Localization Algorithms for Conference Systems

BSc Thesis

Niels de Koeijer & Joris Bentvelsen

Delft University of Technology

TU Delft

Localization Algorithms for Conference Systems

BSc Thesis

by

Niels de Koeijer & Joris Bentvelsen

Student number:	4480767 (Niels de Koeijer)	
Student number:	4460642 (Joris Bentvelsen)	
Thesis committee:	Dr. ir. R. Heusdens,	TU Delft, supervisor
	Dr. ir. R. C. Hendriks	TU Delft, supervisor
	Dr. J. Martinez,	TU Delft, supervisor
	Dr. ir. J. J. A. Baselmans	TU Delft
	Dr. Carmen G. Almudever	TU Delft

An electronic version of this thesis is available at http://repository.tudelft.nl/.

TU Delft

Abstract

This report details the design and implementation of a subsystem within a localization system designed for the Bosch DICENTIS wireless conference system. The goal of this localization system is to determine the location of individual units. These units each contain a microphone and a loudspeaker and are used by the participants at a conference. Potential use cases for the locations of these units are beam-forming and the mapping of participants within a room.

The subsystem discussed in this report is tasked with transforming estimated propagation times of audio signals into locations of the units. To this end, a TOA-based and a TDOA-based algorithm for the self-localization of individual conference system units is presented. Experimental results show that both methods are able to recover positions, with the TOA-based method slightly outperforming the TDOA-based method. Real-life measurement results show a root mean square error in position of about 8 cm for the TOA-based method.

Preface

This thesis is written in context of the Bachelor Graduation Project. The project was commissioned by the company Bosch with the goal to design a localization system for their DICENTIS Conference System. Initially, the goal was to implement the designed system on a DICENTIS System lent to us by Bosch. However, due to the difficulties in altering the software on of the DICENTIS System, it was decided to construct a prototype which could potentially be implemented instead.

We would like to express our gratitude to our daily supervisor dr. Jorge Martinez and our supervisors dr.ir. Richard Heusdens and dr.ir. Richard Hendriks for their guidance during the project. Furthermore, we would like to thank our contact at Bosch, Hans van der Schaar for lending us a DICENTIS Conference System and taking the time to receive us at Bosch. Finally we would like to thank our colleagues: Tim Ammerlaan, Tim Al, Jelle Tams and Nuriel Rozsa for an enjoyable and productive collaboration.

Niels de Koeijer & Joris Bentvelsen
Delft, June 2018

Contents

1
Introduction

Conference systems

Conference Systems are used in various settings. From large conferences at the United Nations to smaller conferences within companies, these systems have eased conversations consisting of large groups of people. A conference system consists of a number of units with a speaker and a microphone. Usually every attendee of the conference has his own unit. When an attendee talks through his microphone his voice is sent to all other units and played back through their respective speakers. Knowledge of the location of each individual attendee is useful. For example, such knowledge can aid the video recording of conferences. It is often desirable to record the person who is speaking. In heated debates, the person who is speaking can change rapidly. In these circumstances, it is difficult for a human to direct the camera. Knowledge of the location of attendees can be used to automate this process. This can be done by directing the camera automatically to whoever is speaking.

Large conferences are often held in rooms dedicated for holding conferences. In those rooms wired units are used which are often fixed in the same location for each conference. Nowadays, the localization of the units is solved by determining the locations of the units by hand. The process involved is often time intensive. However, this is not an issue for dedicated conference rooms, as the location of each conference system unit only has to be measured up once.

However, not all conferences take place in dedicated rooms. Sometimes conferences are held in locations where no wired system can be fixed. For such locations a wireless conference system is used where the units are placed whenever there is a need for a conference. In that case, locations of each individual conference system have to be redetermined every time, which is impractical.

Figure 1.1: A conference system unit from the DICENTIS line that is used during this project

Bosch creates portable conference system units that are used for these conferences like these. Bosch would like to solve the existing impracticality of measuring by hand by integrating a localization system into their DICENTIS line of wireless conference systems. These conference systems consist of a number wireless conference system units and a central hub. This central hub connects all the individual conference system units by relaying the audio recordings from one unit to the other units. In Figure 1.1 a conference system unit from the DICENTIS line that was used in this project can be seen. In the ideal case, all conference system units can be localized with just the press of a button. Designing a localization system for such a conference system is the topic of the Bachelor Final Project that this thesis is a part of.

State of the art analysis

Localization is an active field of research. As chips are getting cheaper and the power usage keeps declining wireless sensors are being deployed in large numbers. This is because they deliver very interesting opportunities: for example, the monitoring of buildings and environments [1]. In many cases the determination of the actual location of those sensors is very valuable or even required. It is predicted that in 10 years the number of sensors per person developed countries will exceed one thousand [2]. As such, it becomes impractical if not unfeasible to perform the localization of these sensors by hand. A well known localization method is the global positioning system (GPS). However, GPS sensors are relatively expensive, have a high power consumption and obtain poor accuracy indoors [3].

For these reasons, a lot of research has been done to find different methods to localize sensors. Methods using different media such as radio frequent and audio have been developed to perform the localization [4]. These different media can be used in different ways. It is possible to perform localization when only the received signal strength (RSS) is available by estimating distances using path-loss models [5]. This determination of distances is called ‘ranging’. Another method of ranging is creating events from which emission times and arrival times between a pair of nodes can be determined. When both the emission and arrival times are used to localize, the algorithm is named a Time of Arrival (TOA) algorithm. Many algorithms have been developed to find the locations of the nodes using this kind of information such as the ones in [6] and [7]. In case only the arrival times are used the algorithm falls under the Time Difference Of Arrival (TDOA) techniques. Those algorithms use the differences in arrival time to determine locations [8]. These algorithms are particularly useful in case it is hard to obtain a precise estimation of the time of emissions. An other technique to localize objects is fingerprinting. With this technique in advance samples of the received signal strength (RSS) are taken at many places in a certain environment. The resulting fingerprint can then be used to localize an object which is again placed in the fingerprinted area. Unlike most ranging techniques like TOA and TDOA this technique is not as sensitive to an absence of a line of sight [9]. However, an obvious disadvantage of this technique is the requirement of fingerprinting an environment in advance.

In [10] a method is described which can localize the locations of smartphones quite accurately. However, the algorithm used makes the assumption the speaker and microphone are colocated. For a smartphone this is almost the case, however this simplification cannot be made for the system of Bosch where the microphone are considerably far apart.

Subdivision of the System

From the previous section is follows that localization can be done in many different ways.

For this project it is decided that time-based ranging techniques are used. These ranging techniques will automate the initially tedious measurement process that is done by hand. The first step in finding distances through time-based ranging is the accurate determination of the arrival times. This is not a trivial task due to different sources of disturbances such as environmental noise and reflections. Obtaining accurate time measurements in spite of these challenges is the task of the first subsystem: ‘Ranging’.

Then there are some issues with determining distances directly from the send and receive times. First of all, conference system units can be wireless. As a result, their system clocks may not be completely synchronized. This results in a possible difference in the local time each unit. This introduces timing errors. In addition to this, there are also a number of other delays in the path between transmitter and receiver. For example, there is an offset between having claimed to send and actually sending a signal. It is clear that, to obtain more accurate distance measurements from send and receive times, another stage of processing must be done to the timing information provided by the ”Ranging” step. This is the second stage of the localization process, and is called ‘Delay Estimation’.

Once the timing information has been refined, the distances can be determined using the propagation speed of the medium. These distances are then used to find the coordinates of the conference system units. This is the third and last stage of the localization process: the ‘Localization Algorithm’. The localization system is subdivided into these three stages. An overview of the stages and their respective inputs and outputs are given by Figure 1.2. This thesis will describe this last stage in the localization process.

Document Structure

The main topic of this thesis is the design and implementation of the localization algorithm for the specific case of localizing conference system units. Chapter 2 specifies the program of requirements for the whole localization

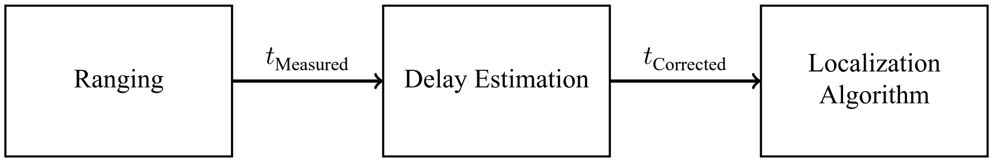

Figure 1.2: Schematic overview of the different stages in the localization process. The last stage, 'Localization Algorithm' will output the coordinates of all the conference system units

system. From these general requirements the requirements for the specific localization algorithm stage are further elaborated on in chapter 3. In chapter 4 a set of promising existing techniques are judged and compared based on the requirements described in chapter 3. From these two existing techniques algorithms are selected. These two algorithms are described in more detail in chapter 5. Their performance is compared in chapter 6. Finally, conclusions and recommendations are given in chapter 7.

2
Program of Requirements

This is the program of requirements that holds for the complete system. The goal of this program of requirements is to specify the features of the complete localization system that is to be developed. This is done by listing the assumptions and requirements of the complete system.

Assumptions

When discussing the system requirements with the contact person at Bosch, there were a set of assumptions that were allowed to be made. The made assumptions for the overarching project are as follows:

1. As indicated by the representative at Bosch when discussing the system requirements, there is line of sight between the Bosch DICENTIS units and between any given unit and the access point.
2. Minimal distance between units is 75 cm.
3. Microphones on the Bosch DICENTIS units can accurately capture audio from a distance of at least 30 m.
4. There is at least one unit per 15 m^2.

Mandatory requirements

These are criteria of which the system should always, at the very least, comply with. These can be subdivided into functional and non-functional requirements. Functional requirements being requirements of what the system must do, and non-functional requirements being attributes that the system must have.

1. **Non-Functional requirements**
   (a) The localization system must be within 10 cm accurate.
   (b) The localization system must be scalable up to 120 units.
   (c) A 3D localization method is necessary.
   (d) The localization should work in conference rooms with dimensions up to 30×30 m.
2. **Functional requirements**
   (a) The localization speed should have a maximum duration of 15 minutes for systems of more than 100 units.
   (b) The localization speed should have a maximum duration of 5 minutes for systems of less than 20 units.
   (c) The localization algorithm has to comply with the current Bosch DICENTIS conference system hardware characteristics.
   (d) The localization procedure can only be initiated remote interface, so no one accidentally starts it during a conference.
   (e) The system as a whole should not exceed the maximum allowed sound pressure level.
   (f) The addition to the system of Bosch should not bypass any safety precautions taken by Bosch.

Trade-off requirements

These are criteria of which it is preferable to comply with as much as possible:

1. Minimize the number of manually configured anchor points.
2. Minimize the number of units required for the system to work.
3. The localization should work with the least possible additional hardware.
4. The localization speed should be as fast as possible.
5. The localization should be as accurate as possible.
6. A purely 2D localization method is usable besides a 3D localization method.

3
Design Criteria for Localization Algorithms

In this chapter the main criteria for designing the localization algorithms are discussed. These are based on the Program of Requirements, design choices for other subsystems, and limitations of the Bosch DICENTIS Conference System. These criteria are then summarized in a program of requirements for the subsystem.

3.1. Design Limitations for Localization Algorithm Subsystem

Non Co-location of Receiver and Transmitter

In this project, the signals sent between units are transmitted via audio. As a result, the transmitter and receiver are the units loudspeaker and the microphone. The loudspeaker and microphone can be up to 30 cm apart. That is to say, they are not co-located. This is an important criteria, as many algorithms are designed for co-located transmitter and receiver and thus cannot be used. In principle, the units also have an on board WIFI module, so Radio Frequent (RF) signals could also have been used. In this case transmitter and receiver are co-located. However, the DICENTIS units cannot communicate directly with one another: all WIFI communication must be done through the central hub. This is impractical for many measurement techniques (TOA, TDOA, AOA). This leaves audio as the only viable signal type, and non co-location must be taken into account.

Measurement Technique

The type of algorithm used is strongly dependent on the type of measurements used. This raises the question: what type of measurement technique is best? In chapter 1 these measurements were assumed to be time based distance measurements. However, this does not necessarily have to be the case. For example, instead of using the measured distance between nodes, it is also possible to use the angle of arrival (AOA) of signals to determine location [11]. There are also different methods of obtaining distance measurements. For example, instead of using time, the signal power of the received signals (RSS) can be used to obtain a distance estimate between units. This can be done by using path loss models which relate loss of signal power to distance in certain environments [12]. However, from the data sheet [13], it can be seen that the conference microphones used with the units are strongly directional. As a result, the incoming received signal power will depend strongly on the orientation of the units. This makes AOA and RSS based measurement methods impractical. The choice is to use time based distance measurement techniques time of arrival (TOA) and time difference of arrival (TDOA) for this project.

Minimum Information Required

Some algorithms require less information to solve the localization problem than others. This can be measured in two different ways.

- **Lower bound on amount of receivers and transmitters**: This is the minimum amount of receivers and/or transmitters required for an algorithm to function. This lower bound can be met by using more DICENTIS units (each unit adds both a receiver and a transmitter) or by adding additional separate receivers and transmitters. It is desirable to require no additional hardware and for the system to work with minimal amounts of units, therefor it is advantageous to keep this bound as low as possible.

- **Required number of anchor nodes**: Anchor nodes are nodes in a network of which the position is known. In the case of the DICENTIS units, these would be loudspeakers or microphones of which the postion is known before starting localization. Algorithms can be subdivided into two classes with respect to anchors: anchor-based and anchor-less algorithms [14]. Anchor-based algorithms use a number of known anchor nodes to estimate the other nodes in a network. These algorithms typically become better when more anchor nodes are used, and require a minimum amount of anchors to work. Anchor-less algorithms do not require any anchor nodes. As a result however, the coordinates that follow from an anchor-less algorithm are only defined up to random translation and rotation. To relate these coordinates to a known coordinate system, Procrustes analysis can be used to find the appropriate translation and rotation [15]. To do this, at least 3 anchors must be known for 2 dimensional localization problems, and at least 4 anchors must be known for 3 dimensional case.

  In the case of the DICENTIS units, it is desirable for there to be as little as possible anchor nodes required, as these would likely have to be measured by hand.

Scalability and Computational intensity
From the requirements given in chapter 2 it can be seen that the localization system should be scalable up 120 units with a maximum duration of 15 minutes. For this reason, it is important to consider how the time duration of the algorithm scales with the number units. This run time can be related to the ‘Big O Notation’ [16].

Accuracy in Presence of Noise
In real world situations, measurement noise will be present. For algorithms this noise is typically normally distributed with zero mean and standard deviation σ. To determine an algorithms robustness in the presence of noise, simulations can be performed. In the early stages of this project it was determined experimentally that a standard deviation of 3cm for the distance measurements would be realistic. This models a 95.45% chance of measurement errors being smaller than 6cm, which is within the requirement of accuracy specified by the program of requirements.

3.2. Program of Requirements Localization Algorithm Subsystem
Based on the design limitations derived in the previous section, the following program of requirements specific to the algorithm subsystem can be compiled.

Assumptions

- The errors in the distance measurements have a standard deviation of at most 3 cm.
- There are no missing links in the system, that is, there are distance measurements between all nodes in the network.

Mandatory requirements

1. **Functional requirements**
   - The localization algorithm must use time-based measurement techniques TOA or TDOA.
   - The localization algorithm must provide coordinates for both the three dimensional as well as the two dimensional case.
2. **Non-Functional requirements**
   - The maximum localization error should be below 10cm, also in presence of noise as specified in ‘Assumptions’.
   - The algorithm should be scalable up to 120 units.
   - The calculation time of the algorithm should be shorter than 5 minutes for a system of more than 100 units and shorter than 1 minute for a system of less than 20 units.

Trade-Off requirements

3. **Non Functional requirements**
   - The localization algorithm should work with as little anchors as possible.

- The set-up time should be as easy and fast as possible.
- The localization algorithm should require as little units as possible.
- The localization algorithm should require as little additional hardware (hardware other than DICENTIS units) as possible.
- The localization algorithm should be as accurate as possible and as resistant as possible against noise.
- The run time of the localization algorithm should be as fast as possible.

Key Performance Metrics

- Root Mean Square Error (RMSE)
- Algorithm run-time

4

Analysis of existing techniques

In the previous chapters the the situation surrounding the problem is elaborated. Also an interpretation is made of what the general program of requirements means for the requirements on the algorithm designed in this thesis. In this chapter an analysis of existing techniques is performed to decide which techniques are most promising for the localization of the conference systems.

The design criteria given in the previous chapter strongly narrow the search for a good algorithm. As stated in the design criteria the localization algorithm should have time measurements as inputs. When both the time of emission and time of arrival are known, the distance from a source to a microphone can be calculated directly. In case an algorithm uses this direct distance information it falls under the category of 'Time Of Arrival' (TOA) algorithms. In case the emission times are not available the distance from the source to microphone is not known. In this case for an sound event it is only possible to calculate the difference in arrival times to all microphones. An algorithm which uses these differences falls under the category of Time Difference Of Arrival (TDOA) algorithms [17]. There are several reasons why it may not be possible to know the emission times. For the BOSCH Dicentis Units, these are difficult to obtain unless the software running on the units is altered. In such cases there is no choice but using an TDOA algorithm. However, even when estimations of the emission times can be made a TDOA algorithm can have an advantage over a TOA algorithm. This is because for TOA precise synchronization between sources and microphones is required. Also the onset time, the time for the generation of a signal, have to be taken into account for TOA methods [8]. However, TDOA also has a disadvantage to TOA. When the sensors are very close, the time differences will also become very small. In that case the noise becomes relatively large [18] and TDOA becomes inaccurate.

Assuming co-location in a non co-located situation introduces large errors. This narrows down the list of possible algorithms. For example, the algorithm described in [19] simultaneously synchronizes and localizes units cannot be used due to the co-location assumption. Being limited to only time-based measurement also narrows down the list. A use number of algorithms use multiple techniques to solve the localization problem, the so called hybrid algorithms [20]. The algorithm described in [21] for example uses RSS to detect for lack of line of sight in TOA measurements.

Finally, 4 algorithms are found to be promising. These are analyzed in more detail below using the criteria given in chapter 3 to judge which one is best fit to the problem at hand.

4.1. TDOA Algorithms

Anchorless TDOA

A TDOA algorithm that suits the subsystems criteria well is one based on a paper by Pollefeys [22]. This algorithm can work without anchors and does not assume co-location of receivers and transmitters. Both receivers and transmitters are estimated. The algorithm works by first estimating the emission times of the sources. This is this algorithms main strength, as it requires only arrival times to fully solve the localization problem. The emission time estimation part of the algorithm requires at least 5 sources and 10 microphones to work. These emission times are then used to estimate distances between sources and microphones, similarly to TOA algorithms. In the algorithm an least squares problem is set up and then a number of steps are performed to simplify this least squares problem. Finally this results in a closed form solution, rather than an iterative solution. This has the advantage of lower computational complexity than if it were an iterative solution.

Simulations of this algorithm show that it is relatively inaccurate in presence of noise. The process of estimating emission times works well in the noise free case, but errors are present when noise is included.

TDOA Fusion

An other TDOA algorithm is described in [23]. At least three or four anchors anchors are required in case localization is performed in respectively 2 or 3 dimensions. This means the algorithm is anchor-based. As mentioned, this comes with the disadvantage that the anchors must be measured beforehand. This algorithm can localize either microphones or sources: not both however. The anchors can be microphones to localize sources, or sources to localize microphones. Nodes are localized completely independently from each other. This has two advantages. First of all, there is no minimum number of nodes required for successful localization: only a minimum number of anchors. Secondly, this means that the algorithm scales linearly with the number of to be estimated nodes. This means very low computational complexity. In addition to this, the problem is formulated in such a way that a closed form solution is obtained.

This algorithm is relatively accurate, even in presence of measurement noise. Its performance scales with the number of anchors used.

4.2. TOA Algorithms

Multidimensional Scaling

A well known algorithm is Multidimensional Scaling (MDS). It is an algorithm used for many different applications. Given the complete set of pairwise distances between nodes, MDS tries to assign relative coordinates to all nodes such that the locations fit the data the best. This algorithm is not only used for localization problems but also for visualization of all kind of experimental results [24]. MDS scales with the cube of the to be located nodes. In principe, it is an anchor-less algorithm. It requires atleast 3 nodes for 2 dimensions or 4 nodes for 3 dimensions [25].

The main problem with MDS for our system is that it requires pairwise distances between all nodes. However, the distance measurements can only be performed from speakers to microphones and not from microphone to microphone or from speaker to speaker. Consequently, pairwise distances can not directly be obtained. This is due to the loudspeaker and microphone not being co-located: there will be two different distances between two DICENTIS units.

If the average of these two distances is computed a kind-of pairwise distance is obtained. This setup was simulated for different co-location offsets, estimating the real distances between microphone and loudspeaker. The co-location offset is defined as the distance from the source to the microphone of an unit. It turns out that for very small offsets this method does still give accurate results, but for distances over 5cm soon the localization errors became very large. This confirms that the non co-location of the situation introduces large errors in algorithms that assume co-location.

A bilinear method

An other interesting algorithm is the algorithm described in [6]. The algorithm is able to localize individual sources and microphones without knowledge of the position of any of them, similar to the algorithm by Pollefeys. This makes it an anchor-less algorithm. First, a closed form solution is found. This closed form solution has as a requirement that at least one microphone is co-located with one source. All other measurements do not have to be co-located. This closed form solution is then used as a starting point for further refinement using an iterative technique gradient descent. Being an iterative algorithm, it is typically relatively computationally complex. The algorithm requires at least four sources and four microphones to work.

Tests of the algorithm show high accuracy. This is likely due to the refinement of closed form solution using gradient descent.

4.3. Final Selection of Algorithms

To get an better view which algorithms performs best all are implemented and tested in a basic situation. A comparison of all algorithms can be seen in Table 4.1. From this table it immediately becomes clear that the Pollefeys TDOA algorithm and the MDS TOA algorithm have clear disadvantages. Both are very inaccurate. In addition to this, the Pollefeys TDOA algorithm requires a large amount of receivers to work, which is undesirable.

This leaves the TOA algorithm based on Crocco and the TDOA-Fusion algorithms as main contenders. The

Table 4.1: Algorithm comparison based on the criteria given in chapter 3

Algorithm	Measurement	Colocation Assumed	Minimum for 3D		Anchor Taxonomy	Accuracy	Computational Complexity
			Receivers	Sources			
Pollefeys	TDOA	No	10	5	Anchor-less	-	+
TDOA-Fusion	TDOA	No	4	1	Anchor-based	+	++
Crocco	TOA	No	4	4	Anchor-less	++	–
Averaged-MDS	TOA	Yes	4	4	Anchor-less	–	-

TOA algorithm has a high accuracy at cost of high computational complexity. For the TDOA algorithm, the computational complexity is quite low while the accuracy is still decent. The main disadvantage of the TDOA algorithm is that it is anchor-based and thus always requires the anchors to be measured up before the localization can be started. In addition to this, the TDOA algorithm only localizes either sources or microphones, not both. Crocco is typically more accurate, but as was mentioned, TDOA is less prone to measurement offsets than TOA.

As neither algorithm is clearly better than the other, both algorithms will be implemented fully. The design of both algorithms are discussed in chapter 5. Their implementations are tested through simulations and a real world tests in chapter 6 to get an indication of which algorithm is better.

5

Design of Algorithms

This chapter details the design of the TOA and TDOA algorithms. The algorithms are based on the papers by Crocco [6] and Sayed [23] respectively. First the notation and definitions used in the explanation of both algorithm are introduced.

5.1. Notation and Definitions

This section defines the notation is used for the theory behind the TOA and TDOA algorithms. The dimension $\mathcal{D}$ of the localization problem can be 2D or 3D. The notation introduced in this section hold for both, but examples are given in 3D coordinates.

Consider N receivers (microphones) in $\mathcal{D}$-dimensional space. The coordinates of the location of receiver i are denoted in Cartesian coordinates with a column vector as $\mathbf{r}_i = (x_i, y_i, z_i)^{\mathrm{T}}$. The receiver location matrix $\mathbf{R}$ is of size $(N \times \mathcal{D})$ and is defined as:

$$\mathbf{R} = \begin{bmatrix} \mathbf{r}_1{}^{\mathrm{T}} \\ \mathbf{r}_2{}^{\mathrm{T}} \\ \vdots \\ \mathbf{r}_N{}^{\mathrm{T}} \end{bmatrix} \tag{5.1}$$

In similar fashion, consider M acoustic sources in $\mathcal{D}$-dimensional space. Acoustic sources can be sound made by a speaker, or even hand claps. The coordinates of the location of acoustic source j are denoted by $\mathbf{s}_j = (a_j, b_j, c_j)^{\mathrm{T}}$. The acoustic source coordinate matrix $\mathbf{S}$ is of size $(M \times \mathcal{D})$ and is defined as:

$$\mathbf{S} = \begin{bmatrix} \mathbf{s}_1{}^{\mathrm{T}} \\ \mathbf{s}_2{}^{\mathrm{T}} \\ \vdots \\ \mathbf{s}_M{}^{\mathrm{T}} \end{bmatrix} \tag{5.2}$$

Time of flight (TOF) is the time that it takes a signal from source j to reach receiver i. To measure TOF, it is required to know the emission time of the signal from the source and the arrival time of the signal at the receiver. Typically, this is done through TOA measurements. However, the term TOF is used as TOA measurements can contain measurement offsets such as onset times and internal delays. TOF assumes these delays to not be present. The estimated distance d_{ij} between receiver i and acoustic source j can be determined based on measured TOF. A measured TOF t_{ij} between receiver i and acoustic source j likely contain a measurement errors. This measurement error can modelled using a normally distributed random variable X with zero mean and variance σ^2. The measured TOF t_{ij} can then be expressed as:

$$t_{ij} = c^{-1}\|\mathbf{r}_i - \mathbf{s}_j\| + n_{ij} \tag{5.3}$$

In this equation, c represents the signal propagation velocity and n_{ij} is a realisation of the measurement noise random variable $X \sim \mathcal{N}(0,\, \sigma^2)$. From this TOF measurement an estimated distance d_{ij} is expressed as:

$$d_{ij} = c \cdot t_{ij} \tag{5.4}$$

These estimated distances are put in the estimated distances matrix $\mathbf{D}$ where $\mathbf{D}_{ij} = d_{ij}$ for $1 \leq i \leq N$ and $1 \leq j \leq M$. With this the distances matrix $\mathbf{D}$ contains the distances between all receiver-source pairs.

To measure TOF, the emission times for all the sources are required. In some cases, emission times are difficult to determine or unknown. In this case, Time Difference of Arrival (TDOA) measurements can be used instead. For TDOA measurements, the difference in arrival time of a signal sent by a source j between a receiver i and a reference receiver k is calculated. This TDOA measurement is expressed using the notation Δt_{ik}. The TDOA measurement Δt_{ik} for a source j is expressed as the difference between two TOF measurements:

$$\begin{aligned} \Delta t_{ik} &= t_{ij} - t_{kj} \\ &= c^{-1}\left(d_{ij} - d_{kj}\right) \end{aligned} \tag{5.5}$$

Again assuming N receivers and M sources. The equation above holds for $1 \leq i \leq N$ for a source j where $1 \leq j \leq M$. One reference receiver k can be chosen from the any of the N receivers. This reference receiver k functions as a time reference for all TDOA measurements. This is to ensure that all TDOA measurements are in reference to the same time scale. The TDOA measurement can be related to a distance Δd_{ik} by multiplying with the propagation velocity c:

$$\Delta d_{ik} = c\Delta t_{ik} \tag{5.6}$$

This distance can then be written in terms of TOF measurements by using (5.5):

$$\Delta d_{ik} = d_{ij} - d_{kj} \tag{5.7}$$

(5.7) gives an interpretation of the distance Δd_{ik}. Namely the difference in distances of receiver $\mathbf{r}_i$ and reference receiver $\mathbf{r}_k$ to a common source $\mathbf{s}_j$.

5.2. TDOA-Fusion Algorithm

Using the definitions defined in the previous section, an algorithm to find source coordinate matrix $\mathbf{S}$ is derived. This algorithm is given in general dimension $\mathcal{D}$, so it can be easily adapted for 2D and 3D. This algorithm requires complete or partial knowledge of the receiver coordinates $\mathbf{R}$ to find source coordinates $\mathbf{S}$. In principle, this can be done the other way around as well: $\mathbf{S}$ can be used to find $\mathbf{R}$. The derivation is given for the former case however.

The algorithm works as follows. Each source $\mathbf{s}_j$ is determined independently of one another, using the difference in arrival time of its signal to all N microphones. As mentioned in section 5.1, TDOA measurements are defined relative to a common reference receiver k. This k can be chosen out of any of the N known receivers. After a k is chosen, the resulting TDOA measurements are enough to localize a source $\mathbf{s}_j$. This is done by rewriting the equations defined in section 5.1 to a specific matrix equation, from which a closed form solution for $\mathbf{s}_j$ can be found.

However, the resulting $\mathbf{s}_j$ will differ based off of the chosen reference receiver k. The reason for this is that for one reference k only N different TDOA measurements Δd_{ik} are used, as for i it holds that $1 \leq i \leq N$. For different choices of reference receiver k different sets of N TDOA measurements are used. The TDOA fusion algorithm exploits this fact by performing the estimation of $\mathbf{s}_j$ for each possible choice of reference receiver k. This results in N different estimations of $\mathbf{s}_j$ that can then be averaged to find a final estimation of $\mathbf{s}_j$. This result should be more accurate, because more information is used to determine $\mathbf{s}_j$.

Now that the general idea of the TDOA fusion algorithm is clear, the algorithm is derived.

Finding a Closed Form Solution

The TDOA-Fusion algorithm will now be explained in more detail. The first step in this process is finding a closed form solution for $\mathbf{s}_j$. The first step in doing so is rewriting (5.7) from section 5.1 and squaring it:

$$\left(\Delta d_{ik} + d_{kj}\right)^2 = d_{ij}^2 \tag{5.8}$$

By expressing d_{ij} and d_{kj} in the expression above in terms of euclidean distance using (5.4) and (5.3), expanding terms, and rearranging, the following expression is obtained:

$$(\mathbf{r}_i - \mathbf{r}_k) \cdot \mathbf{s}_j = -d_{kj}\Delta d_{ik} + \frac{1}{2}\left(\|\mathbf{r}_i\|^2 - \|\mathbf{r}_k\|^2 - \Delta d_{ik}^2\right) \tag{5.9}$$

To see exactly how this expression is obtained, please consult appendix A.1. (5.8) can now be expressed as a matrix equation. (5.9) describes a system of equations that holds for $1 \leq i \leq N$ for a chosen reference receiver k and source j. As mentioned before, the goal is to find $\mathbf{s}_j$ for a specific reference receiver k. As such, the matrix equation will be setup in such a way that a closed form solution for $\mathbf{s}_j$ for a specific reference receiver k can be found.

To do this, a number of matrices and vectors must be defined first. First, define matrix $\mathbf{A}_k$ which contains the first term in (5.9).

$$\mathbf{A}_k = \begin{bmatrix} (\mathbf{r}_1 - \mathbf{r}_k)^\mathrm{T} \\ \vdots \\ (\mathbf{r}_{k-1} - \mathbf{r}_k)^\mathrm{T} \\ (\mathbf{r}_{k+1} - \mathbf{r}_k)^\mathrm{T} \\ \vdots \\ (\mathbf{r}_N - \mathbf{r}_k)^\mathrm{T} \end{bmatrix} \tag{5.10}$$

Note that this matrix is relative to a specific $\mathbf{r}_k$: each choice of k corresponds to a different A_k matrix. Lastly, note that the row for which $i = k$ is omitted in matrix. This is to prevent a redundant zero row in the matrix.

Next, vector $\mathbf{b}_k$ is defined. This vector contains the TDOA distance measurement contained in (5.9), Δd_{ik}. Similarly to the A_k matrix, the row for which $i = k$ is omitted.

$$\mathbf{b}_k = \begin{bmatrix} -\Delta d_{1k} \\ \vdots \\ -\Delta d_{k-1k} \\ -\Delta d_{k+1k} \\ \vdots \\ -\Delta d_{Nk} \end{bmatrix} \tag{5.11}$$

Lastly, define the vector $\mathbf{c}_k$. This vector contains the final term in (5.9).

$$\mathbf{c}_k = \frac{1}{2} \begin{bmatrix} \|\mathbf{r}_1\|^2 - \|\mathbf{r}_k\|^2 - \Delta d_{1k}^2 \\ \vdots \\ \|\mathbf{r}_{k-1}\|^2 - \|\mathbf{r}_k\|^2 - \Delta d_{k-1k}^2 \\ \|\mathbf{r}_{k+1}\|^2 - \|\mathbf{r}_k\|^2 - \Delta d_{k+1k}^2 \\ \vdots \\ \|\mathbf{r}_N\|^2 - \|\mathbf{r}_k\|^2 - \Delta d_{Nk}^2 \end{bmatrix} \tag{5.12}$$

These definitions are used to write the system of equations defined by (5.9) as a matrix equation. This is done as follows:

$$\mathbf{A}_k \mathbf{s}_j = d_{kj} \mathbf{b}_k + \mathbf{c}_k \tag{5.13}$$

However, this equation only holds given that there is no measurement noise. When there is measurement noise, it can be recast into a linear least squares problem where an estimate of $\mathbf{s}_j$ is found such that it minimizes the cost function.

$$\arg\min_{\mathbf{s}_j} \left\| \mathbf{A}_k \mathbf{s}_j - (d_{kj} \mathbf{b}_k + \mathbf{c}_k) \right\|^2 \tag{5.14}$$

In the case of a linear least squares problem, an estimate for $\mathbf{s}_j$ is found by rewriting (5.13) by using the pseudo-inverse of matrix $\mathbf{A}_k$. This is necessary, as matrix $\mathbf{A}_k$ is not square. Denoting the pseudo-inverse of matrix $\mathbf{A}_k$ as $\mathbf{A}_k^+$, the least square estimate for $\mathbf{s}_j$ is:

$$\mathbf{s}_j = \mathbf{A}_k^+ \left(d_{kj} \mathbf{b}_k + \mathbf{c}_k \right) \tag{5.15}$$

As was mentioned before, it is assumed that N entries of the receiver position matrix $\mathbf{R}$ are known. In combination with the TDOA distances, this implies that matrices $\mathbf{A}_k$ and vectors $\mathbf{b}_k$ and $\mathbf{c}_k$ are known in (5.15). The vector $\mathbf{s}_j$ and the distance between reference receiver k and source j, d_{kj}, remain unknown. To solve this problem, an additional equation must be added. This is found in the definition of d_{kj}:

$$d_{kj}^2 = \left\| \mathbf{r}_k - \mathbf{s}_j \right\|^2 \tag{5.16}$$

The solution to this problem is found by substituting the least square estimate for $\mathbf{s}_j$ into (5.16) and solving for d_{kj} using the quadratic equation. The result is then substituted into (5.15) to find $\mathbf{s_j}$. This is done in appendix A.2. Rewriting gives the following quadratic form:

$$0 = \alpha d_{kj}^2 + \beta d_{kj} + \gamma \tag{5.17}$$

Where α, β, and γ are:

$$\alpha = \left\|\mathbf{A}_k^+\mathbf{b}_k\right\|^2 - 1 \tag{5.18}$$

$$\beta = 2(\mathbf{A}_k^+\mathbf{b}_k)\cdot(\mathbf{A}_k^+\mathbf{c}_k) - 2\mathbf{r}_k\cdot(\mathbf{A}_k^+\mathbf{b}_k) \tag{5.19}$$

$$\gamma = \left\|\mathbf{r}_k\right\|^2 + \left\|\mathbf{A}_k^+\mathbf{c}_k\right\|^2 - 2\mathbf{r}_k\cdot(\mathbf{A}_k^+\mathbf{c}_k) \tag{5.20}$$

This gives the following solution for d_{kj}:

$$d_{kj} = \frac{-\beta \pm \sqrt{\beta^2 - 4\alpha\gamma}}{2\alpha} \tag{5.21}$$

As this is a solution for the quadratic equation, two solutions are possible. The definition of d_{kj} ensures that it is always positive. As a result, the positive solution of the two is always chosen. This value of d_{kj} is then substituted into (5.15), yielding the closed-form solution $\mathbf{s}_j$ for reference receiver k. Let $\mathbf{s}_{j;k}$ denote the estimate for $\mathbf{s}_j$ corresponding to reference receiver k. The steps described above can now be performed for all N possible choices for reference receiver k. The results can then be averaged to obtain a final estimate for $\mathbf{s}_j$:

$$\mathbf{s}_j = \frac{1}{N}\sum_{m=1}^{N}\mathbf{s}_{j;m} \tag{5.22}$$

This procedure can be followed for all sources to localize an arbitrarily high amount of sources. It is interesting to note that a TDOA Fusion can also be used when TOF measurements are available. In this case the d_{kj} term will be known and does not need to be calculate by solving a quadratic equation.

5.3. TOA Algorithm

This section describes an algorithm which requires knowledge of departure times as well as arrival times. The algorithm is based on the algorithm of Crocco [6]. In contrast to the algorithm described in the previous section, this algorithm can retrieve both the receiver and source locations. This algorithm can work for both 2D and 3D localization problems.

Formulation of the Problem

The goal of the localization problem is to estimate the receiver and source locations contained in matrices $\mathbf{R}$ and $\mathbf{S}$ respectively. This can be done by finding estimations for the locations for receivers and sources such that they are in correspondence with the estimated distance measurements between them. This can be done by solving the following nonlinear least squares problem:

$$\hat{\mathbf{R}}, \hat{\mathbf{S}} = \arg\min_{R,S}\sum_{i=1}^{N}\sum_{j=1}^{M}\left[\left\|\mathbf{r}_i - \mathbf{s}_j\right\| - d_{ij}\right]^2 \tag{5.23}$$

The receiver and source locations are estimated by choosing the receiver and source locations in such a way that the difference between distance between them ($\|\mathbf{r}_i - \mathbf{s}_j\|$) and the estimated distance (d_{ij}) is minimized. The estimated receiver and source location matrices $\hat{\mathbf{R}}$ and $\hat{\mathbf{S}}$ for which the sum of all these differences is as small as possible gives a solution to the localization problem.

However, solving (5.23) directly is difficult due to the presence of many local minima. If the solution of the minimization is in a local minimum rather than in the global minimum, the solution will not be optimal. Nonlinear least squares problems are often solved using iterative algorithms. These algorithms require an initial estimate. Choosing a good initial estimate for $\mathbf{R}$ and $\mathbf{S}$ will reduce the chance of getting stuck in a local minimum.

Another thing to note during the solving of (5.23) is the presence of multiple global minima. The reason for this is that the coordinate system used for the sources and receivers in (5.23) is invariant to translation, rotation and reflection about the 3 axis. This is because only distances between estimated sources and receivers are considered. For example, rotating both receiver $\mathbf{r}_i$ and source $\mathbf{s}_j$ about an axis does not change the distance $\|\mathbf{r}_i - \mathbf{s}_j\|$ between them. This invariance can be resolved by fixing the coordinate system by imposing constraints on the coordinates of a number of receivers and sources. An example of one of the constraints could be fixing the first source in the origin: $\mathbf{s}_1 = (0, 0, 0)$. These constraints can also be applied after finding $\hat{\mathbf{R}}$ and $\hat{\mathbf{S}}$ as every global minimum can be transformed into another one. The rotation, translation and reflection required to transform one global minima to another can be found through Procrustes analysis.

Closed-Form Approach

As is mentioned above, solving (5.23) requires a good initial estimate for $\mathbf{R}$ and $\mathbf{S}$. Crocco [6] describes a method to find such an initial estimate. Currently, the nonlinear least squares problem given by (5.23) has $3(N + M)$ unknowns. This can be reduced to a nonlinear least squares problem with only 9 unknowns. Solving this problem is easier and will provide the initial estimate for solving (5.23). The closed form solution has only one restriction about the measurements: the first source and microphone need to be co-located. If the first event and microphone are not co-located but relatively close to each other, the closed form solution will still give a good initial guess for problem the problem stated in (5.23).

Bilinear Matrix Form

To find this simplified nonlinear least squares problem, consider a situation for which there are no TOF measurement errors. When this is the case, the estimated distances d_{ij} and the distance between receiver $\mathbf{r}_i$ and source $\mathbf{s}_j$ should be equal for all i and j. In this, case the following expression holds for all i and j:

$$\|\mathbf{r}_i - \mathbf{s}_j\|^2 = d_{ij}^2 \tag{5.24}$$

This gives a set of NM equations. The next step to slightly rewrite these equations into so-called bilinear form. The reasoning for this is that this bilinear form of the set of equations given by (5.24) has a useful matrix equation form. When an equation is in bilinear form, it means that it is linear in each of its two argument separately. In the case of (5.24), these two arguments are $\mathbf{r}_i$ and $\mathbf{s}_j$. Currently, (5.24) is not in bilinear form. To see this, the euclidean norm in (5.24) is expanded (for 3 dimensions):

$$x_i^2 + y_i^2 + z_i^2 + a_j^2 + b_j^2 + c_j^2 - 2x_i a_j - 2y_i b_j - 2z_i c_j = d_{ij}^2 \tag{5.25}$$

Clearly, a linear increase in input argument leads to a quadratic increase in the equation above. In this case, the issue is that there are six quadratic terms. To acquire a bilinear form, these must be eliminated. To do this, first, subtracting the $(1, j)^{\text{th}}$ equation of the set given by (5.24) from the $(i, j)^{\text{th}}$ gives:

$$\|\mathbf{r}_i - \mathbf{s}_j\|^2 - \|\mathbf{r}_1 - \mathbf{s}_j\|^2 = d_{ij}^2 - d_{1j}^2 \tag{5.26}$$

This gives a new set of $(N-1)M$ equations. In similar fashion, subtracting the $(i, 1)^{\text{th}}$ equation of the set given by (5.26) from the $(i, j)^{\text{th}}$ gives:

$$\|\mathbf{r}_i - \mathbf{s}_j\|^2 - \|\mathbf{r}_1 - \mathbf{s}_j\|^2 - \|\mathbf{r}_i - \mathbf{s}_1\|^2 + \|\mathbf{r}_1 - \mathbf{s}_1\|^2 = d_{ij}^2 - d_{1j}^2 - d_{i1}^2 + d_{1j}^2 \tag{5.27}$$

Which is a set of $(N-1)(M-1)$ equations with $2 \le i \le N$ and $2 \le j \le M$. This is the desired bilinear form. This see this, the euclidean norms in (5.27) are expanded and the terms are collected. The resulting expression is:

$$-2(\mathbf{r}_i - \mathbf{r}_1) \cdot (\mathbf{s}_j - \mathbf{s}_1) = d_{ij}^2 - d_{1j}^2 - d_{i1}^2 + d_{1j}^2 \tag{5.28}$$

In this case a linear increase in either $\mathbf{r}_i$ or $\mathbf{s}_j$ result in a linear increase in the equation. This confirms that this is indeed the bilinear form. The bilinear set of equations given by (5.28) can be expressed as a matrix equation. To do this a number of new matrices must be defined.

First, the $((N-1) \times \mathcal{D})$ matrix $\tilde{\mathbf{R}}$ can be defined as:

$$\tilde{\mathbf{R}} = \begin{bmatrix} (\mathbf{r}_2 - \mathbf{r}_1)^{\mathrm{T}} \\ (\mathbf{r}_3 - \mathbf{r}_1)^{\mathrm{T}} \\ \vdots \\ (\mathbf{r}_N - \mathbf{r}_1)^{\mathrm{T}} \end{bmatrix} \tag{5.29}$$

In similar fashion, the $((M-1) \times \mathcal{D})$ matrix $\tilde{\mathbf{S}}$ can be defined as:

$$\tilde{\mathbf{R}} = \begin{bmatrix} (\mathbf{s}_2 - \mathbf{s}_1)^{\mathrm{T}} \\ (\mathbf{s}_3 - \mathbf{s}_1)^{\mathrm{T}} \\ \vdots \\ (\mathbf{s}_M - \mathbf{s}_1)^{\mathrm{T}} \end{bmatrix} \tag{5.30}$$

Finally, the entries of the $(N-1 \times M-1)$ adjusted estimated distances matrix $\tilde{\mathbf{D}}$ are defined as:$\tilde{\mathbf{D}}_{i-1,j-1} = d_{ij}^2 - d_{1j}^2 - d_{i1}^2 + d_{1j}^2$ for $2 \leq i \leq N$ and $2 \leq j \leq M$. Using these definitions the bilinear system of equations given by (5.28) are written as:

$$-2\tilde{\mathbf{R}}\tilde{\mathbf{S}}^{\mathrm{T}} = \tilde{\mathbf{D}} \tag{5.31}$$

Rank Approximation via Singular Value Decomposition

Both $\tilde{\mathbf{R}}$ and $\tilde{\mathbf{S}}$ are at most rank $\mathcal{D}$ because they have the amount of columns equal to the localization problem dimension $\mathcal{D}$. For this reason, the matrix product between $\tilde{\mathbf{R}}$ and $\tilde{\mathbf{S}}^{\mathrm{T}}$ as given by (5.31) should be at most rank $\mathcal{D}$ as well. However, when the distance measurements which determine $\tilde{\mathbf{D}}$ are noisy this is probably not the case. As stated by the Eckart–Young–Mirsky theorem a low-rank approximation of a matrix can be found using the singular value decomposition (SVD) [26]. This approximation is found by only considering the first k singular values for a rank k approximation. So by calculating the SVD of $\tilde{\mathbf{D}}$ and only considering the first $\mathcal{D}$ singular values the best rank $\mathcal{D}$ approximation of $\tilde{\mathbf{D}}$ is found. Then the SVD of $\tilde{\mathbf{D}}$ is shown in (5.32).

$$\tilde{\mathbf{D}} = \mathbf{UVW} \tag{5.32}$$

The rank $\mathcal{D}$ approximation is then found by only considering the first $\mathcal{D}$ singular values found on the diagonal of matrix $\mathbf{V}$. This is done by truncating $\mathbf{V}$ to a $(\mathcal{D} \times \mathcal{D})$ matrix which only contains the first $\mathcal{D}$ singular values. After truncation, matrix $\mathbf{U}$ is $((N-1) \times \mathcal{D})$, and matrix $\mathbf{V}$ is $(\mathcal{D} \times (M-1))$.

It is important to note however, that it is assumed that matrices $\tilde{\mathbf{R}}$ and $\tilde{\mathbf{S}}$ are in fact rank $\mathcal{D}$. For this to be the case, they must have at least $\mathcal{D}$ linearly independent rows. This can only be accomplished if there are at least $\mathcal{D}+1$ of each receivers and sources (due to the first source/receiver being subtracted from the rest). In addition to this, they may not be located in the same plane for $\mathcal{D}=3$ or not on the same line for $\mathcal{D}=2$. Otherwise the rows will not be linearly independent of one another. This is the origin for the lower bound for the amount of receivers and sources for this algorithm.

The rank-approximated matrix $\tilde{\mathbf{D}}$ allows $\hat{\mathbf{R}}$ and $\hat{\mathbf{S}}$ to be expressed in terms of the components of the SVD. This is done using a so-called mixing $(\mathcal{D} \times \mathcal{D})$ matrix $\mathbf{C}$ to combine (5.32) and (5.31) as follows:

$$\tilde{\mathbf{R}} = \mathbf{UC} \tag{5.33}$$

$$-2\tilde{\mathbf{S}}^{\mathrm{T}} = \mathbf{C}^{-1}\mathbf{VW} \tag{5.34}$$

As aforementioned, a closed form solution can be found when the first source and the first receiver coincide: $\mathbf{s}_1 = \mathbf{r}_1$. Additionally, due to the invariance of translation and rotation of the solution, this first source and receiver can be freely chosen in the origin. In that case $\mathbf{R}$ and $\mathbf{S}$ can easily be obtained from $\tilde{\mathbf{R}}$ and $\tilde{\mathbf{S}}$ respectively. With source and receiver co-located in the origin, there is still invariance of rotation. This fact can be exploited to rotate matrix $\mathbf{C}$ to an upper triangular matrix. This can be done as follows. Consider (5.33) rotated by post multiplying $\hat{\mathbf{R}}$ by a $(\mathcal{D} \times \mathcal{D})$ rotation matrix $\mathbf{Z}$:

$$\tilde{\mathbf{R}}\mathbf{Z} = \mathbf{UCZ} \tag{5.35}$$

It can be seen that a rotation of $\mathbf{R}$ results in a rotation of matrix $\mathbf{C}$. As the rotation matrix $\mathbf{Z}$ can be arbitrarily chosen, the matrix $\mathbf{C}$ can be arbitrarily rotated. $\mathbf{C}$ is a real square matrix, so it can be decomposed into an orthogonal matrix $\mathbf{Q}$ and upper triangular matrix $\mathbf{P}$ using QR-factorization.

$$\mathbf{CZ} = \mathbf{QPZ} \tag{5.36}$$

$\mathbf{Q}$ is also a rotation matrix. It applies a certain rotation to $\mathbf{P}$ to create $\mathbf{C}$. If $\mathbf{Z}$ is chosen such that it undoes the rotation of $\mathbf{Q}$, then the desired upper triangular matrix $\mathbf{P}$ is obtained. As a result, $\mathbf{C}$ can always be transformed into an upper triangular matrix. $\mathbf{P}$ is essentially a particular rotation of all possible $\mathbf{C}$ matrices. This new $\mathbf{P}$ matrix is used to rewrite (5.33) and (5.34) into:

$$\tilde{\mathbf{R}} = \mathbf{UP} \tag{5.37}$$

$$-2\tilde{\mathbf{S}}^{\mathsf{T}} = \mathbf{P}^{-1}\mathbf{VW} \tag{5.38}$$

A new nonlinear least squares problem can be defined based on (5.26). This nonlinear least squares problem is then rewritten in terms of matrices $\mathbf{U}$, $\mathbf{V}$, $\mathbf{W}$, and $\mathbf{P}$. This will eventually yield a closed the solution. First, defining the new non-linear least squares problem as:

$$\hat{\mathbf{R}}, \hat{\mathbf{S}} = \arg\min_{R,S} \sum_{i=2}^{N} \sum_{j=2}^{M} \left[\|\mathbf{r}_i - \mathbf{s}_j\|^2 - \|\mathbf{r}_1 - \mathbf{s}_j\|^2 - d_{ij}^2 + d_{1j}^2 \right]^2 \tag{5.39}$$

This expression is rewritten in terms of the entries of the matrices $\tilde{\mathbf{R}}$ and $\tilde{\mathbf{S}}$ by using $\mathbf{r}_1 = \mathbf{s}_1 = 0$ and expanding the norms in (5.39).

$$\hat{\mathbf{R}}, \hat{\mathbf{S}} = \arg\min_{\tilde{R},\tilde{S}} \sum_{i=2}^{N} \sum_{j=2}^{M} \left[\|\mathbf{r}_i\|^2 + 2\mathbf{r}_i \cdot \mathbf{s}_j - d_{ij}^2 + d_{1j}^2 \right]^2 \tag{5.40}$$

this is equivalent to:

$$\hat{\mathbf{R}}, \hat{\mathbf{S}} = \arg\min_{\tilde{R},\tilde{S}} \sum_{i=2}^{N} \sum_{j=2}^{M} \left[\|\mathbf{r}_i - \mathbf{r}_1\|^2 + 2(\mathbf{r}_i - \mathbf{r}_1) \cdot (\mathbf{s}_j - \mathbf{s}_1) - d_{ij}^2 + d_{1j}^2 \right]^2 \tag{5.41}$$

Expressing the equation above in terms of entries of matrices $\tilde{\mathbf{R}}$ and $\tilde{\mathbf{S}}$ allows for expression in terms of the entries of $\mathbf{U}$, $\mathbf{V}$, $\mathbf{W}$, and $\mathbf{P}$ using the previously defined relations. This reduces the minimization problem from one of size $(N-1)(M-1)$ to one in terms of the 6 entries of matrix $\mathbf{P}$. This is difficult to do in general dimension $\mathcal{D}$ however. Therefore, it is shown for $\mathcal{D} = 3$. The derivation for the two dimensional case is shown in appendix A.3. Per term in (5.41), the following matrix relation should be used:

$2\|\mathbf{r}_i - \mathbf{r}_1\|^2$	Can be expressed using the ith row of:	$\tilde{\mathbf{R}} = \mathbf{UP}$	(5.42)
$2(\mathbf{r}_i - \mathbf{r}_1) \cdot (\mathbf{s}_j - \mathbf{s}_1)$	Can be expressed using (i, j)th entry of:	$-2\tilde{\mathbf{R}}\tilde{\mathbf{S}}^{\mathsf{T}} = \mathbf{UVW}$	(5.43)

Using these relations, (5.41) is rewritten to:

$$\begin{aligned}\hat{\mathbf{P}} = \arg\min_{P} \sum_{i=2}^{N} \sum_{j=2}^{M} [&(P_{11}U_{i1})^2 + (P_{12}U_{i1} + P_{22}U_{i2})^2 + (P_{13}U_{i1} + P_{23}U_{i2} + P_{33}U_{i3})^2 \\ &+ U_{i1}V_{11}W_{1j} + U_{i2}V_{22}W_{2j} + U_{i3}V_{33}W_{3j} - d_{ij}^2 + d_{1j}^2]^2\end{aligned} \tag{5.44}$$

Expanding exponent terms:

$$\begin{aligned}\hat{\mathbf{P}} = \arg\min_{P} \sum_{i=2}^{N} \sum_{j=2}^{M} [&U_{i1}^2(P_{11}^2 + P_{12}^2 + P_{13}^2) + U_{i2}^2(P_{22}^2 + P_{23}^2) + U_{i3}^2(P_{33}^2) + 2U_{i1}U_{i3}(P_{13}P_{33}) \\ &+ 2U_{i1}U_{i2}(P_{12}P_{22} + P_{13}P_{32}) + 2U_{i2}U_{i3}(P_{23}P_{33}) - k_{ij}]^2\end{aligned} \tag{5.45}$$

$$\text{Where: } k_{ij} = -U_{i1}V_{11}W_{1j} - U_{i2}V_{22}W_{2j} - U_{i3}V_{33}W_{3j} + d_{ij}^2 - d_{1j}^2 \tag{5.46}$$

This equation will be transformed into a matrix equation, leading to the closed form solution which will provide the entries to the $\mathbf{P}$ matrix. To do this a number of matrices and vectors need to be defined. First, define the vector $\mathbf{k}$ which contains the values of k_{ij} for $1 \leq i \leq N-1$ and $1 \leq j \leq M-1$.

$$\mathbf{k} = \begin{bmatrix} k_{11} \\ k_{21} \\ \vdots \\ k_{N-1,1} \\ k_{12} \\ \vdots \\ k_{N-1,M-1} \end{bmatrix} \tag{5.47}$$

Define vector $\mathbf{g}_i$ as:

$$\mathbf{g}_i = \begin{bmatrix} U_{i1}^2 \\ U_{i2}^2 \\ U_{i3}^2 \\ 2U_{i1}U_{i2} \\ 2U_{i1}U_{i3} \\ 2U_{i2}U_{i3} \end{bmatrix} \tag{5.48}$$

These individual column vectors are put into the rows of a matrix $\mathbf{G}$ for $1 \leq i \leq N-1$:

$$\mathbf{G} = \begin{bmatrix} \mathbf{g}_1{}^{\mathrm{T}} \\ \mathbf{g}_2{}^{\mathrm{T}} \\ \vdots \\ \mathbf{g}_{N-1}{}^{\mathrm{T}} \end{bmatrix} \tag{5.49}$$

Define the $(N-1)(M-1) \times 6$ matrix $\mathbf{H}$ containing matrix $\mathbf{G}$ vertically stacked $M-1$ times. Lastly, define $\mathbf{f}$ as the vector containing the unknown terms of $\mathbf{P}$:

$$\mathbf{f} = \begin{bmatrix} P_{11}^2 + P_{12}^2 + P_{13}^2 \\ P_{22}^2 + P_{23}^2 \\ P_{33}^2 \\ P_{12}P_{22} + P_{13}P_{23} \\ P_{13}P_{33} \\ P_{23}P_{33} \end{bmatrix} \tag{5.50}$$

The minimization problem given in (5.45) can now be expressed in matrix form. This minimization problem is a linear least squares problem that finds the entries of the vector $\mathbf{f}$.

$$\mathbf{f} = \arg\min_{\mathbf{f}} \|\mathbf{H}\mathbf{f} - \mathbf{k}\|^2 \tag{5.51}$$

The solution to this linear least squares problem is given by:

$$\mathbf{f} = (\mathbf{H}^{\mathrm{T}}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{k} \tag{5.52}$$

Once $\mathbf{f}$ is found, its definition (5.50) can are used to find the entries of matrix $\mathbf{P}$. This involves solving a quadratic system of equations, with the following solution:

$$P_{33} = \pm\sqrt{f_3} \qquad P_{23} = f_6/P_{33} \tag{5.53}$$

$$P_{22} = \pm\sqrt{f_2 - P_{23}^2} \qquad P_{13} = f_5/P_{33} \tag{5.54}$$

$$P_{12} = (f_4 - P_{13}P_{23})/P_{22} \qquad P_{11} = \pm\sqrt{f_1 - P_{12}^2 - P_{13}^2} \tag{5.55}$$

This fully defines $\mathbf{P}$. $\mathbf{R}$ and $\mathbf{S}$ can be now be found by substituting the values of $\mathbf{P}$ into (5.37) and (5.38). This concludes the finding of the closed form solution.

Refinement of closed form solution

With the closed form approach a solution for the system is found in case the first source and microphone are co-located. However, in case noise is present this solution does not necessarily needs to be the optimal solution to the least squares problem of (5.23). This is because the closed form solution finds the solution to the least squares problem of (5.39). This is a simplified version of the original least squares problem, mainly due to the rank-$\mathcal{D}$ approximation step. Furthermore, although with an increasing error, from simulations it turns out that the closed form approach does also give reasonable results in case the first source and microphone are not perfectly co-located. This means that the closed form solution can be a good starting point for the original least squares problem to refine these obtained coordinates both in presence of noise or in case of co-location offsets. This is because it is likely that the closed form solution is close to the global minimum of(5.23).

In this section section a method, the gradient descent algorithm, is explained which can solve the least squares problem using the coordinates obtained from the closed form solution. The cost function that needs to be minimized is shown in (5.56).

$$f(\mathbf{R},\mathbf{S}) = \sum_{i=1}^{N}\sum_{j=1}^{M}\left[\|\mathbf{r}_i - \mathbf{s}_j\| - d_{ij}\right]^2 \tag{5.56}$$

The gradient descent algorithm tries to minimize this cost function by iteratively adjusting $\mathbf{R}$ and $\mathbf{S}$ in such a way that the total cost decreases every step. In what direction and with what relative size every variable should change is determined by calculating the partial derivatives of f to all variables. Then every variable is independently updated with a small step in the decreasing cost direction.

The coordinate matrices $\mathbf{R}$ and $\mathbf{S}$ are stacked to obtain the matrix $\mathbf{X}$. This $((N+M)\times\mathcal{D})$ matrix is given by (5.57).

$$\mathbf{X} = \begin{bmatrix} {\mathbf{r}_1}^{\mathrm{T}} \\ \vdots \\ {\mathbf{r}_N}^{\mathrm{T}} \\ {\mathbf{s}_1}^{\mathrm{T}} \\ \vdots \\ {\mathbf{s}_M}^{\mathrm{T}} \end{bmatrix} \tag{5.57}$$

The gradient is written as a matrix $\mathbf{J}$ with all partial derivatives of f to all variables in $\mathbf{X}$. Matrix $\mathbf{J}$ is a function of $\mathbf{X}$ and distances matrix $\mathbf{D}$ in the sense that it should be evaluated for those values to obtain the values of the derivatives.

$$\mathbf{J}(\mathbf{X},\mathbf{D}) = \begin{bmatrix} \partial f/\partial{\mathbf{r}_1}^{\mathrm{T}} \\ \vdots \\ \partial f/\partial{\mathbf{r}_N}^{\mathrm{T}} \\ \partial f/\partial{\mathbf{s}_1}^{\mathrm{T}} \\ \vdots \\ \partial f/\partial{\mathbf{s}_M}^{\mathrm{T}} \end{bmatrix} \tag{5.58}$$

The expressions for $\partial f/\partial\mathbf{r}_i$ and $\partial f/\partial\mathbf{s}_\mathbf{j}$ are given in (5.59) and (5.60) respectively. The derivations of these expressions are given in appendix A.4.

$$\frac{\partial f(\mathbf{R},\mathbf{S})}{\partial\mathbf{r}_\mathbf{i}} = 2\sum_{j=1}^{N}\left[\|\mathbf{r}_i - \mathbf{s}_j\| - d_{ij}\right]\frac{\mathbf{r}_i - \mathbf{s}_j}{\|\mathbf{r}_i - \mathbf{s}_j\|} \tag{5.59}$$

$$\frac{\partial f(\mathbf{R},\mathbf{S})}{\partial\mathbf{s}_\mathbf{j}} = 2\sum_{i=1}^{M}\left[\|\mathbf{r}_i - \mathbf{s}_j\| - d_{ij}\right]\frac{\mathbf{s}_j - \mathbf{r}_i}{\|\mathbf{r}_i - \mathbf{s}_j\|} \tag{5.60}$$

The gradient descent algorithm then works very straightforward. First $\mathbf{X}$ is initialized with the coordinates found with the closed form solution. Then $\mathbf{J}(\mathbf{X},\mathbf{D})$ is evaluated and the coordinate vector $\mathbf{X}$ is updated by subtracting the partial derivatives multiplied by stepsize δ from the previous coordinate vector. This step is shown in (5.61).

$$\mathbf{X}^{n+1} = \mathbf{X}^{n} - \delta\mathbf{J}(\mathbf{X}^{n},\mathbf{D}) \tag{5.61}$$

The step size δ should not be chosen to large because that could result in the movement of variables outside the basin of the global minimum of function (5.23). The step size can be determined empirically, but it also possible to use an algorithm. The line search algorithm tries to determine the optimal step size at every iteration of the gradient descent. It evaluates (5.61) for many different step sizes and determines what the cost for the newly determined $\mathbf{X}$ would be. Then it selects the step size that resulted in the biggest reduction of the cost function.

Finally a stopping criteria is also required. This criteria determines when the gradient descent finished. For this algorithm a predefined step number is used as the stopping criteria.

Large error detection

In case an measurement has an very large error, the closed form solution can result in imaginary solutions. It turns out that if one measurement does not comply with the others this is already the case. For example, such an erroneous result could occur due to reflections of the audio signal.

To be able to deal with such problems an algorithm is designed which can detect which events likely contain measurement errors. From the set of M events the algorithm generates all possible combinations of 4 events. All these combinations are fed into the TOA closed form algorithm. In case a measurement is wrong, it will likely return imaginary results. The algorithm keeps track which combinations result in imaginary results. The audio events which are often present in combinations that return imaginary results can then be assumed to be wrong.

6 Implementation and Verification of Algorithms

This chapter explains how the algorithms are implemented. Afterwards it is explained how the algorithms should be used. In the next part the algorithms are tested in two ways. First the results of simulations are shown to analyze the behaviour of both algorithms. Then the processes of integrating the system into the complete system is elaborated and the test results of the system in the real world are shown.

6.1. Practical usage of algorithms

TDOA Algorithm

The TDOA algorithm is an anchor-based algorithm which means that anchors are required. This means the first step in localization is the (manual) measurement of the locations of the anchors. The anchors can be microphones or speakers of the DICENTIS unit or external units can be used. Then the ranging process can start. When this is finished the algorithm determines the locations of the sources.

TOA Algorithm

The TOA algorithm is not an anchor-based algorithm, but if the system needs to pinpointed to our world still 3 or 4 anchors are required. The TOA algorithm can then be used in several ways. By using all the speaker of the DICENTIS units to generate an event and letting all microphones record those events, the relative locations of all microphones and speakers can be determined with the algorithm. An addition to this method can be made. When one wireless speaker is continuously generating events and someone is walking around with that speaker, from many more locations the time of flights to all microphones are known. The locations of these events will also be calculated by the algorithm, even though they are irrelevant. However, this extra information does increase the localization accuracy of the microphones. In case a powerful omni-directional speaker is used for this addition it is also likely that the added events contain less measurement noise. Finally the TOA algorithm can be anchored if required. For this procrustes analysis can be used.

6.2. Implementation details

For the gradient descent of the TOA algorithm the step size should be chosen in such a way the that the algorithm converges as quick as possible. In case the step size is too small the algorithm will require unnecessarily many steps to converge. In chapter 5 an line search algorithm was explained. After verification this algorithm turns out to work. However, the steps size turned out to vary very little in any cases. So for runtime considerations the line search algorithm is removed and the step size is fixed to 0.01.

6.3. Simulations

The first step in verifying the correct behaviour of the algorithms is the execution of simulations. In this section a couple simulation setups are explained. Then both algorithms are tested in these simulations setups and the results are compared. The simulations are performed to determine the performance of the algorithms in different situations. The parameters that will be considered in the simulations are:

- The amount of noise on the distance measurements. This noise is assumed to be Gaussian with a particular standard deviation σ and mean μ.
- The number of sources.
- The number of microphones. This is only relevant for the TOA algorithm, because the TDOA algorithm uses a fixed number of anchors (in this case microphones) to localize speakers.
- The locations of the anchors.
- The co-location offset, which is only applicable to the TOA algorithm.

The effect of the values of these parameters will be considered independently. For this a base set of parameters is used. These parameters are chosen in such a way that they represent a realistic situation. The values of these parameters are given in table 6.1. The performance of the algorithms in particular situations are measured by using

Table 6.1: The parameters which are considered as base values from which the simulations are performed.

Reference Setup Parameters	
#Anchors	4
#Sources	12
#Microphones (only for TOA)	24
Standard Deviation [m]	0.03
Mean [mu]	0.0
Room Dimension [m]	[10,10,4]
Source/Microphone Locations	Random
Anchors locations	Corners
Colocation offset [m]	0.0

the root mean square error (RMSE). This paramter is determined for the sources and microphones separately. Each particular simulation is executed $T = 500$ times with different realizations of the noise and also the locations are re-randomized every iteration so the results can be averaged over T samples. In this way the RMSE of the microphones and sources for a particular set of parameters can be calculated as a performance indicator of a particular setup. Additionally, because each setup is executed T times the standard deviation of the RMSE's of setups can also be determined. The RMSE is determined as in (6.1). In these equations N is the number of microphones or sources in a setup, $\hat{x}_{ik}$ denotes the estimated location and x_{ik} denotes the actual position of source or microphone i in iteration k.

$$\text{RMSE} = \frac{1}{T}\sum_{k=1}^{T}\sqrt{\frac{1}{N}\sum_{i=1}^{N}\|\mathbf{x}_{ik} - \hat{\mathbf{x}}_{ik}\|^2} \tag{6.1}$$

$$\text{Standard deviation RMSE} = \sqrt{\frac{1}{T-1}\sum_{k=1}^{T}|RMSE_k - \mu_k|^2} \tag{E.1}$$

$$\mu_k = \frac{1}{T}\sum_{k=1}^{T} RMSE_k$$

The effect of noise in arrival times

Both algorithms have a certain set of input data. The TDOA algorithm only requires the arrival times and the TOA algorithm also requires times of departure. In this part the noise of the departure times will be assumed zero and so only the noise of the arrival times will be nonzero. The initial assumption as stated in the program of requirements is a standard deviation of no more than 3cm and a mean of no more than 1cm. However, in this first setup the effects on the RMSE for standard deviations from 0 to 15 cm are simulated. Figure 6.1 shows the resulting mean RMSE and Figure B.1 shows the standard deviations. The gradient descent method is clearly performing the best.

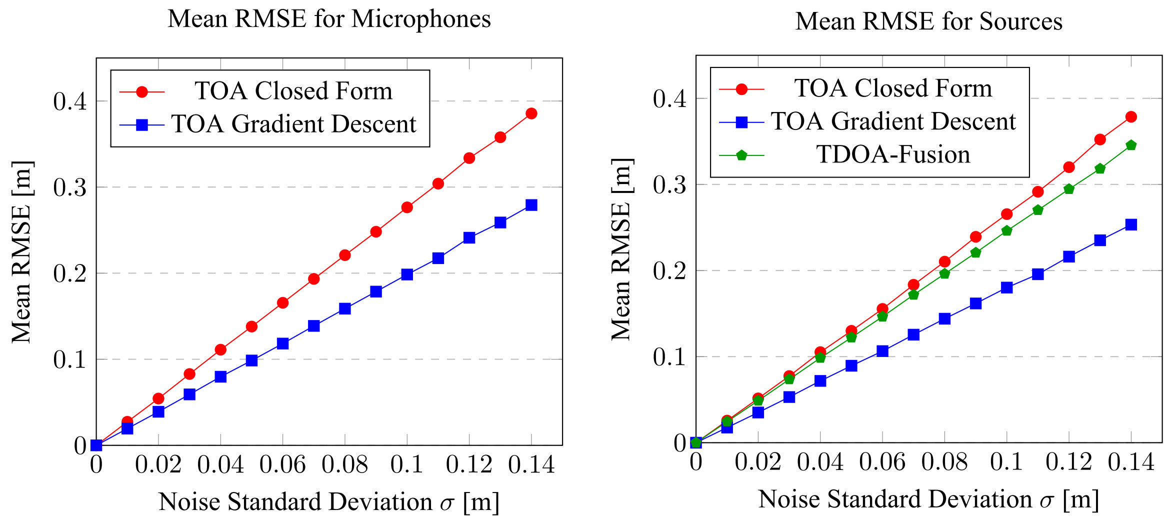

Figure 6.1: For each value of the noise 500 trials are performed to obtain the mean of the RMSE as a function of the arrival time noise standard deviation.

Different number of sources and microphones

The next setup simulates the effect of using more sources and microphones. Because the TDOA algorithm determines the locations of the sources independently, the accuracy will not change. However, for the TOA algorithm an increase in accuracy for both the microphones and sources is expected. This is because more sources will increase the accuracy of the microphone estimation and this higher accuracy in microphone localization in turn can result in a higher accuracy in speaker localization. The number of sources is increased from 4 to 50 and the results are shown in Figure 6.2 and Figure B.2.

An increasing number of microphones will effect the accuracy of the TOA algorithm in a similar way. It turns out that the RMSE decreases significantly when increasing the number of microphones from 4 to 10. From that point however, adding more microphones does not increase the accuracy anymore. The figures of the RMSE for an increasing number of microphones are shown in appendix B.1.

Locations of anchors

Another important factor for the localization accuracy are the locations of the anchors. In the TOA algorithm the anchors are used to rotate, shift and reflect the found general solution to the real locations. When the anchors are all in front of the room, close to each other, it is expected that small errors of the anchored mics result in larger errors at the other side of the room. For that reason it is preferable to have the anchors in the corners of the room. However, measuring the locations of the anchors when they are in the corners of a room is likely to be more difficult then when they are close together in the front of the room. Because the four anchors should not be placed in a plane, they are placed in a tetrahedron with sides of 1 meter. Table 6.2 summarizes the results in comparison with the base situation where the anchors are at the corners of the room. For the TOA algorithms the accuracy was effected a lot, but not as much as for the TDOA algorithm as was already anticipated in the introduction. For a commercial implementation such an tetrahedron could be pre-fabricated where the anchors are placed at the fixed corners. This can greatly reduce the time required for measuring the anchor locations.

Table 6.2: Root mean square error for comparison for different anchor locations in a a room of 10x10x10. In the reference setup the four anchors are placed at 4 corners. In the second setup the anchors are placed in a tetrahedron with sides of length 1m at x=5, y=5 and z=0.

RMSE Microphone locations [cm]

	Reference	Tetrahedron 1
TOA CF	5.9	12.5
TOA GD	4.2	11.6

RMSE Source locations [cm]

	Reference	Tetrahedron 1
TOA CF	5.3	13.2
TOA GD	3.7	12.1
TDOA	5.0	52.8

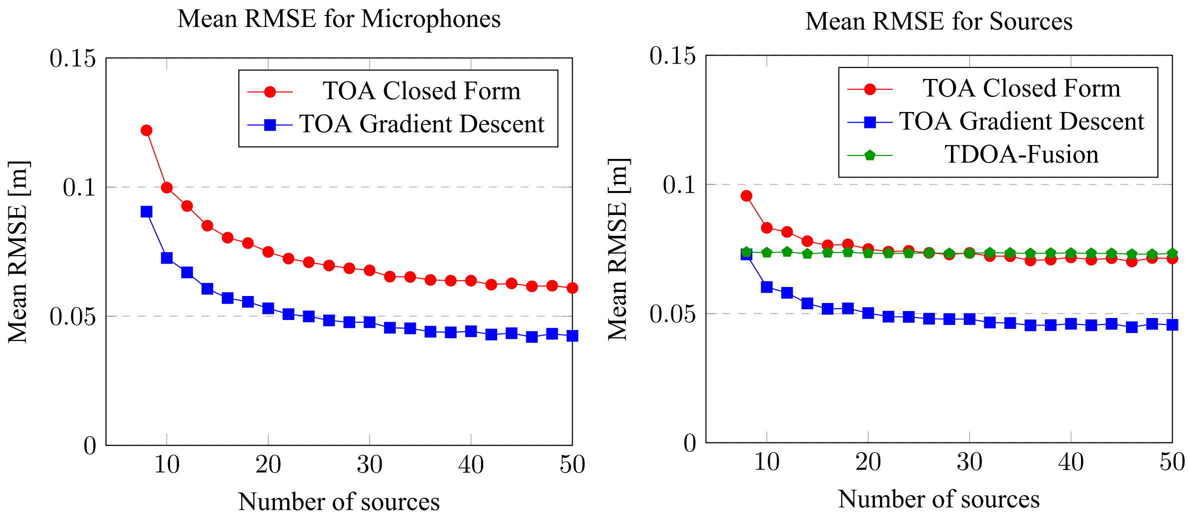

Figure 6.2: For each value of the noise 500 trials are performed to obtain the mean of the RMSE as a function of the arrival time noise standard deviation.

Table 6.3: Root mean square error for comparison of a three dimensional setup to a two dimensional setup.

RMSE Microphone locations [cm]		
	Reference	2D
TOA CF	5.9	4.4
TOA GD	4.2	2.0

RMSE Source locations [cm]		
	Reference	2D
TOA CF	5.3	3.9
TOA GD	3.7	1.7
TDOA	5.0	3.5

Co-location offset

Another parameter is the co-location offset. The co-location offset is defined as the distance from the source to the microphone of an unit. This parameter is important because our closed form solution assumes that the first speaker and source are perfectly co-located, while this is most likely not the case. Figure 6.3 shows the RMSE of the sources and speakers again, but now for an increasing co-location offset from 0 to 15cm. The results show that the closed form solution is susceptible for an co-location offset, but that the gradient descent method is still able to successfully minimize the localization error significantly.

Setup in 2 dimensions

Above simulations are done for setups in three dimensions. However, in some cases the sources and microphones may lay in a plane. In that case, the algorithms can be simplified to a 2D version. Table 6.3 compares the base situation where the events and sources are in distributed in a 3D space of 10 by 10 by 10 meters with an situation where the events and sources are distributed in a plane of 10 by 10 meters. It can be seen that for all algorithms the localization error decreases significantly and for the gradient descent with more than 50%.

6.4. Results of Integrated System

This section describes a real-world test of the TOA and the TDOA algorithms based off of actual measurements.

Unit Cube Measurement

The first measurement setup tht will be discussed is one of a unit cube. The measurement setup is as follows. A unit cube is constructed by placing 4 PVC towers of one meter approximately one meter apart. Seven AKG C417 microphones are placed on the vertices of the unit cube: 4 are placed at the tops of the PVC towers, 3 are attached to the bottom of the PVC towers. All microphones are measured up so results can be compared with reality. Microphones are connected to a laptop running a MATLAB program via the Scarlett 18i20 USB Audio Interface. One Bosch DICENTIS Conference System unit is used as audio source. To obtain emission times, one additional

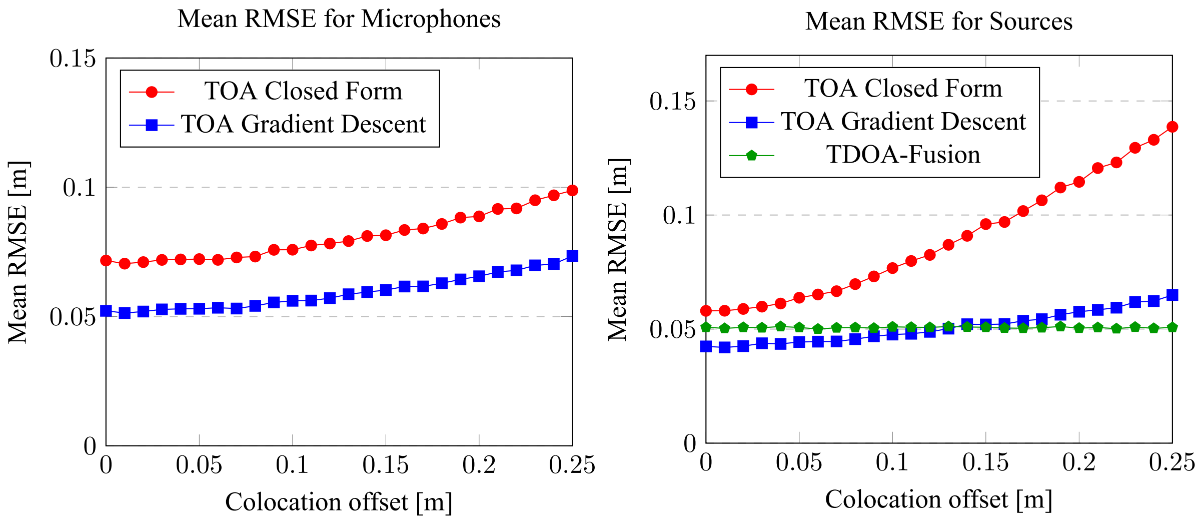

Figure 6.3: The mean RMSE for an increasing distance between the first soure and the first event (co-location offset).

AKG C417 microphone is taped onto the loudspeaker of the unit. The reception time of the transmitted signal of this microphone will be used as emission time for TOA measurements. This is necessary, as it is impossible to obtain emission times from the DICENTIS conference units. This requires access to the software that run on the units, which is not realistic to alter within this project. To simulate multiple audio sources, the DICENTIS unit is placed on multiple pre-defined locations: halfway between each PVC tower at 4 different heights.

The measurement is conducted in the vicinity of many tables and chairs in the “Tellegen Hall” at the TU Delft University. These can cause eventual reflections, which contribute to possible faulty measurements.

An image of the measurement setup can be seen in figure 6.4. In addition, a 3 dimensional view of the measured microphone and source locations can be seen in figure 6.5. The co-located measurement required for the TOA algorithm is also performed.

After performing the measurements, the results are fed to both algorithms. The results plotted versus the ground truth for the TOA algorithm can be seen in Figure 6.6. 4 anchors are chosen to transform the results. Both the microphone locations as well as the source locations are successfully recovered. For the microphone locations, the RMSE is found to be 7.98 cm. This is within the bounds set by the program of requirements. The source locations have an RMSE of 10.02 cm. The error detection also works. A number of the performed measurements contained incorrect results. It could be seen that these measurements were extracted successfully. For the TDOA algorithm the results are shown in figure 6.7. The given results use 4 anchors for fair comparison with the TOA algorithm. The RMSE is found to be 12.3 cm. When all 7 microphones are chosen as anchors, the RMSE is much lower at 8.3 cm.

This is however a relatively simple setup. These tests show that both algorithms perform adequately with real-world measurements. The calculated RMSE may not be completely representative for the ‘true’ RMSE. This is due to possible inaccuracies the initial measurements done by hand to determine the positions of microphones and speakers.

Figure 6.4: Unit cube measurement setup.

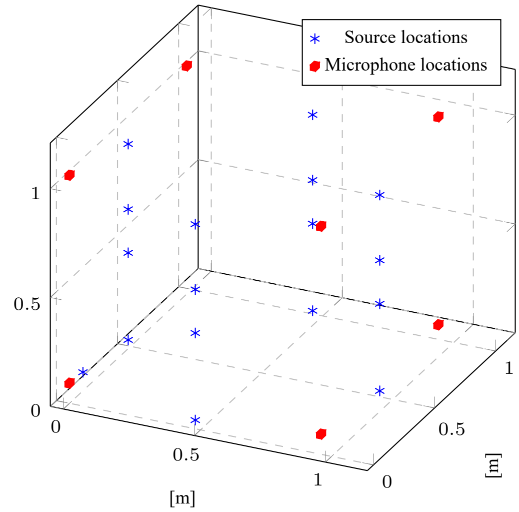

Figure 6.5: Measured microphone and source locations. Unit cube for reference.

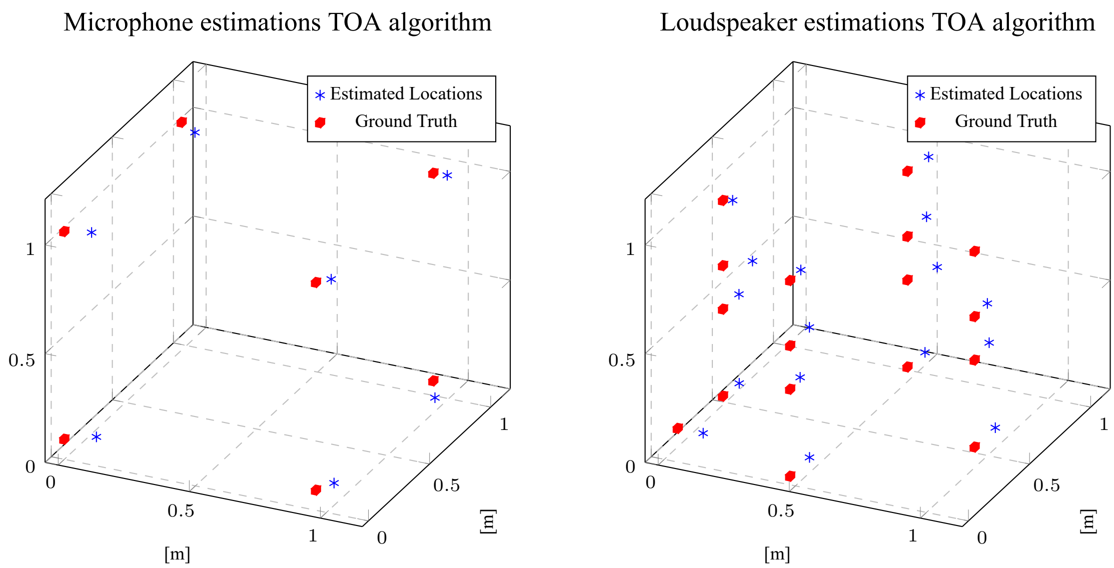

Figure 6.6: Recovered locations by the TOA algorithm. Only the gradient descent refined coordinates are shown for clarity.

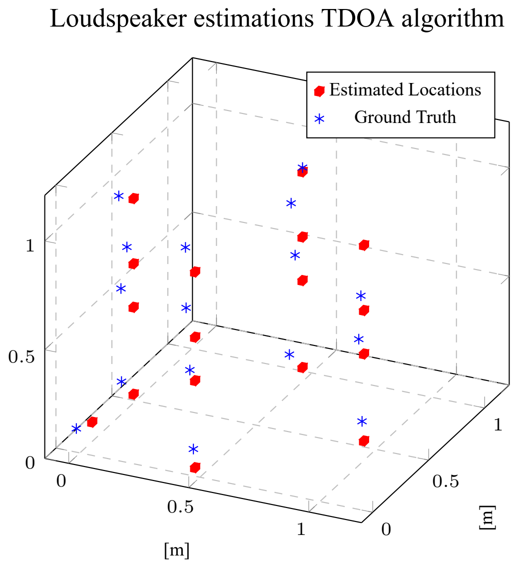

Figure 6.7: Recovered speaker locations by the TDOA algorithm

7 Discussion and Conclusion

After a survey of possible localization techniques, in this thesis the two methods with good chance to perform the best are implemented and tested for the particular situation of conferencing systems. In the thesis many simulations are done to study the performance of the algorithms under different situations. In the program of requirements it is stated that the algorithm should find the locations of the units with an average error of at most 10 cm. It turns out that the results of both the simulations and the real world setup comply with this requirement in many different situations. In fact, even with larger errors occurring than initially assumed, the system is able to provide localization with accuracy within the requirements. This is due to the error detection being able to effectively detect erroneous measurements.

According to the program of requirements the algorithms should also work in 2 dimensional situations. Those 2D versions are also implemented and it is shown that even better results can be obtained in this case: an improvement of over 50% relative to 3 dimensions.

A simulation about the effect of the locations shows that it is not necessarily required to have the anchors in the corners of the room. The TOA algorithm can still comply with the program of requirements in case the measurement noise standard deviation is smaller than 3 cm . The proposed tetrahedron setup is interesting because the manual setup time is reduced.

The requirements on the execution time and scalability are not sufficiently tested. In the current prototype the algorithm runs on a laptop using MATLAB. To determine whether the algorithm satisfies these conditions, it needs to be tested on the DICENTIS system.

To conclude, both algorithms were verified to behave in compliance with the program of requirements. Because the TOA gradient descent method is outperforming the other algorithms in terms of accuracy, easy of use and because the fact that both sources and receivers are localized it is recommended to use this algorithm for the localization of conferencing systems.

Future work

While the presented results comply with the requirements, there are a number of improvements in terms of speed and namely scalability that can be made in future projects:

- The current error detection is also likely too computational complex to run on a simple access point like the hub of the DICENTIS system. In addition to this, the current error detector is too coarse: if a measurement error occurs for one microphone and one source, all measurements to that particular source are removed. In future work more efficient methods for finding erroneous measurements could be developed.

- In the real world tests the anchors were always microphones. However, for the TDOA algorithm the distance measurements could become more reliable if speakers are used as anchors to localize microphones. In this setup the speakers can be placed at the corners, directed to the microphones. In future work the accuracy of such a setup can be evaluated.

A

Additional Derivations

A.1. TDOA-Fusion: Rewriting to obtain matrix form

This section details the rewriting (5.8) into a form that can be put into matrix form to find a closed form solution. As a starting point, consider the equation below:

$$\left(\Delta d_{ik} + d_{kj}\right)^2 = d_{ij}^2 \tag{A.1}$$

d_{ij} represents the distance between microphone $\mathbf{r}_i$ and receiver $\mathbf{s}_j$. As a result, d_{ij} and can be rewritten using equations (5.4) and (5.3) in terms of the euclidean distance between them:

$$\left(\Delta d_{ik} + d_{kj}\right)^2 = \left\|\mathbf{r}_i - \mathbf{s}_j\right\|^2 \tag{A.2}$$

Expanding terms on the LHS results in:

$$\Delta d_{ik}^2 + 2 d_{kj} \Delta d_{ik} + d_{kj}^2 = \left\|\mathbf{r}_i - \mathbf{s}_j\right\|^2 \tag{A.3}$$

The next goal is to eliminate the d_{kj}^2 term. Similarly to the step above, the d_{kj}^2 term can be expressed with an euclidean norm:

$$\Delta d_{ik}^2 + 2 d_{kj} \Delta d_{ik} + \left\|\mathbf{r}_k - \mathbf{s}_j\right\|^2 = \left\|\mathbf{r}_i - \mathbf{s}_j\right\|^2 \tag{A.4}$$

To eliminate this new norm, the RHS of (A.4) must be rewritten slightly. This is done by introducing an $\mathbf{r}_k$ term in norm on the RHS of the equation, and by making use of the rule for euclidean norms: $\|\mathbf{a} - \mathbf{b}\|^2 = \|\mathbf{a}\|^2 + \|\mathbf{b}\|^2 - 2\mathbf{a} \cdot \mathbf{b}$

$$\begin{aligned} \Delta d_{ik}^2 + 2 d_{kj} \Delta d_{ik} + \left\|\mathbf{r}_k - \mathbf{s}_j\right\|^2 &= \left\|(\mathbf{r}_i - \mathbf{r}_k) - (\mathbf{s}_j - \mathbf{r}_k)\right\|^2 \\ &= \left\|\mathbf{r}_i - \mathbf{r}_k\right\|^2 + \left\|\mathbf{s}_j - \mathbf{r}_k\right\|^2 - 2(\mathbf{r}_i - \mathbf{r}_k) \cdot (\mathbf{s}_j - \mathbf{r}_k) \\ &= \left\|\mathbf{r}_i - \mathbf{r}_k\right\|^2 + \left\|\mathbf{r}_k - \mathbf{s}_j\right\|^2 - 2(\mathbf{r}_i - \mathbf{r}_k) \cdot (\mathbf{s}_j - \mathbf{r}_k) \end{aligned} \tag{A.5}$$

The desired norm can now be eliminated:

$$\Delta d_{ik}^2 + 2 d_{kj} \Delta d_{ik} = \left\|\mathbf{r}_i - \mathbf{r}_k\right\|^2 - 2(\mathbf{r}_i - \mathbf{r}_k) \cdot (\mathbf{s}_j - \mathbf{r}_k) \tag{A.6}$$

Using the same rule for euclidean norms as used in (A.5) on the remaining norm in (A.6):

$$\Delta d_{ik}^2 + 2 d_{kj} \Delta d_{ik} = \left\|\mathbf{r}_i\right\|^2 + \left\|\mathbf{r}_k\right\|^2 - 2\mathbf{r}_i \cdot \mathbf{r}_k - 2(\mathbf{r}_i - \mathbf{r}_k) \cdot (\mathbf{s}_j - \mathbf{r}_k) \tag{A.7}$$

$$\Delta d_{ik}^2 + 2 d_{kj} \Delta d_{ik} = \left\|\mathbf{r}_i\right\|^2 + \left\|\mathbf{r}_k\right\|^2 - 2\mathbf{r}_i \cdot \mathbf{r}_k - 2\mathbf{r}_i \cdot (\mathbf{s}_j - \mathbf{r}_k) + 2\mathbf{r}_k \cdot (\mathbf{s}_j - \mathbf{r}_k) \tag{A.8}$$

$$\Delta d_{ik}^2 + 2 d_{kj} \Delta d_{ik} = \left\|\mathbf{r}_i\right\|^2 + \left\|\mathbf{r}_k\right\|^2 - 2\mathbf{r}_i \cdot \mathbf{s}_j + 2\mathbf{r}_k \cdot \mathbf{s}_j - 2\mathbf{r}_k \cdot \mathbf{r}_k \tag{A.9}$$

Rewriting and using the rule $\mathbf{a} \cdot \mathbf{a} = \|a\|^2$:

$$\Delta d_{ik}^2 + 2 d_{kj} \Delta d_{ik} = \left\|\mathbf{r}_i\right\|^2 - \left\|\mathbf{r}_k\right\|^2 - 2(\mathbf{r}_i - \mathbf{r}_k) \cdot \mathbf{s}_j \tag{A.10}$$

Rearranging and dividing by 2:

$$(\mathbf{r}_i - \mathbf{r}_k) \cdot \mathbf{s}_j = -d_{kj} \Delta d_{ik} + \frac{1}{2}\left(\left\|\mathbf{r}_i\right\|^2 - \left\|\mathbf{r}_k\right\|^2 - \Delta d_{ik}^2\right) \tag{A.11}$$

A.2. TDOA-Fusion: Solving quadratic equation

In this appendix, (A.12) is solved for d_{kj}.

$$d_{kj}^2 = \left\|\mathbf{r}_k - \mathbf{A}_k^+ \left(d_{kj}\mathbf{b}_k + \mathbf{c}_k\right)\right\|^2 \tag{A.12}$$

Rewriting by making use for the following rule: $\|\mathbf{a} - \mathbf{b}\|^2 = \|\mathbf{a}\|^2 + \|\mathbf{b}\|^2 - 2\mathbf{a} \cdot \mathbf{b}$

$$d_{kj}^2 = \underbrace{\|\mathbf{r}_k\|^2}_{1} + \underbrace{\left\|\mathbf{A}_k^+ \left(d_{kj}\mathbf{b}_k + \mathbf{c}_k\right)\right\|^2}_{2} - \underbrace{2\mathbf{r}_k \cdot \left(\mathbf{A}_k^+ \left(d_{kj}\mathbf{b}_k + \mathbf{c}_k\right)\right)}_{3} \tag{A.13}$$

The different terms in RHS of the equation above are numbered. Term 2 can be expanded using the rule: $\|\mathbf{a} + \mathbf{b}\|^2 = \|\mathbf{a}\|^2 + \|\mathbf{b}\|^2 + 2\mathbf{a} \cdot \mathbf{b}$. Term 3 can be rewritten slightly.

$$d_{kj}^2 = \underbrace{\|\mathbf{r}_k\|^2}_{1} + \underbrace{\left\|\mathbf{A}_k^+ d_{kj}\mathbf{b}_k\right\|^2 + \left\|\mathbf{A}_k^+ \mathbf{c}_k\right\|^2 + 2(\mathbf{A}_k^+ d_{kj}\mathbf{b}_k) \cdot (\mathbf{A}_k^+ \mathbf{c}_k)}_{2} - \underbrace{2\mathbf{r}_k \cdot \left(\mathbf{A}_k^+ d_{kj}\mathbf{b}_k\right) - 2\mathbf{r}_k \cdot \left(\mathbf{A}_k^+ \mathbf{c}_k\right)}_{3} \tag{A.14}$$

The d_{kj} terms can now be extracted out of terms 2 and 3:

$$d_{kj}^2 = \underbrace{\|\mathbf{r}_k\|^2}_{1} + \underbrace{d_{kj}^2 \left\|\mathbf{A}_k^+ \mathbf{b}_k\right\|^2 + \left\|\mathbf{A}_k^+ \mathbf{c}_k\right\|^2 + 2d_{kj}(\mathbf{A}_k^+ \mathbf{b}_k) \cdot (\mathbf{A}_k^+ \mathbf{c}_k)}_{2} - \underbrace{2d_{kj}\mathbf{r}_k \cdot (\mathbf{A}_k^+ \mathbf{b}_k) - 2\mathbf{r}_k \cdot (\mathbf{A}_k^+ \mathbf{c}_k)}_{3} \tag{A.15}$$

The equation can now be put into quadratic form by rearranging:

$$0 = d_{kj}^2 \left(\left\|\mathbf{A}_k^+ \mathbf{b}_k\right\|^2 - 1\right) + d_{kj} \left(2(\mathbf{A}_k^+ \mathbf{b}_k) \cdot (\mathbf{A}_k^+ \mathbf{c}_k) - 2\mathbf{r}_k \cdot (\mathbf{A}_k^+ \mathbf{b}_k)\right) + \|\mathbf{r}_k\|^2 + \left\|\mathbf{A}_k^+ \mathbf{c}_k\right\|^2 - 2\mathbf{r}_k \cdot (\mathbf{A}_k^+ \mathbf{c}_k) \tag{A.16}$$

$$0 = \alpha d_{kj}^2 + \beta d_{kj} + \gamma \tag{A.17}$$

Where α, β, and γ are:

$$\alpha = \left\|\mathbf{A}_k^+ \mathbf{b}_k\right\|^2 - 1 \tag{A.18}$$

$$\beta = 2(\mathbf{A}_k^+ \mathbf{b}_k) \cdot (\mathbf{A}_k^+ \mathbf{c}_k) - 2\mathbf{r}_k \cdot (\mathbf{A}_k^+ \mathbf{b}_k) \tag{A.19}$$

$$\gamma = \|\mathbf{r}_k\|^2 + \left\|\mathbf{A}_k^+ \mathbf{c}_k\right\|^2 - 2\mathbf{r}_k \cdot (\mathbf{A}_k^+ \mathbf{c}_k) \tag{A.20}$$

This gives the following solution for d_{kj}:

$$d_{kj} = \frac{-\beta \pm \sqrt{\beta^2 - 4\alpha\gamma}}{2\alpha} \tag{A.21}$$

Using the definitions above, the solution for d_{kj} can be obtained.

A.3. Crocco: 2 dimensional closed form

In this appendix the closed form solution for $\mathbf{R}$ and $\mathbf{S}$ are determined for $\mathcal{D} = 2$ using the relations given by (5.42) and (5.43), First, equation (5.40) can be rewritten to:

$$\hat{\mathbf{P}} = \arg\min_{P} \sum_{i=2}^{N} \sum_{j=2}^{M} [(P_{11}U_{i1})^2 + (P_{12}U_{i1} + P_{22}U_{i2})^2 + U_{i1}V_{11}W_{1j} + U_{i2}V_{22}W_{2j} - d_{ij}^2 + d_{1j}^2]^2 \quad \text{(A.22)}$$

Expanding exponent terms:

$$\hat{\mathbf{P}}, \hat{\mathbf{S}} = \arg\min_{P} \sum_{i=2}^{N} \sum_{j=2}^{M} [U_{i1}^2(P_{11}^2 + P_{12}^2) + U_{i2}^2(P_{22}^2) + 2U_{i1}U_{i2}(R_{12}R_{22}) - k_{ij}]^2 \quad \text{(A.23)}$$

$$\text{Where: } k_{ij} = -U_{i1}V_{11}W_{1j} - U_{i2}V_{22}W_{2j} + d_{ij}^2 - d_{1j}^2 \quad \text{(A.24)}$$

This equation will be transformed into a matrix equation, leading to the closed form solution which will provide the entries to the $\mathbf{P}$ matrix. To do this a number of matrices and vectors need to be defined. First, define the vector $\mathbf{k}$ which contains the values of k_{ij} for $1 \le i \le N-1$ and $1 \le j \le M-1$.

$$\mathbf{k} = \begin{bmatrix} k_{11} \\ k_{21} \\ \vdots \\ k_{N-1,1} \\ k_{12} \\ \vdots \\ k_{N-1,M-1} \end{bmatrix} \quad \text{(A.25)}$$

Define vector $\mathbf{x}_i$ as:

$$\mathbf{x}_i = \begin{bmatrix} U_{i1}^2 \\ U_{i2}^2 \\ 2U_{i1}U_{i2} \end{bmatrix} \quad \text{(A.26)}$$

These individual column vectors will be put into the rows of a matrix $\mathbf{X}$ for $1 \le i \le N-1$:

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1{}^{\mathrm{T}} \\ \mathbf{x}_2{}^{\mathrm{T}} \\ \vdots \\ \mathbf{x}_{N-1}{}^{\mathrm{T}} \end{bmatrix} \quad \text{(A.27)}$$

Define the $(N-1)(M-1) \times 3$ matrix $\mathbf{Y}$ containing matrix $\mathbf{X}$ vertically stacked $M-1$ times. Lastly, define $\mathbf{f}$ as the vector containing the unknown terms of $\mathbf{P}$:

$$\mathbf{f} = \begin{bmatrix} P_{11}^2 + P_{12}^2 \\ P_{22}^2 \\ P_{12}P_{22} \end{bmatrix} \quad \text{(A.28)}$$

The minimization problem given in (5.45) can now be expressed in matrix form. This minimization problem is a linear least squares problem that finds the entries of the vector $\mathbf{f}$.

$$\mathbf{f} = \arg\min_{\mathbf{f}} \|\mathbf{Y}\mathbf{f} - \mathbf{k}\|^2 \quad \text{(A.29)}$$

The solution to this linear least squares problem is given by:

$$\mathbf{f} = (\mathbf{P}^{\mathrm{T}}\mathbf{P})^{-1}\mathbf{P}^{\mathrm{T}}\mathbf{k} \quad \text{(A.30)}$$

Once $\mathbf{f}$ is found, its definition (5.50) can be used to find the entries of matrix $\mathbf{P}$. This involves solving a quadratic system of equations, with the following solution:

$$P_{33} = \pm\sqrt{f_2} \quad \text{(A.31)}$$

$$P_{12} = \frac{f_3}{P_{22}} \quad \text{(A.32)}$$

$$P_{11} = \pm\sqrt{f_1 - P_{12}^2} \quad \text{(A.33)}$$

A.4. Crocco: Derivation of gradient

In this section the gradient of (A.34) is derived. That is, an expression for all the partial derivatives is determined.

$$f(\mathbf{R}, \mathbf{S}) = \sum_{i=1}^{N} \sum_{j=1}^{M} \alpha_{ij}^2 \tag{A.34}$$

$$\alpha_{ij} = [\|\mathbf{r}_i - \mathbf{s}_j\| - d_{ij}] \tag{A.35}$$

The derivation to find the the partial derivatives to r_i will be shown here. The derivation for the partial derivatives to s_j follows the same steps.

In (A.36) the partial derivative to r_i is written. Then the derivative operator is shifted into the summation in (A.37). The summation from i=1 to N will result in N different r_i terms. Because the derivative is taken to one particular r_i all other terms but the particular r_i are removed in (A.38).

$$\frac{\partial}{\partial r_i}(f) = \frac{\partial}{\partial r_i}\left(\sum_{i=1}^{N} \sum_{j=1}^{M} \alpha_{ij}^2\right) \tag{A.36}$$

$$\frac{\partial}{\partial r_i}(f) = \sum_{i=1}^{N} \sum_{j=1}^{M} \frac{\partial}{\partial r_i}\left(\alpha_{ij}^2\right) \tag{A.37}$$

$$\frac{\partial}{\partial r_i}(f) = \sum_{j=1}^{M} \frac{\partial}{\partial r_i}\left(\alpha_{ij}^2\right) \tag{A.38}$$

The next step is to actually compute the partial derivative of α_{ij}^2. Following from the chain rule:

$$\frac{\partial}{\partial r_i}\left(\alpha_{ij}^2\right) = 2 \cdot \alpha_{ij}\left(\frac{\partial}{\partial r_i}\right)\alpha_{ij} \tag{A.39}$$

Now the partial derivative of α_{ij} to r_i is computed:

$$\frac{\partial}{\partial r_i}\left(\alpha_{ij}\right) = \frac{\partial}{\partial r_i}\left([\|\mathbf{r}_i - \mathbf{s}_j\| - d_{ij}\right) \tag{A.40}$$

$$\frac{\partial}{\partial r_i}\left(\alpha_{ij}\right) = \frac{2(\mathbf{r}_i - \mathbf{s}_j)}{2\|\mathbf{r}_i - \mathbf{s}_j\|} \tag{A.41}$$

Now combining (A.35), (A.39) and (A.41) to substitute into (A.38) results in:

$$\frac{\partial}{\partial r_i}(f) = 2\sum_{j=1}^{N} [\|\mathbf{r}_i - \mathbf{s}_j\| - d_{ij}]\frac{\mathbf{r}_i - \mathbf{s}_j}{\|\mathbf{r}_i - \mathbf{s}_j\|} \tag{A.42}$$

The same steps can be performed to arrive at the expression for the partial derivatives to s_j:

$$\frac{\partial}{\partial s_j}(f) = 2\sum_{i=1}^{M} [\|\mathbf{r}_i - \mathbf{s}_j\| - d_{ij}]\frac{\mathbf{s}_j - \mathbf{r}_i}{\|\mathbf{r}_i - \mathbf{s}_j\|} \tag{A.43}$$

B

Additional Simulation Results

B.1. Additional simulation results

Standard deviation for increasing noise and sources

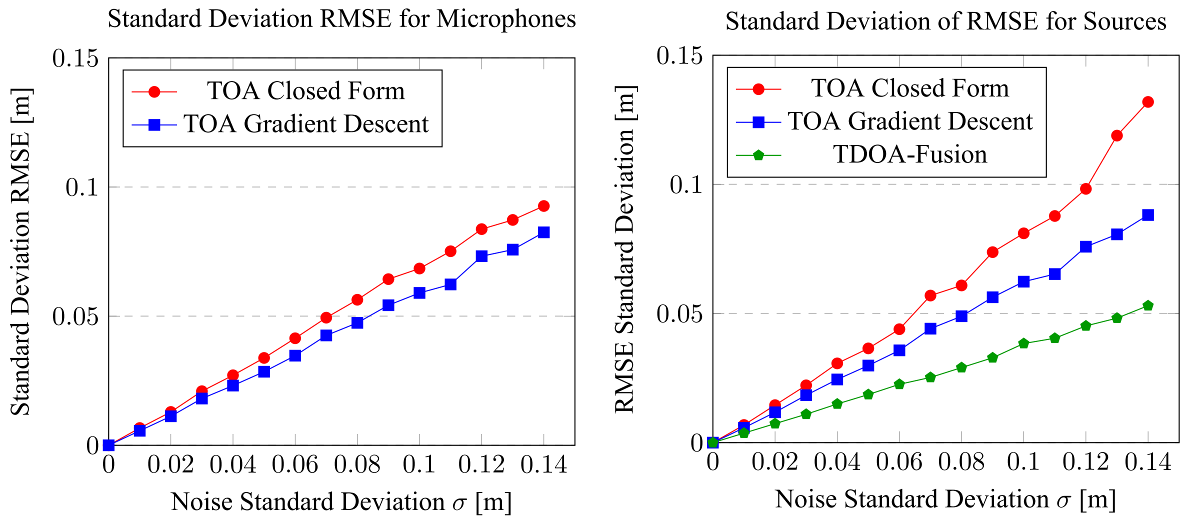

Figure B.1: Standard deviations of the RMSE's for an setup with an increasing amount of Gaussian noise on the arrival times.

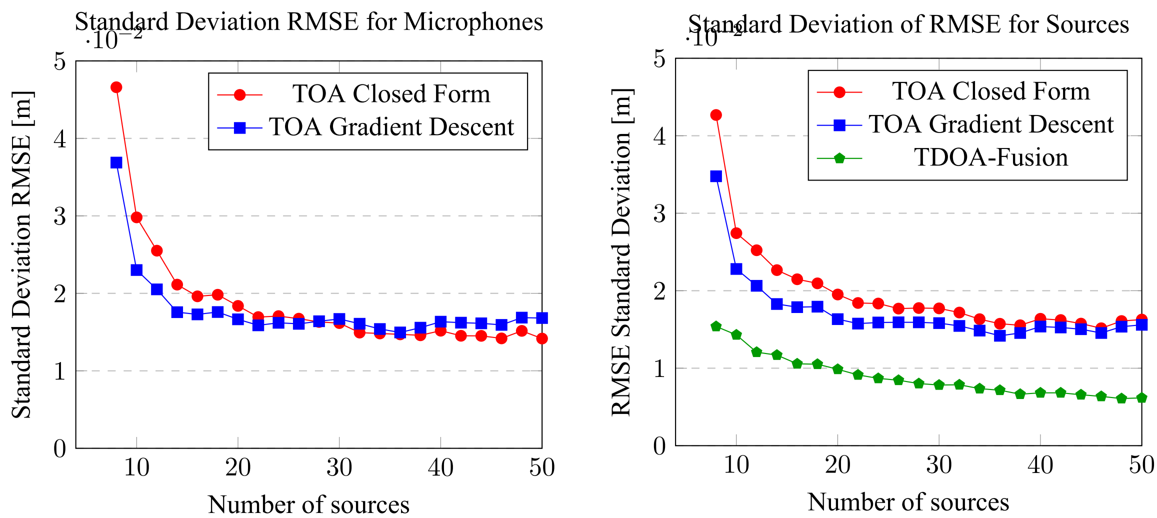

Figure B.2: For each value of the noise 500 trials were performed to obtain the standard deviation of the RMSE as a function of the arrival time noise standard deviation.

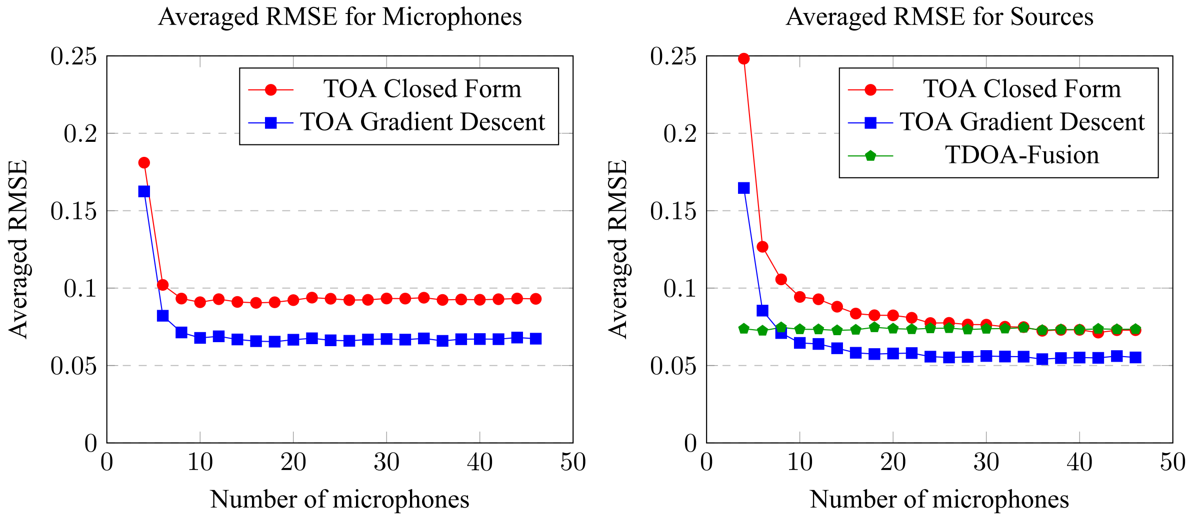

Figure B.3: For each value of the noise 500 trials were performed to obtain the standard deviation of the RMSE as a function of the number of microphones.

Bibliography

[1] G. Mao, B. Fidan, and B. D. Anderson, "Wireless sensor network localization techniques," *Computer Networks*, vol. 51, no. 10, pp. 2529 – 2553, 2007.

[2] N. Iliev and I. Paprotny, "Review and comparison of spatial localization methods for low-power wireless sensor networks," *IEEE Sensors Journal*, vol. 15, no. 10, pp. 5971–5987, 2015.

[3] R. C. Shit, S. Sharma, D. Puthal, and A. Y. Zomaya, "Location of things (lot): A review and taxonomy of sensors localization in iot infrastructure," *IEEE Communications Surveys & Tutorials*, 2018.

[4] A. Mackey and P. Spachos, "Performance evaluation of beacons for indoor localization in smart buildings," in *2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pp. 823–827, Nov 2017.

[5] C. Feng, W. S. A. Au, S. Valaee, and Z. Tan, "Received-signal-strength-based indoor positioning using compressive sensing," *IEEE Transactions on Mobile Computing*, vol. 11, pp. 1983–1993, Dec 2012.

[6] M. Crocco, A. Del Bue, and V. Murino, "A bilinear approach to the position self-calibration of multiple sensors," *IEEE Transactions on Signal Processing*, vol. 60, no. 2, pp. 660–673, 2012.

[7] L. Liu and K. K. Guntaka, "A non-iterative localization approach based on multi-dimensional scaling method for wireless sensor networks," in *Proceedings of the 11th IEEE International Conference on Networking, Sensing and Control*, pp. 328–333, April 2014.

[8] W. Yu, N. D. Gaubitch, and R. Heusdens, "Distributed tdoa-based indoor source localisation," *Unpublished paper*, 2018.

[9] M. Bshara, U. Orguner, F. Gustafsson, and L. V. Biesen, "Fingerprinting localization in wireless networks based on received-signal-strength measurements: A case study on wimax networks," *IEEE Transactions on Vehicular Technology*, vol. 59, pp. 283–294, Jan 2010.

[10] M. Parviainen, P. Pertilä, and M. S. Hämäläinen, "Self-localization of wireless acoustic sensors in meeting rooms," in *2014 4th Joint Workshop on Hands-free Speech Communication and Microphone Arrays (HSCMA)*, pp. 152–156, May 2014.

[11] P. Rong and M. L. Sichitiu, "Angle of arrival localization for wireless sensor networks," in *Sensor and Ad Hoc Communications and Networks, 2006. SECON'06. 2006 3rd Annual IEEE Communications Society on*, vol. 1, pp. 374–382, IEEE, 2006.

[12] A. Kulaib, R. Shubair, M. Al-Qutayri, and J. W. Ng, "An overview of localization techniques for wireless sensor networks," in *Innovations in Information Technology (IIT), 2011 International Conference on*, pp. 167–172, IEEE, 2011.

[13] Bosch, *DICENTIS Stem Microphone*, 8 2016. rev 5.

[14] A. Mesmoudi, M. Feham, and N. Labraoui, "Wireless sensor networks localization algorithms: a comprehensive survey," *arXiv preprint arXiv:1312.4082*, 2013.

[15] A. Ross, "Procrustes analysis," *Course report, Department of Computer Science and Engineering, University of South Carolina*, 2004.

[16] P. Danziger, "Big o notation," *Source internet: http://www. scs. ryerson. ca/~ mth110/Handouts/PD/bigO. pdf, Retrieve: April*, 2010.

[17] A. Plinge, F. Jacob, R. Haeb-Umbach, and G. A. Fink, "Acoustic microphone geometry calibration: An overview and experimental evaluation of state-of-the-art algorithms," *IEEE Signal Processing Magazine*, vol. 33, no. 4, pp. 14–29, 2016.

[18] Y. Kang, Q. Wang, J. Wang, and R. Chen, "A high-accuracy toa-based localization method without time synchronization in 3d space," *IEEE Transactions on Industrial Informatics*, 2018.

[19] R. T. Rajan and A.-J. van der Veen, "Joint ranging and synchronization for an anchorless network of mobile nodes," *IEEE Transactions on Signal Processing*, no. 8, pp. 1925–1940, 2015.

[20] A. Mesmoudi, M. Feham, and N. Labraoui, "Wireless sensor networks localization algorithms: a comprehensive survey," *arXiv preprint arXiv:1312.4082*, 2013.

[21] C.-D. Wann and H.-C. Chin, "Hybrid toa/rssi wireless location with unconstrained nonlinear optimization for indoor uwb channels," in *Wireless Communications and Networking Conference, 2007. WCNC 2007. IEEE*, pp. 3940–3945, IEEE, 2007.

[22] M. Pollefeys and D. Nister, "Direct computation of sound and microphone locations from time-difference-of-arrival data," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pp. 2445–2448, IEEE, 2008.

[23] A. H. Sayed, A. Tarighat, and N. Khajehnouri, "Network-based wireless location: challenges faced in developing techniques for accurate wireless location information," *IEEE signal processing magazine*, vol. 22, no. 4, pp. 24–40, 2005.

[24] I. Borg and P. Groenen, *Modern Multidimensional Scaling: Theory and Applications*. Springer, 2005.

[25] Y. Shang and W. Ruml, "Improved mds-based localization," in *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, vol. 4, pp. 2640–2651, IEEE, 2004.

[26] G. H. Golub, A. Hoffman, and G. W. Stewart, "A generalization of the eckart-young-mirsky matrix approximation theorem," *Linear Algebra and its applications*, vol. 88, pp. 317–327, 1987.