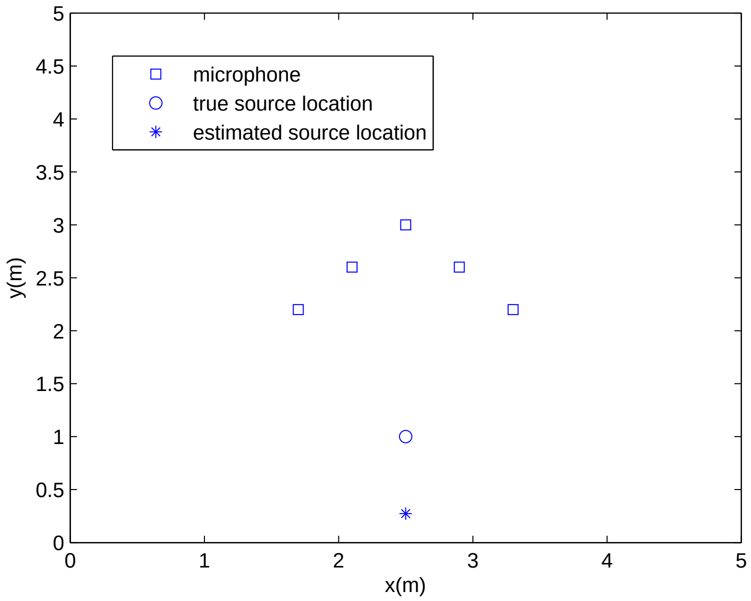

Figure 5.43: Example 6 with simulation, Bi-level method

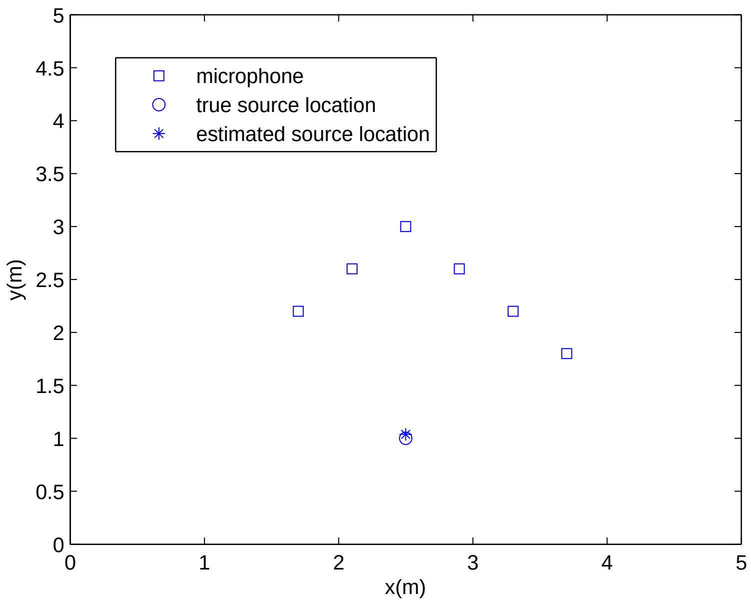

Figure 5.44: Example 6 with simulation, adding microphone (3.7,1.8), Bi-level method

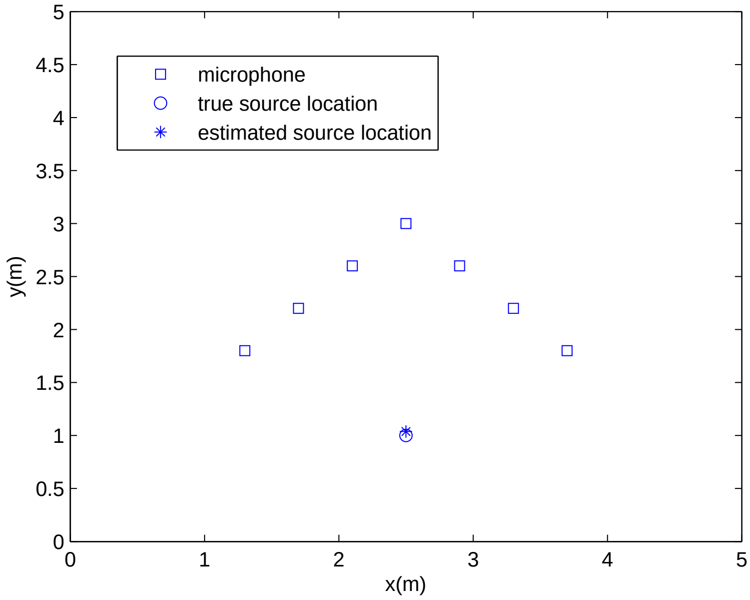

Figure 5.45: Example 6 with simulation, adding microphone (1.3,1.8), Bi-level method

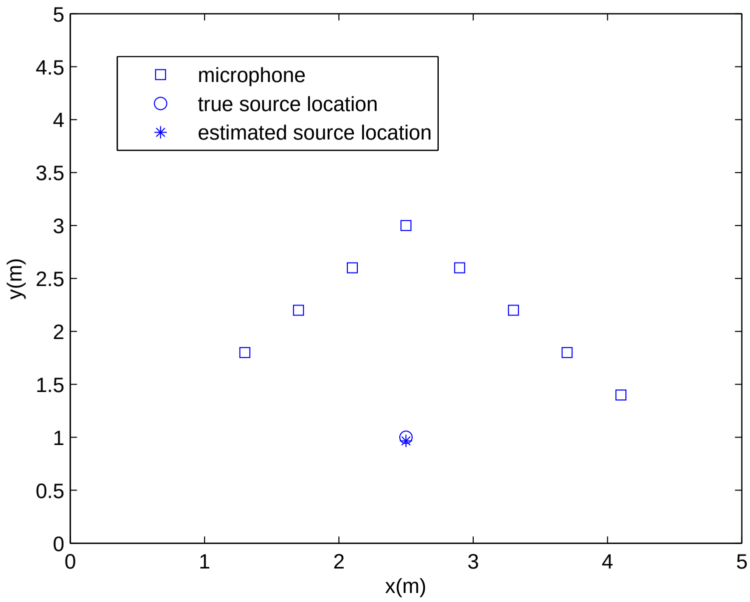

Figure 5.46: Example 6 with simulation, adding microphone (4.1,1.4), Bi-level method

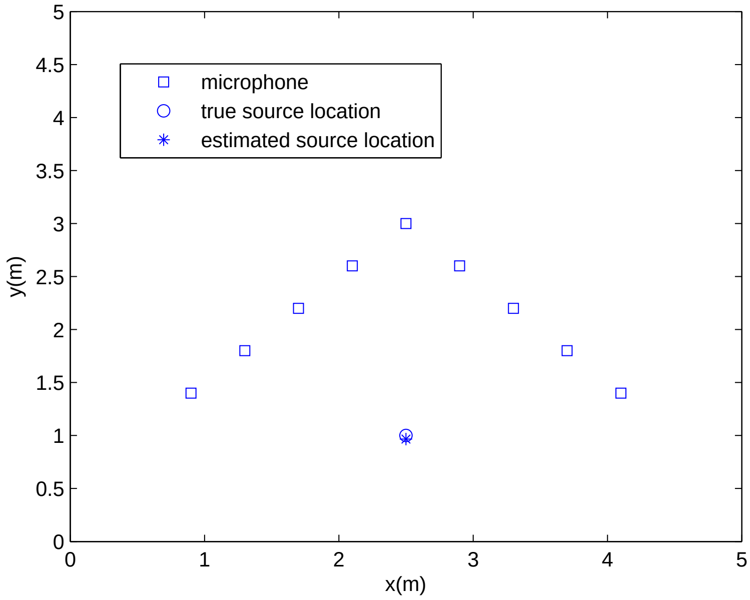

Figure 5.47: Example 6 with simulation, adding microphone (0.9,1.4), Bi-level method

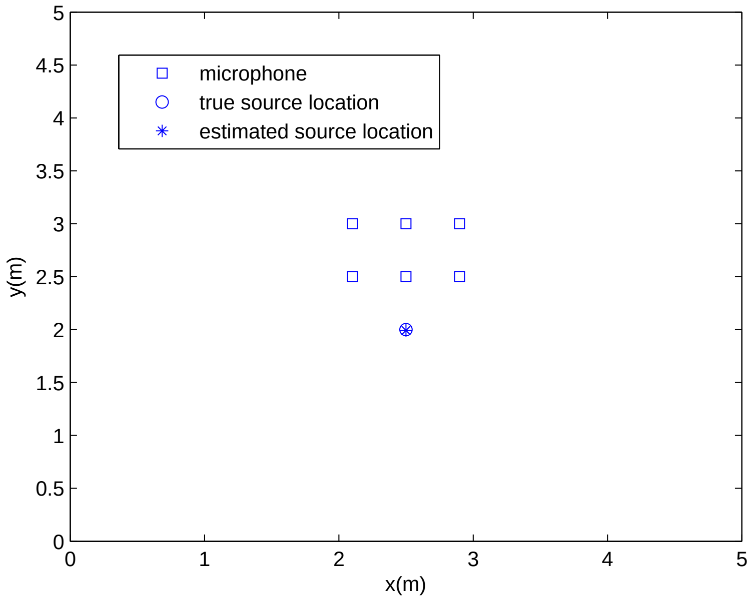

Figure 5.48: Example 7 with simulation, Bi-level method

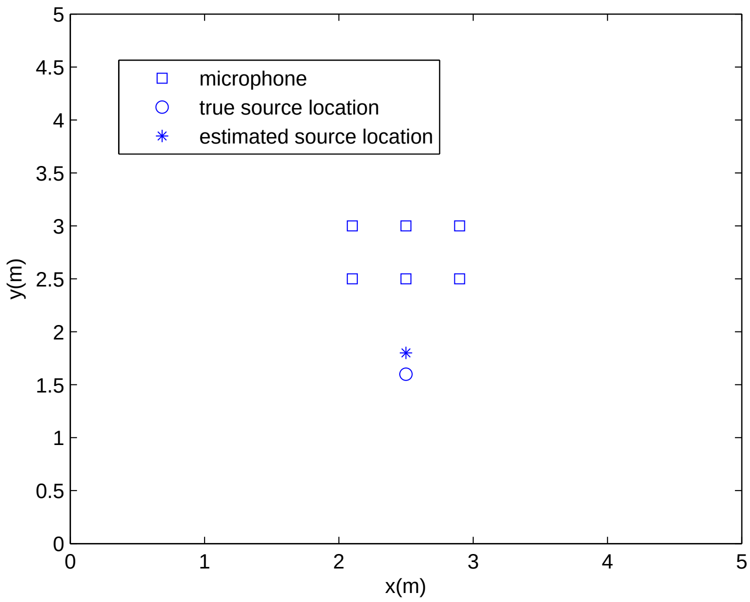

Figure 5.49: Example 8 with simulation, Bi-level method

In the next example, we move the source to the position $(2.5, 1.6)$ and microphones' positions are the same as previous example, see 5.49. Adding microphones to the right hand side or the left hand side regularly, See Figures 5.50 and 5.51. The corresponding errors are 0.2007, 0.0377 and 0.0377.

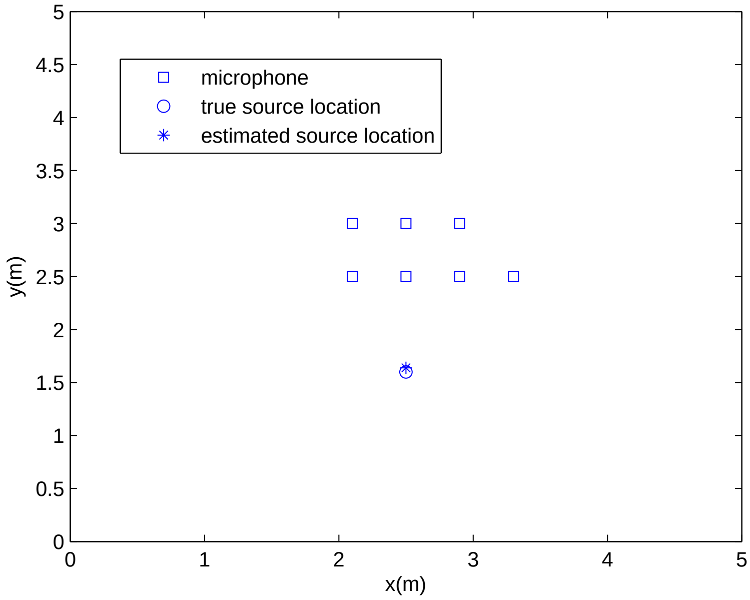

Figure 5.50: Example 8 with simulation, adding microphone (3.3,2.5), Bi-level method

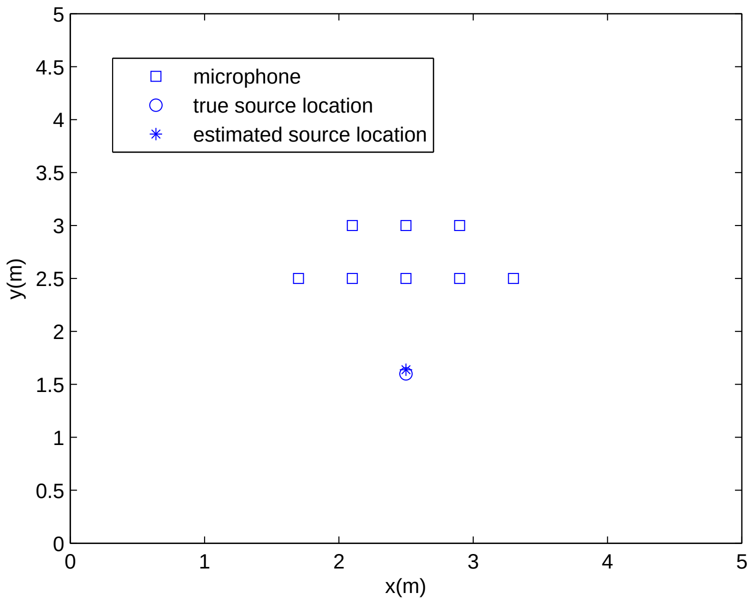

Figure 5.51: Example 8 with simulation, adding microphone (1.7,2.5), Bi-level method

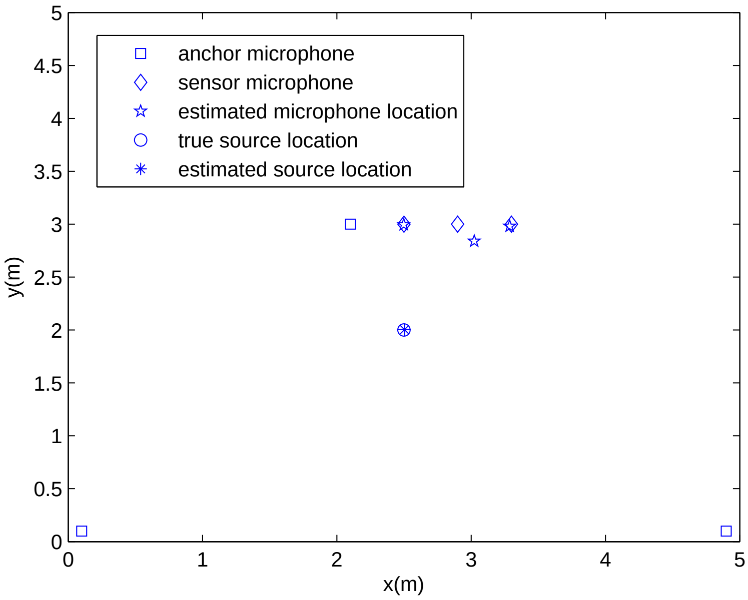

Figure 5.52: Mixed SDP-SOCP relaxation model for source localization combined with sensor network localization

5.6 Mixed SDP-SOCP relaxation model for source localization combined with sensor network localization

Here we give a example for Mixed SDP-SOCP relaxation model for source localization combined with sensor network localization. Assume we just know three anchor microphones $\boldsymbol{a}_i, i = 1, 2, 3$ located at the coordinates $\{(0.1, 0.1), (4.9, 0.1), (2.1, 3)\}$. The true positions of three sensor microphones $\boldsymbol{r}_i, i = 1, 2, 3$ are $\{(2.5, 3), (2.9, 3), (3.3, 3)\}$ and the true source location is $(2.5, 2)$. We know the distance between a_1 and r_1,a_2 and r_1,a_1 and r_2,a_2 and r_3,r_1 and r_3,r_2 and r_3. Figure 5.52 shows the estimated position of sensor microphones and source location.

Chapter 6

The review of time difference of arrival

In this chapter, we give four methods to find the time difference of arrival (TDOA), and use the cross correlation method to solve our actual problem.

6.1 Review of time difference of arrival

Generalized cross correlation(GCC) method

For a pair of microphones, the received signals at the two microphones can be expressed as

$$x_1(t) = s(t) + n_1(t)$$

$$x_2(t) = s(t + D) + n_2(t)$$

where D is the time difference of arrival(TDOA) of two microphones, and $\{n_i(t), t = 1, 2\}$ represent the noise. The Fourier transforms of these two microphone signals can be represent as

$$X_1(w) = S(w) + N_1(w)$$

$$X_2(w) = S(w)e^{jwD} + N_2(w)$$

Then we can obtain the cross power spectral density with no pre-filtering

$$\begin{aligned}\Phi_{x_1x_2}(w) &= X_1(w)X_2^*(w) \\ &= [S(w) + N_1(w)][S(w)e^{jwD} + N_2(w)]^* \\ &= S(w)S^*(w)e^{-jwD} + S(w)N_2^*(w) + S^*(w)N_1(w)e^{-jwD} + N_1(w)N_2^*(w) \\ &= \Phi_{ss}(w)e^{-jwD} + S(w)N_2^*(w) + S^*(w)N_1(w)e^{jw\tau_{12}} + N_1(w)N_2^*(w)\end{aligned}$$

where $\Phi_{ss}(w)$ is the power spectral density of the source signal.

The noise signals are assumed to be uncorrelated with the signal and each other, thus it is not necessary to do the cross-correlation computation for the last three terms in the above equation. The cross-correlation of signals is an inverse Fourier transform operation.

$$\begin{aligned}R_{x_1x_2}(\tau) &= F^{-1}[\Phi_{ss}(w)e^{-jwD}] \\ &= R_{ss}(\tau) * \delta(\tau - D) \\ &= \textstyle\int_{-\infty}^{\infty} R_{ss}(t) \cdot \delta(\tau - D - t)dt \\ &= \textstyle\int_{-\infty}^{\infty} R_{ss}(t) \cdot \delta(t - (\tau - D))dt \\ &= R_{ss}(\tau - D)\end{aligned}$$

Thus the TDOA estimate is calculated from

$$\hat{D} = \arg\max_{\tau} R_{x_1x_2}(\tau)$$

With the Fourier transforms of these filters denoted by $G_1(w)$ and $G_2(w)$, the generalized cross-correlation(GCC) function is given by

$$\begin{aligned}R_{x_1x_2}(\tau) &= \tfrac{1}{2\pi}\textstyle\int_{-\infty}^{\infty}(G_1(w)X_1(w))(G_2(w)X_2(w))^* e^{jw\tau}dw \\ &= \tfrac{1}{2\pi}\textstyle\int_{-\infty}^{\infty}(G_1(w)G_2^*(w)X_1(w)X_2^*(w)e^{jw\tau}dw\end{aligned}$$

Defining the frequency dependent weighting function(sometimes called prefilter) $\Psi(w) = G_1(w)G_2^*(w)$, the GCC function can be expressed as

$$R_{x_1x_2}(\tau) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \Psi(w)\Phi_{x_1x_2}(w)e^{jw\tau}dw$$

The prefilter used in the phase transform(PHAT) is

$$\Psi_{PHAT}(w) = \frac{1}{\mid \Phi_{x_1x_2}(w) \mid}$$

Then the generalized cross-correlation with phase transform(GCC-PHAT) is given by

$$\begin{aligned}
R_{x_1x_2}^{PHAT}(\tau) &= \tfrac{1}{2\pi}\int_{-\infty}^{\infty}\left(\tfrac{1}{|\Phi_{x_1x_2}(w)|}\right)\Phi_{x_1x_2}(w)e^{jw\tau}dw \\
&= \tfrac{1}{2\pi}\int_{-\infty}^{\infty}\left(\tfrac{1}{|\Phi_{ss}(w)e^{-jwD}|}\right)(\Phi_{ss}(w)e^{-jwD})e^{jw\tau}dw \\
&= \tfrac{1}{2\pi}\int_{-\infty}^{\infty}\left(\tfrac{1}{|\Phi_{ss}(w)|}\right)(\Phi_{ss}(w)e^{-jwD})e^{jw\tau}dw \\
&= \tfrac{1}{2\pi}\int_{-\infty}^{\infty}\left(\tfrac{1}{\Phi_{ss}(w)}\right)(\Phi_{ss}(w)e^{-jwD})e^{jw\tau}dw \\
&= \tfrac{1}{2\pi}\int_{-\infty}^{\infty} e^{-jwD}e^{jw\tau}dw \\
&= \delta(\tau - D)
\end{aligned}$$

Thus the TDOA estimate is calculated from

$$\hat{D} = \arg\max_{\tau} R_{x_1x_2}^{PHAT}(\tau)$$

The prefilter used in Maximum Likelihood(ML) is

$$\Psi_{ML}(w) = \frac{1}{\mid \Phi_{x_1x_2}(w) \mid}\frac{\mid \gamma_{x_1x_2}(w) \mid^2}{1- \mid \gamma_{x_1x_2}(w) \mid^2}$$

where $\mid \gamma_{x_1x_2}(w) \mid^2 = \frac{|\Phi_{x_1x_2}(w)|^2}{\Phi_{x_1x_2}(w)\Phi_{x_1x_2}(w)}$.

Average square difference function(ASDF) method

This method is to find the minimum error square between two received signals of microphone pair.

$$R_{ASDF}(\tau) = E([x_1(t) - x_2(t+\tau)]^2$$

Thus the TDOA estimate is calculated from

$$\hat{\tau} = \arg\min_{\tau} R_{ASDF}(\tau)$$

Least mean square(LMS) adaptive filter method

Assume $x_1(t)$ is the reference signal, $x_2(t)$ is the desired signal and a finite impulse response(FIR) filter of length L is $h(t) = [h_0, h_1, \cdots, h_{L-1}]$. The LMS output can be expressed as

$$y(t) = h^T(t)X_1(t)$$

where $X_1(t) = [x_1(t), x_1(t-1), \cdots, x_1(t-L+1)]^T$. The error output is

$$e(t) = x_2(t) - h^T(t)X_1(t)$$

An estimate $h(t)$ can be obtained by minimizing $E\{e^2(t)\}$ using adaptive algorithm. With the LMS adaptive algorithm, $h(t)$ is updated according to

$$h(t+1) = h(t) + \mu e(t)X_1(t)$$

where μ is a small positive adaptation step size which controls the rate of convergence. Then the TDOA estimate is given by

$$\hat{\tau} = \arg\max_l \mid h_l \mid .$$

Adaptive eigenvalue decomposition method

In the above methods, a common assumption is that each microphone receives only the direct-path signal. In this section, we introduce the adaptive eigenvalue decomposition method which fully into account the reverberation effect. In the reverberation model, the received signals for a pair of microphones are expressed as

$$x_i(t) = h_i * s(t) + n_i(t), \;\; i = 1, 2$$

where $*$ denotes the convolution and h_i is the channel impulse response between source and the i^{th} microphone.

If we ignore the noise term, we can get

$$x_1(t) * h_2 = s(t) * h_2 * h_1 = x_2(t) * h_1.$$

At the time instant t, above equation can be expressed as a vector matrix form

$$\mathbf{x}^T(t)\mathbf{w} = \mathbf{x}_1^T(t)\mathbf{h}_2 - \mathbf{x}_2^T(t)\mathbf{h}_1 = 0,$$

where

$$\mathbf{x}_i(t) = [x_i(t), x_i(t-1), \cdots, x_i(t-L+1)]^T, \;\; i = 1, 2,$$

$$\mathbf{x}(t) = [\mathbf{x}_1^T(t), \mathbf{x}_2^T(t)]^T,$$

$$\mathbf{w} = [\mathbf{h}_2^T, -\mathbf{h}_1^T]^T,$$

Then we have

$$\mathbf{x}(t)\mathbf{x}^T(t)\mathbf{w} = 0,$$

$$E\{\mathbf{x}(t)\mathbf{x}^T(t)\}\mathbf{w} = 0.$$

Let $\mathbf{R} = E\{\mathbf{x}(t)\mathbf{x}^T(t)\}$ which is the covariance matrix of the microphone signals, above equation can be expressed as

$$\mathbf{R}\mathbf{w} = 0, \tag{6.1.1}$$

which means $\mathbf{w}$ is in the null space of R. More specifically, $\mathbf{w}$ is the eigenvector of $\mathbf{R}$ corresponding to the eigenvalue 0. Thus we can determine those two channel impulse response $\mathbf{h}_1$ and $\mathbf{h}_2$.

With the estimated $\hat{\mathbf{h}}_1$ and $\hat{\mathbf{h}}_2$, the TDOA estimate can be determined by

$$\hat{\tau} = \arg\max_l \mid \hat{h}_{2,l} \mid - \arg\max_l \mid \hat{h}_{1,l} \mid$$

6.2 Experiment comparison

In this section, the same real data in chapter 5 is used to compare TDOA estimates obtained by the cross correlation method with the exact TDOA.

For rectangular shape microphone array, the detail of the comparison between the exact TDOA and the TDOA estimates obtained by the cross correlation method is given in Table 6.1, 6.2, 6.3 and 6.4. In the following tables, the upper

Table 6.1: Speaker 1, D=100cm, Rectangular shape

	M1	M2	M3	M4	M5	M6	M7	M8
M1	0	0.00032	0.00032	0	0.00105	0.00148	0.00148	0.00105
	0	0.000292	0.000292	-6.3E-05	0.001021	0.001438	0.001438	0.000958
M2	-0.00032	0	0	-0.00032	0.000729	0.00116	0.00116	0.000729
	-0.00029	0	-2.1E-05	-0.00038	0.000708	0.001146	0.001125	0.000667
M3	-0.00032	0	0	-0.00032	0.000729	0.00116	0.00116	0.000729
	-0.00029	2.08E-05	0	-0.00035	0.000729	0.001167	0.001146	0.000688
M4	0	0.00032	0.00032	0	0.00105	0.00148	0.00148	0.00105
	6.25E-05	0.000375	0.000354	0	0.001083	0.001521	0.0015	0.001042
M5	-0.00105	-0.00073	-0.00073	-0.00105	0	0.000431	0.000431	0
	-0.00102	-0.00071	-0.00073	-0.00108	0	0.000417	0.000417	-4.2E-05
M6	-0.00148	-0.00116	-0.00116	-0.00148	-0.00043	0	0	-0.00043
	-0.00144	-0.00115	-0.00117	-0.00152	-0.00042	0	-2.1E-05	-0.00048
M7	-0.00148	-0.00116	-0.00116	-0.00148	-0.00043	0	0	-0.00043
	-0.00144	-0.00113	-0.00115	-0.0015	-0.00042	2.08E-05	0	-0.00046
M8	-0.00105	-0.00073	-0.00073	-0.00105	0	0.000431	0.000431	0
	-0.00096	-0.00067	-0.00069	-0.00104	4.17E-05	0.000479	0.000458	0

Table 6.2: Speaker 2, D=100cm, Rectangular shape

	M1	M2	M3	M4	M5	M6	M7	M8
M1	0	0.00083	0.001511	0.001971	0.000704	0.001659	0.002486	0.003085
	0	0.000771	0.001375	0.001771	0.000729	0.001646	0.002417	0.002938
M2	-0.00083	0	0.000681	0.001141	-0.00013	0.000829	0.001657	0.002256
	-0.00077	0	0.000625	0.001	-4.2E-05	0.000875	0.001646	0.002167
M3	-0.00151	-0.00068	0	0.00046	-0.00081	0.000148	0.000976	0.001575
	-0.00138	-0.00063	0	0.000375	-0.00067	0.000271	0.001042	0.001542
M4	-0.00197	-0.00114	-0.00046	0	-0.00127	-0.00031	0.000516	0.001115
	-0.00177	-0.001	-0.00038	0	-0.00104	-0.00013	0.000646	0.001167
M5	-0.0007	0.000126	0.000807	0.001267	0	0.000955	0.001783	0.002382
	-0.00073	4.17E-05	0.000667	0.001042	0	0.000917	0.001688	0.002208
M6	-0.00166	-0.00083	-0.00015	0.000312	-0.00096	0	0.000828	0.001427
	-0.00165	-0.00088	-0.00027	0.000125	-0.00092	0	0.000771	0.001292
M7	-0.00249	-0.00166	-0.00098	-0.00052	-0.00178	-0.00083	0	0.000599
	-0.00242	-0.00165	-0.00104	-0.00065	-0.00169	-0.00077	0	0.000521
M8	-0.00309	-0.00226	-0.00157	-0.00111	-0.00238	-0.00143	-0.0006	0
	-0.00294	-0.00217	-0.00154	-0.00117	-0.00221	-0.00129	-0.00052	0

Table 6.3: Speaker 1, D=200cm, Rectangular shape

	M1	M2	M3	M4	M5	M6	M7	M8
M1	0	0.000193	0.000193	0	0.001135	0.001364	0.001364	0.001135
	0	0.000167	0.000167	-6.3E-05	0.001104	0.001333	0.001333	0.001104
M2	-0.00019	0	0	-0.00019	0.000942	0.001172	0.001172	0.000942
	-0.00017	0	-2.1E-05	-0.00023	0.000938	0.001146	0.001167	0.000938
M3	-0.00019	0	0	-0.00019	0.000942	0.001172	0.001172	0.000942
	-0.00017	2.08E-05	0	-0.00021	0.000938	0.001167	0.001167	0.000938
M4	0	0.000193	0.000193	0	0.001135	0.001364	0.001364	0.001135
	6.25E-05	0.000229	0.000208	0	0.001146	0.001375	0.001396	0.001146
M5	-0.00113	-0.00094	-0.00094	-0.00113	0	0.00023	0.00023	0
	-0.0011	-0.00094	-0.00094	-0.00115	0	0.000229	0.000229	0
M6	-0.00136	-0.00117	-0.00117	-0.00136	-0.00023	0	0	-0.00023
	-0.00133	-0.00115	-0.00117	-0.00138	-0.00023	0	0	-0.00023
M7	-0.00136	-0.00117	-0.00117	-0.00136	-0.00023	0	0	-0.00023
	-0.00133	-0.00117	-0.00117	-0.0014	-0.00023	0	0	-0.00023
M8	-0.00113	-0.00094	-0.00094	-0.00113	0	0.00023	0.00023	0
	-0.0011	-0.00094	-0.00094	-0.00115	0	0.000229	0.000229	0

Table 6.4: Speaker 2, D=200cm, Rectangular shape

	M1	M2	M3	M4	M5	M6	M7	M8
M1	0	0.000592	0.001043	0.001327	0.000951	0.001624	0.002148	0.002485
	0	0.000542	0.000938	0.001167	0.000938	0.001583	0.002083	0.002375
M2	-0.00059	0	0.000451	0.000736	0.000359	0.001032	0.001557	0.001893
	-0.00054	0	0.000396	0.000625	0.000417	0.001063	0.001542	0.001833
M3	-0.00104	-0.00045	0	0.000284	-9.2E-05	0.000581	0.001105	0.001442
	-0.00094	-0.0004	0	0.000229	0	0.000646	0.001146	0.001438
M4	-0.00133	-0.00074	-0.00028	0	-0.00038	0.000296	0.000821	0.001157
	-0.00117	-0.00063	-0.00023	0	-0.00021	0.000438	0.000917	0.001208
M5	-0.00095	-0.00036	9.24E-05	0.000377	0	0.000673	0.001198	0.001534
	-0.00094	-0.00042	0	0.000208	0	0.000646	0.001125	0.001417
M6	-0.00162	-0.00103	-0.00058	-0.0003	-0.00067	0	0.000524	0.000861
	-0.00158	-0.00106	-0.00065	-0.00044	-0.00065	0	0.000479	0.000771
M7	-0.00215	-0.00156	-0.00111	-0.00082	-0.0012	-0.00052	0	0.000337
	-0.00208	-0.00154	-0.00115	-0.00092	-0.00113	-0.00048	0	0.000292
M8	-0.00248	-0.00189	-0.00144	-0.00116	-0.00153	-0.00086	-0.00034	0
	-0.00238	-0.00183	-0.00144	-0.00121	-0.00142	-0.00077	-0.00029	0

row presents the exact TDOA and the lower row presents the TDOA estimates obtained by the cross correlation method.

For L shape microphone array, the detail of the comparison between the exact TDOA and the TDOA estimates obtained by the cross correlation method is given in Table 6.5, 6.6, 6.7 and 6.8.

Table 6.5: Speaker 1 D=100cm, L shape

	M1	M2	M3	M4	M5	M6	M7	M8
M1	0	0.000431	0.000431	0	0	0.000431	0.000431	0
	0	0.000542	0.000688	0.000333	0.000188	0.000542	0.000479	-6.3E-05
M2	-0.00043	0	0	-0.00043	-0.00043	0	0	-0.00043
	-0.00054	0	0.000125	-0.00021	-0.00035	0	-8.3E-05	-0.0006
M3	-0.00043	0	0	-0.00043	-0.00043	0	0	-0.00043
	-0.00069	-0.00013	0	-0.00035	-0.0005	-0.00013	-0.00021	-0.00075
M4	0	0.000431	0.000431	0	0	0.000431	0.000431	0
	-0.00033	0.000208	0.000354	0	-0.00015	0.000208	0.000125	-0.0004
M5	0	0.000431	0.000431	0	0	0.000431	0.000431	0
	-0.00019	0.000354	0.0005	0.000146	0	0.000354	0.000271	-0.00025
M6	-0.00043	0	0	-0.00043	-0.00043	0	0	-0.00043
	-0.00054	0	0.000125	-0.00021	-0.00035	0	-8.3E-05	-0.0006
M7	-0.00043	0	0	-0.00043	-0.00043	0	0	-0.00043
	-0.00048	8.33E-05	0.000208	-0.00013	-0.00027	8.33E-05	0	-0.00052
M8	0	0.000431	0.000431	0	0	0.000431	0.000431	0
	6.25E-05	0.000604	0.00075	0.000396	0.00025	0.000604	0.000521	0

Table 6.6: Speaker 2, D=100cm, L shape

	M1	M2	M3	M4	M5	M6	M7	M8
M1	0	0.00023	0.00023	0	0.000592	0.001547	0.002375	0.002974
	0	0.00025	0.000292	0.000125	0.000563	0.001479	0.00225	0.002729
M2	-0.00023	0	0	-0.00023	0.000362	0.001317	0.002145	0.002744
	-0.00025	0	6.25E-05	-0.0001	0.000333	0.00125	0.002021	0.0025
M3	-0.00023	0	0	-0.00023	0.000362	0.001317	0.002145	0.002744
	-0.00029	-6.3E-05	0	-0.00017	0.000292	0.001188	0.001958	0.002438
M4	0	0.00023	0.00023	0	0.000592	0.001547	0.002375	0.002974
	-0.00013	0.000104	0.000167	0	0.000438	0.001354	0.002104	0.002604
M5	-0.00059	-0.00036	-0.00036	-0.00059	0	0.000955	0.001783	0.002382
	-0.00056	-0.00033	-0.00029	-0.00044	0	0.000917	0.001667	0.002146
M6	-0.00155	-0.00132	-0.00132	-0.00155	-0.00096	0	0.000828	0.001427
	-0.00148	-0.00125	-0.00119	-0.00135	-0.00092	0	0.00075	0.00125
M7	-0.00237	-0.00215	-0.00215	-0.00237	-0.00178	-0.00083	0	0.000599
	-0.00225	-0.00202	-0.00196	-0.0021	-0.00167	-0.00075	0	0.000479
M8	-0.00297	-0.00274	-0.00274	-0.00297	-0.00238	-0.00143	-0.0006	0
	-0.00273	-0.0025	-0.00244	-0.0026	-0.00215	-0.00125	-0.00048	0

Table 6.7: Speaker 1, D=200cm, L shape

	M1	M2	M3	M4	M5	M6	M7	M8
M1	0	-0.0006	-0.00143	-0.00238	-0.00297	-0.00274	-0.00274	-0.00297
	0	-0.0005	-0.00125	-0.00217	-0.0029	-0.00269	-0.00269	-0.00294
M2	0.000599	0	-0.00083	-0.00178	-0.00237	-0.00215	-0.00215	-0.00237
	0.0005	0	-0.00075	-0.00167	-0.00242	-0.00219	-0.00221	-0.00244
M3	0.001427	0.000828	0	-0.00096	-0.00155	-0.00132	-0.00132	-0.00155
	0.00125	0.00075	0	-0.00092	-0.00165	-0.00144	-0.00144	-0.00169
M4	0.002382	0.001783	0.000955	0	-0.00059	-0.00036	-0.00036	-0.00059
	0.002167	0.001667	0.000917	0	-0.00073	-0.00052	-0.00052	-0.00077
M5	0.002974	0.002375	0.001547	0.000592	0	0.00023	0.00023	0
	0.002896	0.002417	0.001646	0.000729	0	0.000208	0.000208	-4.2E-05
M6	0.002744	0.002145	0.001317	0.000362	-0.00023	0	0	-0.00023
	0.002688	0.002188	0.001438	0.000521	-0.00021	0	0	-0.00025
M7	0.002744	0.002145	0.001317	0.000362	-0.00023	0	0	-0.00023
	0.002688	0.002208	0.001438	0.000521	-0.00021	0	0	-0.00025
M8	0.002974	0.002375	0.001547	0.000592	0	0.00023	0.00023	0
	0.002938	0.002438	0.001688	0.000771	4.17E-05	0.00025	0.00025	0

Table 6.8: Speaker 2, D=200cm, L shape

	M1	M2	M3	M4	M5	M6	M7	M8
M1	0	-0.00034	-0.00086	-0.00153	-0.00153	-0.00086	-0.00034	0
	0	-0.00023	-0.00067	-0.00127	-0.0014	-0.00075	-0.00025	4.17E-05
M2	0.000337	0	-0.00052	-0.0012	-0.0012	-0.00052	0	0.000337
	0.000229	0	-0.00044	-0.00104	-0.00119	-0.00052	-2.1E-05	0.000271
M3	0.000861	0.000524	0	-0.00067	-0.00067	0	0.000524	0.000861
	0.000667	0.000438	0	-0.00063	-0.00075	-8.3E-05	0.000417	0.000708
M4	0.001534	0.001198	0.000673	0	0	0.000673	0.001198	0.001534
	0.001271	0.001042	0.000625	0	-0.00013	0.000521	0.001021	0.001313
M5	0.001534	0.001198	0.000673	0	0	0.000673	0.001198	0.001534
	0.001396	0.001188	0.00075	0.000125	0	0.000646	0.001146	0.001438
M6	0.000861	0.000524	0	-0.00067	-0.00067	0	0.000524	0.000861
	0.00075	0.000521	8.33E-05	-0.00052	-0.00065	0	0.0005	0.000792
M7	0.000337	0	-0.00052	-0.0012	-0.0012	-0.00052	0	0.000337
	0.00025	2.08E-05	-0.00042	-0.00102	-0.00115	-0.0005	0	0.000292
M8	0	-0.00034	-0.00086	-0.00153	-0.00153	-0.00086	-0.00034	0
	-4.2E-05	-0.00027	-0.00071	-0.00131	-0.00144	-0.00079	-0.00029	0

Conclusions

In this thesis, the design of distributed broadband beamforming system is studied. We use two methods to solve the minimax optimization problem. First, it is transformed into a semi-definite programming problem so that interior point algorithms can be applied. Also it can be formulated as a non-strictly convex semi-infinite programming problem, adding a small perturbation quadratic function to the objective function to make it strictly convex and the new exchange algorithm is applied. For the source localization problem, we formulate a new mixed SDP-SOCP relaxation model. We obtain a representation for the solution of the mixed SDP-SOCP model and the characterization such that the mixed SDP-SOCP model has an exact relaxation in two-dimensional case. Moreover, we derive the geometry of the localizable region for the proposed mixed model. Many illustrated examples demonstrate those approaches can be applied successfully and some comparisons are presented.

Appendices—The introduction of Matlab codes

Matlab codes for chapter 2

In chapter 2, we use two methods to solve the beamforming design problem, one is semi-definite programming, the other one is semi-infinite programming which actually using multiple exchange algorithm. For solving the semi-definite programming problem, we use the software package SDPA-M (Semidnite Programming Algorithm in MATLAB) Version 2.0 (Fujisawa et al. [2005]) introduced below.

SDPA-M

SDPA-M Version 2.0 which is a Matlab interface of the SDPA Version 6.0 is a fast and numerically stable solver for semi-definite programming problems.

The SDPA-M solves the following standard form semidenite program

$$\begin{aligned} \min \quad & \sum_{i=1}^{m} c_i x_i \\ \text{s.t.} \quad & \mathbf{X} = \sum_{i=1}^{m} \mathbf{F}_i x_i - \mathbf{F}_0,\ \mathbf{X} \succeq 0, \end{aligned}$$

and its dual

$$\begin{aligned} \max\ & \mathbf{F}_0 \cdot \mathbf{Y} \\ \text{s.t.}\ & \mathbf{F}_i \cdot \mathbf{Y} = c_i(i = 1, \cdots m),\ \mathbf{Y} \succeq 0, \end{aligned}$$

Problem data input

- mDIM — Number of primal variable.
- nBLOCK,bLOCKsTRUCT — The number of blocks, the block structure vector. If we deal with a block diagonal matrix F of the form

  $$\mathbf{F} = \begin{pmatrix} \mathbf{B}_1 & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_2 & \mathbf{0} & \cdots & \mathbf{0} \\ \cdot & \cdot & \cdot & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{B}_l \end{pmatrix}$$

  where $\mathbf{B}_i$ is a $p_i \times p_i$ symmetric matrix $(i = 1, 2, \cdots, l)$.

  Define

  $$\begin{aligned} \text{nBLOCK} &= l, \\ \text{bLOCKsTRUCT} &= (\beta_1, \beta_2, \cdots, \beta_l), \\ \beta_i &= \begin{cases} p_i & \text{if } \mathbf{B}_i \text{ is a } p_i \times p_i \text{ symmetric matrix,} \\ -p_i & \text{if } \mathbf{B}_i \text{ is a } p_i \times p_i \text{ diagonal matrix.} \end{cases} \end{aligned}$$

- **c** — Constant vector. We write all the elements $c_1, c_2, \cdots, c_m$ of the constant vector **c**.
- **F** — Constraint matrices. **F** is nBLOCK×(mDIM+1) cell matrix. Here each $F\{i, j\}$ denotes the ith block of $\mathbf{F}_{j-1}$ $(i = 1, \cdots, \text{nBLOCK}, j = 1, \cdots, \text{mDIM+1})$.

The main function is sdpam.m whose calling syntax is as follows:

$$\begin{aligned} &[\text{objVal}, x, X, Y] \\ &= \text{sdpam}(\text{mDIM}, \text{nBLOCK}, \text{bLOCKsTRUCT}, c, F, x0, X0, Y0, \text{OPTION}) \end{aligned}$$

For our beamforming design problem, since the passband region and stopband region have six block together. We construct the constraint matrix F to be a $z*z*6$ by $n*l+2$ cell where z is the number of discretization, n is number of microphones and l is the filter length. Thus mDIM $= n * l + 1$, nBLOCK $= z * z * 6$ and bLOCKsTRUCT $= 3 * \text{ones}(1, z * z * 6)$. The constant vector $c = [1 \ \text{zeros}(1, n * l)]$.

Perturbation multiple exchange algorithm

The beamforming design problem is also converted into a semi-infinite programming problem, which is solved by the perturbation multiple exchange algorithm presented in chapter 2. Here we use the function quadprog in Matlab to solve. In each iteration, the discretization method is used to find the points that are greater than η.

Matlab codes for chapter 5

In chapter 5, The software package SDPT3 introduced below is used for solving the mixed SDP-SOCP relaxation model.

SDPT3

SDPT3 is designed to solve conic programming problems whose constraint cone is a product of semidenite cones, second-order cones, nonnegative orthants and Euclidean spaces. Objective function is the sum of linear functions and log-barrier terms associated with the constraint cones.

The SDPT3 solves the following standard form of such SQLP problems

$$\begin{aligned}
\min\ & \sum_{j=1}^{n_s}[< c_j^s, x_j^s > -v_j^s \mathrm{logdet}(x_j^s)] + \sum_{i=1}^{n_q}[< c_i^q, x_i^q > -v_i^q \log\gamma(x_i^q)] \\
& + < c^l, x^l > - \sum_{k=1}^{n_l} v_k^l \log(x_k^l) + < c^u, x^u > \\
\text{s.t.}\ & \sum_{j=1}^{n_s} \mathcal{A}_j^s(x_j^s) + \sum_{i=1}^{n_q} A_i^q(x_i^q) + A^l x^l + A^u x^u = b, \\
& x_j^s \in K_s^{s_j} \forall j,\ x_i^q \in K_q^{q_i} \forall i,\ x^l \in K_l^{n_l}\ x^u \in \mathbb{R}^{n_u}.
\end{aligned}$$

where $K_s^{s_j}$ is the cone of positive semidenite symmetric matrices of dimension s_j, $K_q^{q_i}$ is the quadratic or second-order cone of dimension q_i, $K_l^{n_l}$ is the nonnegative orthant of dimension n_l.

It also solves the dual problem

$$\begin{aligned}
\max\ & b^T y + \sum_{j=1}^{n_s}[v_j^s \mathrm{logdet}(z_j^s) + s_j v_j^s (1 - \log v_j^s)] \\
& + \sum_{i=1}^{n_q}[v_i^q \log\gamma(z_i^q) + v_i^q(1 - \log v_i^q)] + \sum_{k=1}^{n_l}[v_k^l \log(z_k^l) + v_k^l(1 - \log v_k^l)] \\
\text{s.t.}\ & (\mathcal{A}_j^s)^T y + z_j^s = c_j^s,\ z_j^s \in K_s^{s_j},\ j = 1, \cdots, n_s \\
& (\mathcal{A}_i^q)^T y + z_i^q = c_i^q,\ z_i^q \in K_q^{q_i},\ i = 1, \cdots, n_q \\
& (\mathcal{A}^l)^T y + z^l = c^l,\ z^l \in K_l^{n_l}, \\
& (\mathcal{A}^u)^T y = c^u,\ y \in \mathbb{R}^m.
\end{aligned}$$

Problem data input

For the above SQLP problem, let L be the total number of blocks in it, the block structure of the problem data is described by an $L \times 2$ cell array named blk.

If the jth block is a semidenite block consisting of a single block of size s_j, then the input data is

$$\begin{aligned}
&\text{blk}\{j,1\} ='s', \;\; \text{blk}\{j,2\} = [s_j],\\
&\text{At}\{j\} = [\bar{s_j} \times m \text{ sparse}],\\
&C\{j\}, X\{j\}, Z\{j\} = [s_j \times s_j \text{ double or sparse}],
\end{aligned}$$

where $\bar{s_j} = s_j(s_j+1)/2$.

If the jth block is a semidenite block consisting of numerous small sub-blocks, say p of them, of dimensions $s_{j1}, s_{j2}, \cdots, s_{jp}$ such that $\sum_{k=1}^{p} s_{jp} = s_j$, then the input data is

$$\begin{aligned}
&\text{blk}\{j,1\} ='s', \;\; \text{blk}\{j,2\} = [s_{j1}, s_{j2}, \cdots, s_{jp}],\\
&\text{At}\{j\} = [\bar{s_j} \times m \text{ sparse}],\\
&C\{j\}, X\{j\}, Z\{j\} = [s_j \times s_j \text{ sparse}],
\end{aligned}$$

where $\bar{s_j} = \sum_{k=1}^{p} s_{jk}(s_{jk}+1)/2$.

If the ith block is a quadratic block consisting of numerous sub-blocks, say p of them, of dimensions $q_{i1}, q_{i2}, \cdots, q_{ip}$ such that $\sum_{k=1}^{p} q_{ip} = q_i$, then the input data is

$$\begin{aligned}
&\text{blk}\{i,1\} ='q', \;\; \text{blk}\{i,2\} = [q_{i1}, q_{i2}, \cdots, q_{ip}],\\
&\text{At}\{i\} = [q_i \times m \text{ sparse}],\\
&C\{i\}, X\{i\}, Z\{i\} = [q_i \times 1 \text{ double or sparse}],
\end{aligned}$$

If the kth block is the linear block, then the input data is

$$\begin{aligned}
&\text{blk}\{k,1\} ='l', \;\; \text{blk}\{k,2\} = n_l,\\
&\text{At}\{k\} = [n_l \times m \text{ sparse}],\\
&C\{k\}, X\{k\}, Z\{k\} = [n_l \times 1 \text{ double or sparse}],
\end{aligned}$$

If the kth block is the unrestricted block, then the input data is

$$\begin{aligned}
&\text{blk}\{k,1\} ='u', \;\; \text{blk}\{k,2\} = n_u,\\
&\text{At}\{k\} = [n_u \times m \text{ sparse}],\\
&C\{k\}, X\{k\}, Z\{k\} = [n_u \times 1 \text{ double or sparse}],
\end{aligned}$$

The main function is sqlp.m whose calling syntax is as follows:

$$[\text{obj}, X, y, Z, \text{info}, \text{runhist}] = \text{sqlp}(\text{blk}, \text{At}, C, b, \text{OPTIONS}, X0, y0, Z0).$$

For our mixed SDP-SOCP relaxation model, in writing matlab code, we construct 3 parts for block. One is $\boldsymbol{Z}$, it is a positive semi-definite matrix, then we have $\text{blk}\{1,1\} =' s'$, $\text{blk}\{1,2\} = 3$. One is $\begin{pmatrix} \beta_i & 0 & 0 \\ 0 & 1 & \beta_i \\ 0 & \beta_i & \alpha_i \end{pmatrix}$, it is also a positive positive semi-definite matrix, then we have $\text{blk}\{p,1\} =' s'$, $\text{blk}\{p,2\} = 3$. At last, one is $\begin{pmatrix} \beta_i \\ \mathbf{s} - \boldsymbol{a}_i \end{pmatrix}$, it is a vector in second order cone, then we have $\text{blk}\{p,1\} =' q'$, $\text{blk}\{p,2\} = 3$. The corresponding At, C, b are the coefficients for those three type variables in the mixed SDP-SOCP relaxation model.

Simulation

For the simulation part, first the matlab code written by Eric A. Lehmann for the purpose of simulating reverberant audio data in small-room acoustics is used. Then we use the cross correlation method introduced in chapter 6 to find the time difference of arrival. Here the function named xcorr in Matlab is used. After finding the TDOA, we try to solve the relaxation models by SDPT3.

Bibliography

F Alizadeh and D Goldfarb. Second-order cone programming. *Math. Programming, Ser. B*, 95:3–51, 2003.

J Aspnes, D K Goldenberg, and Y R Yang. On the computational complexity of sensor network localization. *Algorithmic Aspects of Wireless Sensor Networks: First International Workshop*, 3121:32–44, 2004.

J Aspnes, T Eren, D K Goldenberg, A S Morse, W Whiteley, Y R Yang, B D O Anderson, and P N Belhumeur. A theory of network localization. *IEEE Transactions on Mobile Computing*, 5(12):1663–1678, 2006.

A Auslender, M A Goberna, and M A López. Penalty and smoothing methods for convex semi-infinite programming. *Math. Oper. Res.*, 34:303–319, 2009.

G J Babu. Strong representations for lad estimators in linear models. *Probab. Th. Rel. Fields*, 83:547–558, 1989.

R J Barsanti and M Tummala. Wavelet-based time delay estimates for transient signals. *IEEE Signals, Systems and Computers*, 1:1173–1177, 2003.

S J Benson, Y Ye, and X Zhang. Solving large-scale sparse semidefinite programs for combinatorial optimization. *SIAM Journal on Optimization*, 10(2):443–461, 2000.

D Bertsimas and Y Ye. Semidefinite relaxations, multivariate normal distributions, and order statistics. *Hankbook of Combinatorial Optimization*, 3:1–19, 1998.

B Betró. An accelerated central cutting plane algorithm for linear semi-infinite programming. *Math. Program.*, 101:479–495, 2004.

P Biswas and Y Ye. Semidefinite programming for ad hoc wireless network localization. *IPSN*, pages 46–54, 2004.

P Biswas, T Lian, T Wang, and Y Ye. Semidefinite programming based algorithms for sensor network localization. *IPSN*, pages 46–54, 2004.

P Biswas, T C Lian, T C Wang, and Y Ye. Semidefinite programming based algorithms for sensor network localization. *ACM Transactions on Sensor Networks*, 2(2):188–120, 2006a.

P Biswas, T C Liang, K C Toh, Y Ye, and T C Wang. Semidefinite programming approaches for sensor network localization with noisy distance measurements. *IEEE Transactions on Automation Science and Engineering*, 3(4):360–371, 2006b.

S Boyd, L E Ghaoui, E Feron, and V Balakrishnan. Linear matrix inequalities in system and control theory. *SIAM*, 1994.

J Bruck, J Gao, and A Jiang. Localization and routing in sensor networks by local angle information. *SIGMOBILE*, pages 181–192, 2005.

C G Carter. *Coherence and time delay estimation: an applied tutorial for research, development, test, and evaluation engineers.* Piscataway, NJ: IEEE Press, 1993.

S C Chan and H H Chen. Uniform concentric circular arrays with frequency-invariant characteristics-theory, design, adaptive beamforming and DOA estimation. *IEEE Trans. Signal Processing*, 55(1):165–177, 2007.

Y Chan and K Ho. A simple and efficient estimator for hyperbolic location. *IEEE Transactions on Signal Processing*, 42(8):1905–1915, 1994.

Z J Chen. *Higher Algebra and Analytic Geometry.* China Higher Education Press Beijing and Springer-Verlag Berlin Heidelberg, 2001.

E W Cheney. *Introduction to Approximation Theory.* Chelsea, New York, 1982.

J H Choi and C C Lim. A cholesky factorization based approach for blind FIR channel identification. *IEEE Trans. Signal Processing*, 56(4):1730–1735, 2008.

L C Chu and U Mitra. Analysis of music-based delay estimator for direct-sequence code-division multiple-access systems. *IEEE Transactions on Communications*, 47(1):133–138, 1999.

H Dam, K Teo, S Nordebo, and A Cantoni. The dual parametrization approach to optimal least square fir filters design subject to maximum error constraints. *IEEE Trans on Signal Processing*, 48:2314–2320, 2000.

L Doherty, K S J Pister, and L El Ghaoui. Convex position estimation in wireless sensor networks. *IEEE INFOCOM*, 3:1655–1663, 2001.

T Eren, E K Goldenber, W Whiteley, and Y R Yang. Rigidity, computation, and randomization in network localization. *IEEE INFOCOM*, 4:2673–2684, 2004.

K Fujisawa, Y Futakata, M Kojima, S Matsuyama, S Nakamura, K Nakata, and M Yamashita. Seidefinite programming algorithm in matlab user's manual-version 6.2.0. *Research Reports on Mathematical and Computing Sciences*, 2005.

P Gahinet, A Nemirovski, A J Laub, and M Chilali. LMI control toolbox for use with matlab user's guide version 1. 1995.

S Gannot, D Burshtein, and E Weinstein. Signal enhancement using beamforming and nonstationarity with applications to speech. *IEEE Trans. Signal Processing*, 49(8):1614–1626, 2001.

J Gao, J Bruck, and A A Jiang. Localization and routing in sensor networks by local angle information. *ACM Transactions on Sensor Networks*, 5(1): 181–192, 2009.

M Gillette and H Silverman. A linear closed-form algorithm for source localization from time-differences of arrival. *IEEE Signal Process. Lett*, 15:1–4, 2008.

K Glashoff and K Roleff. A new method for chebyshev approximation of complex-valued functions. *Math. Comp.*, 36:233–239, 1981.

M A Goberna and M A López. *Linear Semi-Infinite Optimization.* Wiley, New York, 1998.

M A Goberna and M A López. Linear semi-infinite programming theory: An updated survey. *European J. Oper. Res.*, 143:390–405, 2002.

M X Goemans and D P Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42(6):1115–1145, 1995.

W Hahn and S Tretter. Optimum processing for delay-vector estimation in passive signal arrays. *IEEE Transactions on Information Theory*, 19(5):608–614, 1973.

R Hettich. An implementation of a discretization method for semi-infinite programming. *Math. Program.*

R Hettich and G Gramlich. A note on an implementation of a method for quadratic semi-infinite programming. *Math. Program.*, 46:249–254, 1990.

R Hettich and K O Kortanek. Semi-infinite programming: Theory, methods and applications. *SIAM Rev.*, 35:380–429, 1993.

O Hoshuyama, A Sugiyama, and A Hirano. A robust adaptive beamformer for microphone arrays with a blocking matrix using constrained adaptive filters. *IEEE Trans. Signal Processing*, 47(10):2677–2684, 1999.

X Huang, X Yang, and K Teo. A sequential quadratic penalty method for nonlinear semidefinite programming. *Optimization*, 52:715–738, 2003.

R S Irving. *Integers, Polynomials, and Rings: A Course in Algebra.* Springer, 2003.

A Jakobsson, A L Swindlehurst, and P Stoica. Subspace-based estimation of time delays and doppler shifts. *IEEE Transactions On Signal Processing*, 46(9): 2472–2483, 1998.

N Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4:373–395, 1984.

R A Kennedy, T D Abhayapala, and D B Ward. Broadband nearfield beamforming using radial beampattern transformation. *IEEE Trans. Signal Processing*, 46: 2147–2156, 1998.

R A Kennedy, D B Ward, and T D Abhayapala. Nearfield beamforming using radial reciprocity. *IEEE Trans. Signal Processing*, 47:33–40, 1999.

H H Kha, H D Tuan, B Vo, and T Q Nguyen. Symmetric orthogonal complex-valued filter bank design by semidefinite programming. *IEEE Transacions on Signal Processing*, 55(9):4405–4414, 2007.

M Khosravy, M R Asharif, and K Yamashita. A pdf-matched short-term linear predictability approach to blind source separation. *International Journal of Innovative Computing, Information and Control*, 5:3677–3690, 2009.

C H Knapp and C G Carter. The generalized correlation method for estimation of time delay. *IEEE Trans, Acoust, Speech, Signal Processing*, ASSP-21:320–327, 1976.

K O Kortanek and H No. A central cutting plane algorithm for convex semi-infinite programming problems. *SIAM J. Optim.*, 3:901–918, 1993.

N Krislock and H Wolkowicz. Explicit sensor network localization using semidefinite representations and clique reductions. *SIAM Journal on Optimization*, 20(5):2679–2708, 2010.

H C Lai and S Y Wu. On linear semi-infinite programming problems–an algorithm. *Numer. Funct. Anal. Optim.*, 13:287–304, 1992.

B K Lau, Y H Leung, K L Teo, and V Sreeram. Minimax filters for microphone arrays. *IEEE Transactions on Circuits and Systems II*, 46(12):1522–1525, 1999.

P J Laurent and C Carasso. An algorithm of successive minimization in convex programming. *Numer. Funct. Anal. Optim.*, 12:377–400, 1978.

W Lee, L Caccetta, K Teo, and V Rehbock. Optimal design of complex fir filters with arbitrary magnitude and group delay responses. *IEEE Trans on Signal Processing*, 54:1617–1627, 2006.

E Lehmann and A Johansson. Prediction of energy decay in room impulse responses simulated with an image-source model. *Journal of the Acoustical Society of America*, 124(1):269–277, 2008.

S Li, X Yang, K Teo, and S Wu. A solution method for combined semi-infinite and semi-definite programming. *ANZIAM Journal*, 45:477–494, 2003.

X Li. *Wireless ad hoc and sensor networks: Theory and applications.* Cambridge University Press, New York, 2008.

Y C Lim and Y Lian. The optimum design of one- and two-dimensional FIR filters using the frequency response masking technique. *IEEE Trans. Circuits and Systems II*, 40(2):88–95, 1993.

W Liu and S Weiss. Design of frequency invariant beamformers for broadband arrays. *IEEE Transactions on Signal Processing*, 56, 2008.

Y Liu, S Ito, H W J Lee, and K L Teo. Semi-infinite programming approach to continuously-constrained linear-quadratic optimal control problems. *Journal of Optimization Theory and Applications*, 108:617–632, 2001.

W S Lu and T Hinamoto. Optimal design of frequency-response-masking filters using semidefinite programming. *IEEE Transacions on Circuits and Systems-I:Fundamental Theory and Applications*, 50(4):715–738, 2003.

D G Luenberger. *Linear and Nonlinear Programming.* Kluwer Academic, Boston/Dordrecht/London, 2004.

Y Makoto, F Katsuki, and K Masakazu. Implementation and evaluation of sdpa 6.0 (semidefinite programming algorithm 6.0). *Optimization Methods and Software*, 18(4):491–505, 2003.

Y Nesterov and A Nemirovskii. *Interior Point Polynomial Algorithms in Convex Programming.* SIAM, 1994.

B P Ng and H P Duan. Designing amplitude/phase response of the frost beamformer. *IEEE Transactions on Signal Processing*, 55, 2007.

S Nordebo, I Claesson, and S Nordholm. Weighted chebyshev approximation for the design of broadband beamformers using quadratic programming. *IEEE Signal Processing Letters*, 1:103–105, 1994.

S Nordholm, V Rehbock, K L Teo, and S Nordebo. Chebyshev approximation for the design of broadband beamformers in the near field. *IEEE Transactions on Circuits and Systems II*, 45(1):141–143, 1998.

A Potchinkov and R Reemsten. The design of fir filter in the complex plane by convex optimization. *Signal Process.*, 46:127–146, 1995.

N B Priyantah, H Balakrishnana, E D Demaine, and S Teller. Mobile-assisted localization in wireless sensor networks. *IEEE INFOCOM*, 1:172–183, 2005.

R Reemsten and S Górner. *Numerical methods for semi-infinite programming: A survey, in* Semi-Infinite Programming. Kluwer Academic, Boston, 1998.

J G Ryan and R A Goubran. Array optimization applied in the near field of a microphone array. *IEEE Transactions on Speech Audio Processing*, 8(2): 173–178, 2000.

A Savvides, C C Han, and M B Strivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. *MobiCom*, pages 166–179, 2001.

L C Shiu. The robot deployment scheme for wireless sensor networks. *International Journal of Innovative Computing, Information and Control*, 6:2941–2953, 2010.

J Smith and J Abel. Closed-form least-squares source location estimation from range-difference measurements. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 35(12):1661–1669, 1987.

A M C So and Y Ye. A semidefinite programming approach to tensegrity theory and realizability of graphs. *Symposium on Discrete Algorithms*, pages 766–775, 2006.

A M C So and Y Ye. Theory of semidefinite programming for sensor network localization. *Mathematical Programming*, 109(2-3):367–384, 2007.

G Still. Discretization in semi-infinite programming: The rate of convergence. *Math. Program.*, 91:53–69, 2001.

J F Sturm. Using sedumi 1.02, a matlab toolbox for optimization over symmertric cones. *Department of Econometrics, Tilburg University*, 1998.

K L Teo, X Q Yang, and L S Jennings. Computational discretization algorithms for functional inequality constrained optimization. *Ann. Oper. Res.*, 28:215–234, 2000.

R Tichatschke and V Nebeling. A cutting plane method for quadratic semi-infinite programming problems. *Optimization*, 19:803–817, 1988.

K C Toh, R H T*ütüncü*, and M J Todd. On the implementation and usage of sdpt3-a matlab software package for semidefinite-quadratic-linear programming, version 4.0. 2006.

H L Van Trees. *Optimum Array Processing*. Wiley, NY, 2002.

P Tseng. Second-order cone programming relaxation of sensor network localization. *SIAM Journal on Optimization*, 18(1):156–185, 2007.

B V Veen and K Buckley. Beamforming: A versatile approach to spatial filtering. *IEEE ASSP Magazine*, 5(2):4–24, 1988.

F Wang, V Balakrishnan, Y Zhou, J Chen, R Yang, and C Frank. Optimal array pattern synthesis using semidefinite programming. *IEEE Transacions on Signal Processing*, 51(5), 2003.

Z Wang, S Zheng, S Boyd, and Y Ye. Further felaxations of the SDP approach to sensor network localization. *SIAM J. Optimization*, 19(2):655–673, 2008a.

Z Wang, S Zheng, Y Ye, and S Boyd. Further relaxations of the semidefinite programming approach to sensor network localization. *SIAM Journal on Optimization*, 19:655–673, 2008b.

H Wolkowicz, R Saigal, and L Vandenberghe. *Handbook of Semidefinite Programming, theory, algorithms and applications*. Kluwer Academic, 2003.

S Wu and S Boyd. sdpsol: A parser/solver for semidefinite programs with matrix structure. *Information systems laboratory, Stanford University*, 1996.

K Yang, G Wang, and Z Q Luo. Efficient convex relaxation methods for robust target localization by a sensor network using time differences of arrivals. *IEEE Transactions on Signal Processing*, 57(7):2775–2784, 2009a.

K Yang, J An, X Bu, and G Sun. Constrained total least-squares location algorithm using time-difference-of-arrival measurements. *IEEE Transactions on Vehicular Technology*, 59(3):1558–1562, 2010.

Z Yang, Y Liu, and X Y Li. Beyond trilateration: On the localizability of wireless ad-hoc networks. *IEEE INFOCOM*, pages 2392–2400, 2009b.

K F C Yiu, X Q Yang, S Nordholm, and K L Teo. Near-field broadband beamformer design via multidimensional semi-infinite linear programming techniques. *IEEE Transactions on Speech and Audio Processing*, 11(6), 2003.

K F C Yiu, N Grbić, S Nordholm, and K L Teo. Multicriteria design of oversampled uniform dft filter banks. *IEEE Signal Processing Letters*, 11(6):541–544, 2004.

Y Yu, K Teo, Y Lim, and G Zhao. Extrapolated impulse response filter and its application in the synthesis of digital filters. *Signal Processing*, 85(3):581–590, 2005.

L P Zhang, S Y Wu, and M A López. New exchange method for convex semi-infinite programming. *Siam J. Optim.*, 20:2959–2977, 2010a.

Z Zhang, K I, T Miyake, and T Imamura. Three-dimension sound localization by binaural model using self organizing map. *International Journal of Innovative Computing, Information and Control*, 6:361–371, 2010b.

Z Zhu, A M C So, and Y Ye. Universal rigidity: Towards accurate and efficient localization of wireless networks. *IEEE INFOCOM*, 2010.