Dissertation zur Erlangung des Doktorgrades der Technischen Fakultät der Albert-Ludwigs-Universität Freiburg im Breisgau

Calibration-Free Localization using Time Differences of Arrival

Institut für Informatik
Technische Fakultät
Albert-Ludwigs-Universität Freiburg

Johannes Wendeberg

Oktober 2013

Calibration-Free Localization using Time Differences of Arrival

Johannes Wendeberg

Dissertation zur Erlangung des Doktorgrades der Technischen Fakultät der Albert-Ludwigs-Universität Freiburg im Breisgau

Dekan:	Prof. Dr. Yiannos Manoli
Erstgutachter:	Prof. Dr. Christian Schindelhauer (Universität Freiburg)
Zweitgutachter:	Prof. Dr. Kalle Åström (Universität Lund, Schweden)

Datum der Disputation: 14. Oktober 2013

Zusammenfassung (deutsch)

Die zunehmende Verbreitung mobiler Technologie im Alltag hat zu einer wachsenden Nachfrage nach Lokalisierungsdiensten bei Handheld-Anwendungen und nach Technologien zur Positionsbestimmung autonomer Systeme geführt. In jüngster Zeit ist eine Verschiebung des Interesses in Richtung Lokalisierung innerhalb von Gebäuden zu beobachten. Desweiteren schafft die zunehmende Verbreitung autonomer intelligenter Systeme in Bereichen, die lange Zeit eine Domäne manueller Arbeitsabläufe waren, Nachfrage nach präzisen Ortsinformationen auch im Innenbereich. Globale Navigationssatellitensysteme wie zum Beispiel GPS sind in diesen Szenarien zumeist nicht verfügbar. Alternativen wie WLAN-Lokalisierung konnten sich bisher aufgrund mangelnder Präzision nicht durchsetzen, und optische Systeme auf Basis von Laserscannern oder Kamerasystemen sind präzise, aber auch sehr teuer, und umständlich einzurichten. Dagegen ist die Hardware für Navigation mittels Schall und Ultraschall kostengünstig, und kann präzise Ortsinformation im Bereich von Zentimetern bieten.

Lokalisierung mittels Time Differences of Arrival (TDoA) beruht auf der Ausbreitungszeit solch eines Signals. Bei herkömmlicher hyperbolischer TDoA-Lokalisierung sind die Referenzpositionen im Voraus bekannt, also entweder Sender oder Empfänger, und die Position eines Ziels ist zu bestimmen. Manuelle Vermessung der Referenzen erreicht man durch ein Maßband, Laser-Distanzmessung, mittels geodätischer Methoden oder durch externe Infrastruktur, beispielsweise GPS. In einigen Fällen ist ein Ortungssystem wünschenswert, das unabhängig von manueller Installation und externen Systemen ist. Gründe hierfür sind die eingeschränkte Flexibilität eines externen Systems, und erhöhte Kosten und Energieverbrauch durch eine externe Lösung.

In der *kalibrierungsfreien TDoA-Lokalisierung* werden Referenzpositionen direkt während des Lokalisierungsprozesses berechnet, wodurch die quälende Notwendigkeit wegfällt, die Positionen der Referenzen im Voraus zu kennen. In dieser Arbeit stellen wir mehrere neue Ansätze zur kalibrierungsfreien TDoA-Lokalisierung vor. Wir betrachten das Problem aus mehrerlei Gesichtspunkten, wofür wir Algorithmen in vier Bereichen entwickelt haben – die Fernfeldannahme, lokale Optimierung, Branch-and-Bound-Algorithmen und probabilistische Zustandsschätzung.

Die Annahme, dass Signale von weit entfernten Orten stammen, die sogenannte *Fernfeldannahme*, vereinfacht das Gleichungssystem und ermöglicht schnelle und stabile Algorithmen in geschlossener Form. Wir stellen die Ellipsoid-TDoA-Methode vor, die darauf beruht, dass TDoA-Messungen von drei Empfängern in der Ebene eine Ellipse bilden. Die Parameter der Ellipsengleichung lassen sich fehlertolerant durch Regression bestimmen, woraus sich Abstände und Winkel zwischen den Empfängern ableiten lassen. Die Ellipsoid-TDoA-Methode ist der erste Algorithmus, der das minimale Problem in der TDoA-Fernfeldannahme ohne Erfordernis von Synchronisation zwischen Empfängern löst. Wir demonstrieren die Robustheit in Simulation und Experimenten und zeigen, dass die Ellipsenmethode auch dann noch stabil ist, wenn die Fernfeldannahme teilweise verletzt wird.

Approximative Algorithmen unter der Fernfeldannahme sind in der Allgemeingültigkeit begrenzt, vor allem dann, wenn die Zahl der Messungen gering ist. Wir betrachten den Bereich der lokalen nichtlinearen Optimierung als Herangehensweise an den allgemeinen Fall der TDoA-Lokalisierung, wobei wir den Fokus auf die Fehlerfälle setzen. Aufgrund der hohen Dimensionalität der kalibrierungsfreien TDoA-Lokalisierung tendieren iterative Optimierungsalgorithmen dazu, das globale Optimum der Fehlerfunktion nicht zu finden, welches die einzige richtige Lösung darstellt. Wir stellen den Cone-Alignment-Algorithmus vor, einen iterativen Algorithmus auf Basis einer Masse-Feder-Simulation, in der Sender- und Empfängerpositionen durch physikalische Partikel repräsentiert werden. Eine TDoA-Fehlerfunktion wird durch physikalische Federn zwischen den Partikeln modelliert. Mittels Relaxierung der Federn wird die Fehlerfunktion minimiert, wobei die Partikel an Schwung gewinnen und so lokale Minima überwinden. In umfangreichen Simulationen analysieren wir den Algorithmus und vergleichen ihn mit den Standardalgorithmen Gradientenabstieg, dem Gauss-Newton-Verfahren und dem Levenberg-Marquardt-Algorithmus. Wir zeigen, dass der Cone-Alignment-Algorithmus die richtige Lösung häufiger findet als die Standardalgorithmen.

Um das Problem der Vollständigkeit in der Optimierung zu lösen, stellen wir einen Branch-and-Bound-Algorithmus in Polynomialzeit vor, der beweisbar alle Lösungen für kalibrierungsfreie TDoA-Probleme bis zu einer Fehlerschranke ϵ approximiert. Der Algorithmus basiert auf der Unterteilung eines fünfdimensionalen Suchraumes, mit dem der Minimalfall von vier Empfängern in der Ebene dargestellt wird, in Teilräume, und Prüfung jedes einzelnen Teilraums darauf, ob er eine mögliche Erklärung der TDoA-Messungen mit Unsicherheit ϵ darstellt. In der praktischen Umsetzung zeigen wir, dass der Algorithmus asymptotisch schneller als die Aufzählung aller Zellen der Größe ϵ ist. Der Branch-and-Bound-Algorithmus ist unseres Wissens die erste theoretische Lösung des allgemeinen Minimalfalls der TDoA-Selbstlokalisierung.

Wir präsentieren außerdem das kalibrierungsfreie Ultraschall-Ortungssystem, das im Rahmen dieser Dissertation entwickelt wurde. In dem System emittiert ein mobiler Ultraschallsender kurze diskrete Impulse, die von mehreren Ultraschallempfängern empfangen werden, wodurch der Sender lokalisiert wird. Die Lokalisierung im System ist kalibrierungsfrei, wofür wir einen Algorithmus zur probabilistischen Zustandsschätzung entwickelt haben. Dieser basiert auf einem Partikelfilter, der sowohl den mobilen Sender, als auch die Positionen der Empfänger schätzt. Der Algorithmus verwendet ein probabilistisches Sensormodell für TDoA-Daten, das explizit die Messunsicherheit berücksichtigt, wodurch er robust gegen Mess-Ausreißer wird, welche eine typische Gefahr in der TDoA-Lokalisierung darstellen. Wir haben die Robustheit des Ultraschall-Ortungssystems in umfangreichen Experimenten überprüft, wo wir zeigen, dass der probabilistische Ansatz genaue Positionsschätzungen für den Sender und die Empfänger selbst bei Mess-Ausreißern gewährleistet, wobei das System mittlere Positionsfehler von unter 5 Zentimetern erreicht.

Abstract

The continuous rise of mobile technology in everyday life has led to an increasing demand for location-based services for handheld applications and autonomous systems. In recent time, a shift of interest towards indoor localization can be observed where global navigation satellite systems, such as the Global Positioning System, are mostly not available. The hardware for positioning and navigation based on sound or ultrasound is affordable and precise in the range of centimeters. Localization based on the *time differences of arrival* (TDoA) relies on the propagation time of such a signal. In conventional hyperbolic TDoA localization the reference positions, i.e. the positions of either receivers or senders of a signal, are known a priori, and the position of a target is subject to locate.

In *calibration-free TDoA localization*, reference positions are calculated during the process of locating the target – or just the references are calculated from unknown signals, eliminating the agonizing need to determine the positions of references by hand. In this thesis, we propose several novel approaches to the domain of calibration-free TDoA localization. Addressing the problem from several points of view, we propose algorithms in four fields – the far-field assumption, local optimization, branch-and-bound algorithms, and probabilistic state estimation.

The assumption that signals originate from remote locations, the so-called *far-field assumption*, simplifies the equation system and allows for fast and robust closed-form algorithms. We propose the Ellipsoid TDoA method, which relies on the fact that TDoA measurements from three receivers in the plane form an ellipse. This ellipse is robustly determined by regression, and the distances and angles between the receivers are calculated from the parameters of the ellipse. The Ellipsoid TDoA method is the first algorithm that solves the minimum problem of the TDoA far-field assumption, requiring no synchronization between receivers. We demonstrate the robustness of the algorithm in simulation and in experiments, where we show that the Ellipsoid method is still reliable when the far-field assumption is violated to some extent.

Far-field algorithms are limited in generality, especially when measurements are rare. We consider the field of non-linear local optimization where we set focus to the failure cases. Due to the high dimensionality of the calibration-free TDoA problem, iterative optimization tends to not find the global optimum, which is the only acceptable solution. We propose the Cone Alignment algorithm, an iterative mass-spring simulation where signal and receiver positions are represented by physical particles which gather momentum to overcome local minima. In numerical simulations we compare the algorithm to standard optimization approaches, where we demonstrate the superiority of Cone Alignment in finding the solution.

To address this intrinsic problem of local optimization, also denoted as the problem of *completeness*, we propose a polynomial time branch-and-bound algorithm that is a proof to enumerate all solutions of calibration-free TDoA up to an error bound ϵ. The algorithm is based on subdivision of a five-dimensional search space into subspaces, by

which the minimum problem of four receivers in the plane is represented, and test of each subspace for being a possible explanation of the observed TDoA measurements, given uncertainty ϵ. In practical implementation we demonstrate that the algorithm is asymptotically faster than enumerating all cells of size ϵ, which is the brute-force approach. The branch-and-bound algorithm is to our knowledge the first theoretical solution to the general minimum case of calibration-free TDoA.

Furthermore, we present the calibration-free ultrasound localization system, where a mobile ultrasound sender is located by emitting short pulses that are received by several ultrasound receivers. We have created a probabilistic state estimation algorithm based on a particle filter that estimates both the mobile sender and the positions of receivers. The algorithm is robust against measurement outliers, which is a common pitfall in TDoA localization. We have verified the robustness of the ultrasound localization system in extensive experiments, where we demonstrate that our approach provides accurate position estimates with errors below 5 cm even in case of measurement outliers of large magnitude.

Acknowledgments

First of all, I wish to thank my supervisor Christian Schindelhauer who gave me the opportunity to do my doctorate at his chair. He was a great advisor, and he granted me a large degree of freedom, while he guided me in scientific methodology and taught me to question scientific conventions with academic rigor. Both, our shared opinions about scientific research and our opposing points of view were a great inspiration for my work.

I would like to thank my cooperation partners who have a large share to the achievements in this thesis. Particularly, my gratitude goes to Fabian Höflinger, who developed the exceptional hardware of the ultrasound localization system. Also, I wish to thank the fellows and students who helped in creation of the system and in our experiments, especially Manuel Bührer, Joachim Hoppe, and Thomas Janson who provided valuable assistance in software and hardware development. I would like to thank Jörg Müller for the fruitful collaboration with the probabilistic algorithm, and Simon Burgess for the inspiring teamwork in the TDoA far-field assumption.

My special thanks go to my co-authors Kalle Åström, Amir Bannoura, Joan Bordoy, Wolfram Burgard, Simon Burgess, Manuel Bührer, Thomas Janson, Joachim Hoppe, Patrick Hornecker, Fabian Höflinger, Yubin Kuang, Zvi Lotker, Jörg Müller, Leonhard Reindl, Christian Schindelhauer, and Rui Zhang for the critical discussion and valuable help in scientific writing. I wish to thank the members of the Research Training Group for fascinating discussions and workshops, and Manuela Kniß, Norbert Küchlin, and Kerstin Pfeiffer for help in administrative matters.

My thanks go to Thomas Janson, Christian Ortolf, and Arne Vater for razor-sharp discussions, but also for being great workmates, and for the support and pleasurable environment. I also thank the fellows who helped to proof-read this thesis. Last, I wish to thank my family for their unconditional support and encouragement in all stages of my life.

This work has partly been supported by the German Research Foundation (Deutsche Forschungsgemeinschaft, DFG) within the Research Training Group 1103 (Embedded Microsystems).

Contents

1 Introduction

The continuous rise and ubiquitous availability of mobile technology in everyday life has led to an increasing demand for location-based services for handheld applications and location-awareness of autonomous systems. Modern mobile applications amalgamate database information, locality information, and user feedback, making end-user handheld devices a context-sensitive multi-purpose assistant. Location information for these applications is usually generated by global navigation satellite systems, in particular the Global Positioning System (GPS). Together with fingerprinting techniques that rely on a database of known Wi-Fi access points, location precision in the range of 10 meters is achieved, even in urban areas.

In recent time, a shift of interest towards indoor localization can be observed. Increasing automatization with autonomous intelligent systems in fields, that had been the domain of manual operation for a long time, have created a demand for precise locality information in indoor areas. GPS is mostly not available in these scenarios. Alternatives such as Wi-Fi ranging could not prevail to date, probably due to the low precision, which is not adequate for many applications. Optical systems based on laser scanners or on camera systems are very precise, but also very expensive. In contrast, the hardware for positioning and navigation based on sound or ultrasound is affordable, and due to the low signal velocity, time-of-arrival based localization can provide precise locality information in the range of centimeters.

1.1 Multilateration based on the Runtime of Signals

Localization based on the time-of-arrival (ToA) uses the propagation time of a signal to calculate the absolute or relative distance between a sender and a receiver of a signal. The approach relies on the characteristics of wave propagation, where the velocity is assumed constant. Examples are the speed of sound, which is approximately $c_{\text{sound}} = 343\,\text{m/s}$ at room temperature, or the speed of light.

If the sender and the receiver share a common time base, therefore their clocks are synchronized, the sender just records the time when a signal is emitted, and the receiver records the time when the emitted signal arrives. The absolute distance can be calculated according to the signal velocity and the time the signal needs to travel from the sender to the receiver. Alternatively, signals of different velocities are sent to a target simultaneously, for instance a radio signal and a sound, which arrive in sequence at different times. The distance is calculated by the time delay, given both velocities are known. Assuming that one signal velocity is very large, as in the case of radio

signals, the equation may be simplified to using the time differences of both signals multiplied by the slower velocity. In an implementation of this technique, a mobile unit is equipped with an additional radio transceiver, receiving an acoustic signal, as for example in the Cricket system which uses Wi-Fi and ultrasound [1].

In another example, one signal is sent by a transceiver to a second transceiver, which responds immediately, so the answer arrives at the first transceiver. The first transceiver can calculate the distance, which is half the round trip time, multiplied by the signal velocity. When the transceivers also listen to their own signals, hardware delays can be eliminated, as in BeepBeep [2], an acoustic localization system for smart phones. The ToA *multilateration* problem is then a problem of intersecting circles, or spheres in three dimensions, which may be solved in closed form or with iterative methods [3]. Multidimensional scaling, a formal representation of the multilateration problem, is described in Section 3.4.

In contrast, in *hyperbolic multilateration* no cooperative action between the sender and the receiver of a signal is taken. The signal is not sent back and forth, and the receiver cannot acknowledge the reception to the sender, neither are their clocks synchronized. However, clock synchronization is allowed between receivers. By sending a signal to multiple receivers, where the times of arrival are detected, the time differences of arrival (TDoA) can be calculated. For example, in case of a microphone array that is connected to a multi-channel audio interface, synchronization of receivers is implicit. Instead of absolute distances, only relative distance differences are measured, forming hyperbolic equations, therefore the name, *hyperbolic* multilateration.

In common literature the term *time of arrival* (ToA) occurs as a generic term for positioning using the propagation time of signals, and specifically to denote the "time of flight", i.e. the measurement of absolute distance from a sender to a receiver. We use the term *time differences of arrival* (TDoA) as a subset of ToA to denote hyperbolic localization given only relative distance measures.

While our focus in this thesis is algorithms for TDoA localization, the problem of TDoA localization is multi-layered. It may be structured by *horizontal* and *vertical* criteria. Horizontal are the assumptions and prerequisites on the scenarios to localize, for example the assumption of distant signal origins, and the different algorithms to calculate Cartesian coordinates of sensors and signals from TDoA measurements, which differ in versatility and complexity of calculation.

Vertical are the different levels of TDoA localization. At a lower level are the calculation of time differences from a recorded audio, ultrasound, or radio signal, either simple amplitude detection, by pattern recognition, or by cross-correlation. In case of detection of discrete timestamps, this includes the assembly of time points to signal entities. The next level is calculation of positions from the time differences, which may be split into calculation of distances and generating coordinates from these. On a high level may be filtering algorithms and plausibility checking of positions.

The problem of TDoA is more difficult than the problem of absolute distances (ToA), as the absolute send time is added as a variable. The higher number of variables

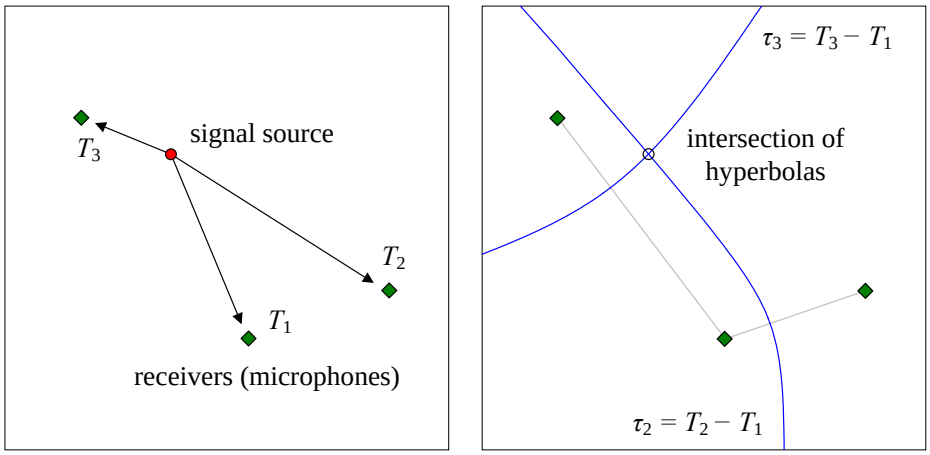


Figure 1.1: *Left*: A signal is emitted at an unknown time, arriving at three receivers at times T_i that can be measured. *Right*: The measurements describe a hyperbola of equal time differences between pairs of receivers. The intersection of two hyperbolas is the actual origin of the signal.

increases the number of required measurements to obtain a finite number of solutions, and it complicates the process of finding these solutions. For instance, given two different receivers in the plane, a distance measurement to each of the receivers is sufficient to narrow down the possible signal origins to two positions. From there, the correct position may be found by plausibility considerations. In contrast, if the time of signal emission is not known, and therefore only relative distances can be observed, the minimum number of receivers increases to three, which results in two intersecting hyperbolas, as illustrated in Fig. 1.1. The minimum number of required measurements for TDoA is analyzed in [4, 5] and in Section 2.4.

In a variant of the TDoA problem, multiple synchronized senders emit signals that are obtained by a receiver which is not synchronized with the senders. If the emission time of the signal is known, the receiver can calculate the time differences, yielding a system of hyperbolic equations. In case of the Global Positioning System the satellites are synchronized by means of high precision clocks. They send the current time encoded into the signal to GPS receivers, which calculate TDoA of the signals by correlation. If the GPS receivers are equipped with a high precision clock then absolute distances to at least three satellites can be measured, however in practice this is a rare case. They measure “pseudoranges” to the satellites, which leads to a hyperbolic system of distance differences. A fourth satellite is required to calculate three position coordinates and the time. Additional satellites increase the reliability and precision of positioning, as over-determined measurements compensate for error. Note that GPS positioning is by far more complicated in real implementation, as high velocity of the satellites, atmospheric disturbances of the signal velocity, and multipath propagation of the signal need to be incorporated into the measurement process [6].

1.2 Calibration-Free TDoA

In classic hyperbolic TDoA localization the reference positions, i.e. the positions of either the receivers or the senders are known a priori, and the positions of senders are subject to locate, or the receiver, respectively. In some publications is suggested to compensate for given erroneous reference positions [7, 8], but in general the positions of reference receivers or signal emitters are precisely determined by external means in an external coordinate system. Manual measurement of receiver positions may be done by a measuring tape, a laser range finder, a grid arrangement or specifically structured sensor array, or by geodetic methods. Alternatively, references are generated by automatic measurements of an external localization infrastructure, such as optical motion tracking, or the Global Positioning System. In case of the Global Positioning System, still the *ephemerides* of the satellites, i.e. the lookup tables of position, are calculated in advance and transmitted to the GPS receiver during the measurement process [6].

In some cases, none of these external means of calibration are feasible, and a localization system is desired where a target can be localized independently of manual effort and of external infrastructures. Reasons may be included by the following:

- **Precision.** External measurement might not be sufficiently accurate. For example, measurement by hand is limited to a precision of several centimeters, GPS is usually limited to meters.
- **Availability.** Some external measurement systems may not be available in certain environments. For instance, GPS cannot be used indoors or under water. Infrastructural reference systems may be damaged or destroyed in a crisis or a natural disaster.
- **Costs.** External reference systems may increase the cost of a localization system in terms of money or energy consumption.
- **Dependence.** For a localization system the independence from an external infrastructure might be desirable for economical or political reasons, or from the perspective of mobility.

On grounds of these reasons, efforts are required to enable hyperbolic localization where the reference positions are not essential to know in advance. Instead, in *calibration-free*, or *anchor-free TDoA localization*, the reference positions are calculated during the process of locating the target – or just the references are calculated from unknown signals – eliminating the need to manually determine the positions of references by external means.

The problem of calibration-free TDoA is challenging. First, the number of unknown variables to be determined is even higher than in conventional TDoA, as also the reference positions need to be calculated. Second, the signal and receiver positions depend on each other, which is adverse to the robustness of the calculation. For instance, one failed time measurement, a large outlier, affects the calculation of receiver

positions, which again affect the calculation of other signal positions. In contrast, in conventional TDoA a failed measurement has only influence on a single signal position. Third, due to the high dimensionality of the problem, many strategies to linearize the problem [9, 10] cannot be applied in general, so iterative approaches are required, which are time-consuming and require careful initialization. However, we believe that the benefits of calibration-free TDoA are worth the challenge.

We contribute in this thesis to the field of calibration-free TDoA localization. Based on the fundament of different assumptions, and for the purpose of different applications, we have developed several approaches addressing the problem of calibration-free TDoA. In general, the approaches described in the following are applied for calibration of unknown receiver positions by using signals from unknown origins. Yet, the algorithms can be adapted to calibrate the positions of signal beacons instead. The difference is rather a technical one, where the underlying mathematical problem is similar. When actually the opposite case is considered, we will specifically point it out in the text.

1.3 Related Work

We present a short digest of selected publications in the field of localization and important contributions in the field of calibration-free TDoA localization.

Popular infrastructure-based approaches for indoor and outdoor applications are GSM localization [11, 12, 13] and Wi-Fi network fingerprinting [14]. The interpretation of the received signal strength indication (RSSI) is a usual approach [15, 16]. When RSSI or ToA/ToF ("time of flight") data is available the problem is reduced to distance vectors, which is solved by linear estimators [3], by multi-dimensional scaling [17, 18, 19], by the iterative Gauss-Newton method [20], by iterative construction [21], or by semidefinite programming [22, 23]. Force-directed approaches are an alternative for relaxing distance constraints in large-scale networks [24, 25] and in the Vivaldi network coordinate system [26].

In most "conventional" TDoA localization approaches the positions of receivers are given a priori. Then, estimating a sender's position using TDoA can be addressed in closed form equations [7, 9, 10, 27, 28, 29, 30] or by iterative approaches [31, 32, 33]. A survey of closed-form TDoA estimators is given in [34]. Sometimes the positions of senders are given, and the positions of receivers are located instead [35]. Calculation of TDoA usually relies on the correlation of signals [36, 37, 38].

Moses et al. were among the first to mention TDoA self-localization in a sensor array. In [39] they describe an approach to use TDoA with additional directivity information, the Angles of Arrival (AOA), to locate unknown sender and receiver positions in a planar world. This would require expensive receiver arrays or directed receivers, especially for an extension to three dimensions. In [40] the authors describe an experiment of self-localization of an array of sensors by TDoA from unknown signal positions.

However, no further description of how the signals and receivers are calculated is given, nor of the failure cases of the experiment. Without new algorithms, but a valuable theoretical contribution, is the PhD thesis of Henrik Stewénius [4], who analyzes the minimum cases of the calibration-free localization problem for absolute and relative travel times.

One of the first approaches regarding specifically the problem of calibration-free TDoA calibration was presented by Biswas and Thrun. In [41] they describe a probabilistic model based on a Bayesian network to calculate sensor and signal positions simultaneously without prior knowledge or assumptions on the positions from nothing but TDoA measurements. The constraints in the Bayesian network are optimized by gradient descent. Although the results in the paper are adequate in terms of feasibility with real hardware, the mathematical pitfalls of the problem are not followed up. The approach appears to fail in some cases due to an "ill-conditioned" setting, or remain stuck in a local minimum of the error function. Besides, the approach is computationally expensive because of the iterative gradient descent algorithm.

Some time later an elegant approach was presented by Pollefeys and Nister [42]. By factorization of a constraint matrix the signal emission times can be calculated in closed form, effectively reducing the TDoA problem to a problem of absolute distances in a bipartite graph, cf. Section 3.4. In our opinion, they are closest to a direct solution of calibration-free TDoA. However, their approach is still remote from the minimum case, as twice as many receivers are required. Furthermore, the algorithm is very prone to measurement errors of the signal runtime. Recently, Kuang and Åström presented a factorization approach [43] that reduces the required number of senders and receivers to only one higher than the minimum case.

Furthermore, several approaches have been published that state assumptions about positions of signals to calculate the positions of receivers. In an early approach presented by Thrun [44], the signals were assumed to originate from a large distance. Then the positions of synchronized receivers can be calculated in closed form. Simultaneously to our work on the far-field problem, Burgess and Kuang presented an elegant factorization approach [45, 46], however their solution does not consider the minimum problem. In a weaker assumption, the distance between receivers can be determined by statistical analysis of the signal distribution, from which the coordinates of receivers can be calculated, which was presented by Pertila et al. [38], Biswas and Thrun [47] and in the PANDAA system [48]. In some recent approaches the receivers are assumed to reside on a line [49, 50], allowing for a solution in closed form.

1.4 Applications of Calibration-Free TDoA

As the receivers do not need to interact with the sound source, a large variety of types of signals can be used. These signals can be natural, such as clicking noises or finger snaps, coughing, or footstep sounds from the heel of a shoe. Such peaked sounds are easily detectable by receiver devices by the edge of their amplitude, either by signal processing

of a recorded audio stream, as described in Section 7.1, or by a simple hardware tone detector circuit as in the Crossbow MICA2/MICAz sensor nodes[1]. Signal sources can be continuous, such as the human voice, or music. Then, the time difference is calculated by cross-correlation of the signal of different receivers. As the signal must be available to every sensor for such a direct comparison, a typical scenario might be an array of microphones connected to an audio interface.

Applications of anchor-free TDoA are manifold. First, TDoA algorithms that do not require prior references are capable to localize nodes in a sensor network – just by unknown signals from the environment. In this way, calibration-free TDoA self-localization is completely passive, requiring no active signaling, except for probably the communication for synchronization and exchange of time points. By this means, wireless sensor nodes may be localized, to improve efficiency of network routing, or to provide location support for experiments. Or a microphone array for concert recordings could be calibrated – just from the recorded music. As soon as the array is calibrated, individual instruments can be located if they can be distinguished by their characteristics.

One application may be teacher's support in a computer lab. In a programming course students use their laptops to log into an interactive teaching assistance system, which provides support for task assignment and feedback between students and their teacher. Configuration-free self-localization might provide a benefit by enabling the teacher to see the positions of the students' laptops on his screen. By recording just the noises of the people of the class, with the built-in microphones of the laptops the students' positions may be calculated, even though they are randomly seated at the beginning of every lesson.

Another practical use is a *calibration-free tracking system*. Using TDoA localization a synthetic signal beacon could be tracked. As no control over the sender is required, the sender can be simple, inexpensive, and lightweight. In case of a sound or ultrasound emitter, no additional radio connection is required. By using calibration-free TDoA algorithms, the need for initial calibration of the receiver positions, which is a cumbersome task, can be avoided. In this way, using calibration-free TDoA, a "plug-and-play" localization system may be created. We have created such a system using an ultrasound emitter and receivers, where we can track the ultrasound target at a precision of centimeters. See Section 7.3 for an introduction of the ultrasound localization system.

The "reverse" problem of calibration-free TDoA is considered in Chapter 9. Here, the application of an calibration-free ultrasound tracking system is discussed, where a moving *receiver* is tracked by means of stationary senders, with the benefit of no limitation of the number of tracked devices. An industrial project for development of such a system for application in industrial intralogistics has begun in the fall of 2012 with the Chair for Computer Networks and Telematics participating.

[1] Moog Crossbow: `http://www.xbow.com`

1.5 Publications

Parts of this thesis have been published in peer-reviewed journals and in proceedings of peer-reviewed conferences. In the following find a list of relevant publications of the author, listed in reverse chronological order.

- Joan Bordoy, Patrick Hornecker, Fabian Höflinger, Johannes Wendeberg, Rui Zhang, Christian Schindelhauer, and Leonhard Reindl. *Robust Tracking of a Mobile Receiver using Unsynchronized Time Differences of Arrival.* In Proceedings of 2013 International Conference on Indoor Positioning and Indoor Navigation (IPIN), 2013.
- Simon Burgess, Yubin Kuang, Johannes Wendeberg, Kalle Åström, and Christian Schindelhauer. *Minimal Solvers for Unsynchronized TDOA Sensor Network Calibration using Far-Field Approximation.* In Proceedings of 9th International Symposium on Algorithms and Experiments for Sensor Systems, Wireless Networks and Distributed Robotics (ALGOSENSORS), 2013.
- Johannes Wendeberg, Fabian Höflinger, Christian Schindelhauer, and Leonhard Reindl. *Calibration-Free TDOA Self-Localization.* Journal of Location Based Services, 2013.
- Johannes Wendeberg, Jörg Müller, Christian Schindelhauer, and Wolfram Burgard. *Robust Tracking of a Mobile Beacon using Time Differences of Arrival with Simultaneous Calibration of Receiver Positions.* In Proceedings of 2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN), 2012.
- Fabian Höflinger, Johannes Wendeberg, Rui Zhang, Manuel Bührer, Joachim Hoppe, Amir Bannoura, Leonhard Reindl, and Christian Schindelhauer. *Acoustic Self-calibrating System for Indoor Smartphone Tracking (ASSIST).* In Proceedings of 2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN), 2012.
- Johannes Wendeberg and Christian Schindelhauer. *Polynomial Time Approximation Algorithms for Localization based on Unknown Signals.* In Proceedings of 8th International Symposium on Algorithms for Sensor Systems, Wireless Ad Hoc Networks and Autonomous Mobile Entities (ALGOSENSORS), 2012.
- Johannes Wendeberg, Thomas Janson, and Christian Schindelhauer. *Self-Localization based on Ambient Signals.* Journal of Theoretical Computer Science, vol. 453, pp. 98–109, 2012.
- Johannes Wendeberg, Fabian Höflinger, Christian Schindelhauer, and Leonhard Reindl. *Anchor-free TDOA Self-Localization.* In Proceedings of 2011 International Conference on Indoor Positioning and Indoor Navigation (IPIN), 2011.

- Christian Schindelhauer, Zvi Lotker, and Johannes Wendeberg. *Network Synchronization and Localization Based on Stolen Signals.* In Proceedings of 18th International Colloquium on Structural Information and Communication Complexity (SIROCCO), 2011.
- Christian Schindelhauer, Zvi Lotker, and Johannes Wendeberg. *Brief Announcement: Network Synchronization and Localization Based on Stolen Signals.* In Proceedings of 30th Annual ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing (PODC), 2011.
- Thomas Janson, Christian Schindelhauer, and Johannes Wendeberg. *Self-Localization Application for iPhone using only Ambient Sound Signals.* In Proceedings of 2010 International Conference on Indoor Positioning and Indoor Navigation (IPIN), 2010.
- Thomas Janson, Christian Schindelhauer, and Johannes Wendeberg. *Self-Localization Based on Ambient Signals.* In Proceedings of 6th International Workshop on Algorithms for Sensor Systems, Wireless Ad Hoc Networks and Autonomous Mobile Entities (ALGOSENSORS), 2010.

The relevant publications for a specific topic are referenced in the respective chapters and listed in the main bibliography. The publications and achievements in this thesis have been created in scientific cooperation with others. The important collaborations are listed in the following.

1.6 Collaborations

Parts of this thesis have been developed in cooperation with others. The probabilistic state estimation approach described in Chapter 6 was created in cooperation with Jörg Müller, who contributed significantly to the mathematical formulation of the motion and sensor model.

The localization framework was created with valuable assistance of Thomas Janson, who contributed by developing the aggregation algorithm and porting the software to Apple iPhone. The software framework is described in Section 7.1, the experiments with the iPhone devices in Section 7.2.

The ultrasound tracking system, which has been developed during this thesis and is presented in Section 7.3, was created in very close collaboration with Fabian Höflinger, who contributed substantially by developing the hardware foundation of the localization system. This includes the development of USB-connected ultrasound receivers, the hardware integration of stand-alone embedded receiver devices, and the development of a portable ultrasound sender.

Furthermore, Wolfram Burgard provided helpful comments about the theoretical aspects of TDoA localization. The initiation of the industrial eCULTS project, which is presented in Chapter 9, benefited from valuable support and suggestions of Heinrich Hippenmeyer, Fabian Höflinger, and Leonhard Reindl.

1.7 Notation

The following table summarizes the notation used throughout this thesis.

Symbol	Denotation
a, b, A, B	scalar variables, sets
$\mathbf{a}$, $\mathbf{b}$	column vectors
$\mathbf{A}$, $\mathbf{B}$	matrices
$(\cdot)^T$	transpose of a vector or matrix
$\lvert \cdot \rvert$	absolute value of a scalar, the complex magnitude, the determinant of a matrix
$\lVert \cdot \rVert$	the Euclidean norm
$\{\ldots\}$	a set of elements
$(\ldots)$	a tuple of elements, a vector
$[\ldots]$	a matrix of elements
$[a, b]$, (a, b)	a closed/open interval
$p(x)$	a probability density function
$p(x \mid y)$	a conditional probability density function
$\mathcal{N}(x, \mu, \sigma^2)$	the Gaussian density function
$\mathcal{O}(\cdot)$	asymptotic upper bound of complexity of an algorithm (Landau notation)

The following table summarizes important variables used throughout this thesis.

Variable	Meaning
c	the velocity of a signal
p	the dimensionality of space
$\mathbf{s}$	a signal emitter position (sender)
$\mathbf{m}$	a receiver position
n	the number of receivers
m	the number of signal sources
t	the emission time of a signal
T	the reception time of a signal
τ	a time difference (TDoA)
δ	a synchronization offset
$\mathbf{x}$	a state vector, the position of a physical particle
$\mathbf{v}$	the velocity of a physical particle
$\mathbf{f}$	the force of a physical particle
$\mathbf{z}$	an observation
$\mu, \boldsymbol{\mu}$	the mean of a distribution
$\sigma^2, \boldsymbol{\Sigma}$	the (co-)variance of a distribution
$\mathbf{R}$	a rotation matrix

1.8 Outline of the Thesis

This thesis is organized as follows. Up to now we have seen some incentives and motivational aspects to use and develop calibration-free TDoA technology. Also a short overview of relevant publications in the field was given, and some applications of hyperbolic localization were suggested.

In the now following Chapter 2 we give a formal introduction to the calibration-free TDoA problem. This includes the discussion of the hyperbolic properties and analysis of calibration-free TDoA from the aspect of uniqueness of solutions. Furthermore, we discuss topological and environmental considerations with respect to TDoA. Afterwards, we propose four fundamental approaches to calibration-free TDoA that have been developed during this thesis, in the domains of far-field approximation, iterative optimization, recursive search space exploration, and probabilistic localization.

We begin with the contributions to the far-field approximation problem, consisting of two statistical estimators of receiver distances, and the Ellipsoid TDoA method that uses ellipsoid characteristics of time differences to find the edges in a triangle, respectively in a tetrahedron, of receivers. Also, we discuss the coordinates-from-distances problem, which follows from far-field approximation.

In Chapter 4 we propose local optimization as a versatile approach to TDoA. First, we give a survey of the gradient descent method and the Gauss-Newton algorithm, for which we have derived the first order derivative for hyperbolic TDoA. Then, we propose the Iterative Cone Alignment method, an optimization approach based on relaxation of spring constraints in a physical mass-spring simulation. In extensive numerical evaluations we discuss the issue of incomplete convergence of local search algorithms, resulting in local minima, where we show that Cone Alignment is superior to gradient descent and Levenberg-Marquardt, finding the correct solution in more than 99 % of random scenarios.

We discuss the problem of convergence in Chapter 5 and present a complete branch-and-bound algorithm that uses hyperbolic constraints to pose an upper bound to the TDoA error function. In a second step, the positions of receivers are approximated by recursive refinement of the search space. We show that the algorithm converges towards the true positions of receivers if the measurement error is limited by ϵ.

In Chapter 6 we turn towards tracking of a continuously moving signal beacon. We present an approach to probabilistic state estimation in a particle filter to create a self-calibrating TDoA algorithm that ensures reliable initialization and robust tracking of the beacon.

Our experiments in the real world are presented in Chapter 7, where we evaluate the algorithmic foundation of calibration-free TDoA in indoor and outdoor settings. We introduce the software platform, capable to run on laptops, the Apple iPhone and on embedded computers, providing a framework for TDoA localization. We demonstrate the feasibility of this framework and of the algorithms to localize a network of computers by environmental clap sounds. Then, we propose our calibration-free TDoA

ultrasound tracking system, where a model train and a RC model car is localized up to the precision of 5 cm. Furthermore, we demonstrate the high robustness of the probabilistic algorithm to measurement outliers. In Chapter 8 we present two advanced algorithms for TDoA calculation by cross correlation, were we show promising results for tracking a human speaker and data transmission by sound.

Then, we introduce the industrial project eCULTS, which is an offspring of the work in this thesis. We summarize in the Conclusions chapter, discussing the range of proposed approaches to the calibration-free TDoA problem and giving an outlook towards future research.

2 Gaining Insight into TDoA

In the previous chapter we have seen an informal introduction to the problem of time differences of arrival and a discussion of imaginable applications. We will now establish the formal definition of the problem and discuss the characteristics of the underlying equation system.

The notation introduced in the following section is maintained throughout this document. However, depending on the application scenario of the algorithm, specific assumptions on positions of signals or receivers are appended, which are described in the respective chapters. We will now start with the basic TDoA setting.

2.1 The Calibration-Free TDoA Problem

Consider a network of n receivers at unknown positions $\mathbf{m}_i$ $(1 \leq i \leq n)$ in p-dimensional Euclidean space $\mathbb{R}^p$, where $p \in \{2, 3\}$. We assume that the clocks of the receivers are synchronized. Now m signals are created at unknown positions $\mathbf{s}_j \in \mathbb{R}^p$ $(1 \leq j \leq m)$ at unknown time points t_j. The signal wavefront propagates spherically from the signal origins $\mathbf{s}_j$ with constant signal velocity c. The signals arrive at the receivers at time points T_{ij}, which can be measured. We assume that the signals are discrete and that they can be distinguished by their time points. Furthermore, we assume that echoes from surrounding walls and from obstacles can be identified and eliminated, such that the receivers obtain the signal in direct line of sight.

Now the problem is to calculate the positions of the receivers $\mathbf{m}_i$, the positions of the signal origins $\mathbf{s}_j$, and the times t_j when the signals were created – only from the times T_{ij} when the signals arrived. The mathematical constraints between the receivers and signals are described by the signal propagation equation

$$\tfrac{1}{c}\|\mathbf{m}_i - \mathbf{s}_j\| \;=\; T_{ij} - t_j \;, \tag{2.1}$$

where only the arrival times T_{ij} and the velocity c are known. All signal times t_j, receiver positions $\mathbf{m}_i$, and signal origins $\mathbf{s}_j$ are unknown in our setting. $\|\cdot\|$ denotes the vector norm in Euclidean space.

An equation system is formed by the equations of n receivers and m signals. Depending on these numbers the equation system may be under-determined, uniquely determined, or over-determined, except for degrees of freedom for translation, rotation, and mirror symmetry, as we will discuss in the next section.

For the cases of at least three receivers in the plane, and four receivers in three-dimensional space, and under the assumption that the signals originate from a distance,

the problem can be solved in closed form. Also, for a minimum number of eight receivers in the plane, respectively ten receivers in 3D space, the equation system can be solved directly [42]. To solve the equation system in general, we have to square the equations. When we distribute the equations we get squared and mixed terms of degree four. According to [4, 42, 51] it does not seem likely that algebraic solutions to the problem in general can be found.

2.2 Hyperbolic Localization

Subtraction of two distance constraint equations of the form (2.1) results in a hyperbolic distance difference equation. For equations $i, k \leq n$ we obtain $n-1$ hyperbolic equations for every signal j:

$$\begin{aligned} \tfrac{1}{c}\|\mathbf{m}_i - \mathbf{s}_j\| - \tfrac{1}{c}\|\mathbf{m}_k - \mathbf{s}_j\| &= (T_{ij} - t_j) - (T_{kj} - t_j) \\ &= T_{ij} - T_{kj} \\ &= \tau_{ikj} \ . \end{aligned} \tag{2.2}$$

Without loss of generality we choose $k = 1$ and $2 \leq i \leq n$, and denote the delay τ_{ij} between the i-th and the first receiver as the *time difference of arrival* (TDoA).

In case the receivers are not synchronized by a common clock or by technical means, the reception time of receiver i is shifted by a constant offset δ_i, which is added to the reception time, so the unsynchronized time of arrival is $\hat{T}_{ij} = t_j + \frac{1}{c}\|\mathbf{m}_i - \mathbf{s}_j\| + \delta_i$ compared to the reference time of receiver 1. The time difference of arrival is therefore $\tau_{ij} = \hat{T}_{ij} - T_{1j} - \delta_i$. For the synchronized case that we consider, unless stated otherwise, the synchronization offset can be omitted.

If signals occur periodically at a known interval a_j then time differences of subsequent signals, which are emitted at times t_{j_1}, t_{j_2} and arrive at a receiver at times T_{ij_1}, T_{ij_2}, form hyperbolic equations:

$$\begin{aligned} \tfrac{1}{c}\|\mathbf{m}_i - \mathbf{s}_{j_1}\| - \tfrac{1}{c}\|\mathbf{m}_i - \mathbf{s}_{j_2}\| &= (T_{ij_1} - t_{j_1}) - (T_{ij_2} - t_{j_2}) \\ &= (T_{ij_1} - T_{ij_2}) - (j_1 - j_2)\, a_j \ . \end{aligned} \tag{2.3}$$

As different receivers do not appear in the same equation, the need for synchronization of receivers is eliminated by this form. Knowing the interval is close to the ToA problem, as only one send time offset between the sender and the receivers needs to be estimated to determine the absolute send times, which yields absolute distances to the sender. Using given transmission intervals is a discrete form of the *frequency difference of arrival* approach (FDoA), which is also known as the *Doppler shift*.

The hyperbolic localization problem also applies to the inverse setting, where synchronized senders emit signals at known times, and a receiver calculates the time differences of arrival, and the receiver is not synchronized with the senders, nor with other receivers. This scenario is well-known from the operation principle of GPS, where

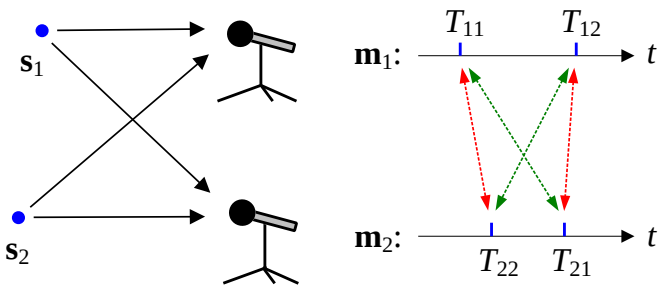


Figure 2.1: Association ambiguity of two simultaneous signals $\mathbf{s}_j$ arriving at two receivers $\mathbf{m}_i$ at times T_{ij}. The closest timestamps in the synchronized time lines suggest the wrong association (red), though the crossed-over association (green) is the correct one.

satellites are synchronized by their high precision clocks, whereas the receivers do not have a precise clock in most cases. They measure pseudoranges to the satellites, which leads to a hyperbolic equation system [6].

As can be seen from the hyperbolic Equation (2.2), the send time t_j is eliminated and the number of equations is reduced by one. This fact is used in optimization, simplifying the equation system and reducing the number of false solutions. On the other hand, the first receiver, as a reference, appears $n-1$ times instead of only one time which induces a specific impact of measurement errors of this receiver. This may create an adverse effect on some algorithms that estimate the uncertainty of a receiver. Even if alternating references are used, or if the complete graph of $n(n-1)$ hyperbolic equations is considered, which increases computational cost, the case of missing signals is difficult to handle in hyperbolic equations. We use hyperbolic equations in the gradient-based optimization approaches described in Chapter 4 and in the recursive search in Chapter 5. We refrain from using hyperbolic equations in the Cone Alignment algorithm and in the probabilistic approach in Chapter 6, and estimate the send times instead.

2.3 Combinatorial Problem of Timestamp Association

In practical application, the relation of a signal obtained by a receiver to signals obtained by other receivers is inherently not known. This relation of received signals to a "signal event" of an emitter, however, are fundamental to TDoA localization. The lack of knowledge of the correct association results in a combinatorial problem, see the example in Fig. 2.1.

Consider m signals which are emitted by various senders, and which are received partly, or all of them, by n receivers. The $m_i \leq m$ signals that are obtained by every receiver i are a subset of the emitted signals, as some signals may be lost due to range limits, or environmental disturbances, or by the detection algorithm that misses signals or detects signals multiple times.

Definition 1 *The association problem of TDoA is the combinatorial problem of connecting the sets of timestamps $I = (I_1, \ldots, I_{m_1})$ and $J = (J_1, \ldots, J_{m_2})$ of two receivers, where $m_1, m_2 \leq m$, by injective mappings $\pi : I \to J$ and $\rho : J \to I$.*

The injective mappings π, ρ are not surjective, as not necessarily all timestamps of the target set are covered due to signal loss. For the number of associations we state the following lemma.

Lemma 1 *In the general TDoA association problem under assumption of missing timestamps, therefore the association is not surjective, exist*

$$\sum_{k=0}^{\min(m_1,m_2)} \binom{m_1}{k} \binom{m_2}{k} k!$$

different configurations associating two sets of timestamps of two receivers.

Proof: Without loss of generality we choose two receivers that obtain the sets of timestamps $I = (I_1, \ldots, I_{m_1})$ and $J = (J_1, \ldots, J_{m_2})$, where $m_1, m_2 \leq m$. Due to the injectivity the inverse exists:

$$I_i = \rho(\pi(I_i)) \quad \text{and} \quad J_l = \pi(\rho(J_l)) \,, \tag{2.4}$$

where $1 \leq i \leq m_1$ and $1 \leq l \leq m_2$. If a timestamp is not associated, assume that the mappings π or ρ are not defined for this timestamp. The combinatorial problem is now to enumerate all mappings between the sets I and J that satisfy the injectivity condition (2.4). We begin with all mappings where $k \leq \min(m_1, m_2)$ timestamps are associated. Choose k timestamps from set I and k timestamps from J. There exist $\binom{m_1}{k}\binom{m_2}{k}$ ways to do this. These selections of k timestamps from I and J can be connected in $k!$ permutations. At last, sum up for all possible k, which yields the proposed sum in Lemma 1. □

Another representation of the previously demonstrated enumeration is a binary matrix of size $(m_1 \times m_2)$, where an association is marked by "1", and "0" otherwise. In every row and in every column can be at most one association mark, due to injectivity, so there can be a maximum of $k \leq \min(m_1, m_2)$ marks, as in the set representation above. In fact, one may not just create as many associations as possible between I and J. One needs to decide instead if two timestamps even belong together.

In a special case, we assume that all receivers receive all m signals, so the mapping is bijective with $k = m$. Then $m!$ ways exist to associate two receivers. This may be extended to n receivers, resulting in $(m!)^{n-1}$ associations. This is a problem of exponential complexity, which grows large even for small numbers of signals and receivers, see Table 2.1 for a numerical example.

Lemma 2 *For a TDoA algorithm, assuming that the problem is uniquely determined, the bijective association problem can be solved in polynomial time complexity by testing constant subsets of receivers and signals.*

	receivers					
signals	1	2	3	4	5	6
1	1	1	1	1	1	1
2	1	2	4	8	16	32
3	1	6	36	216	1296	7776
4	1	24	576	13824	331776	7962624
5	1	120	14400	1728000	$2.07 \cdot 10^8$	$2.49 \cdot 10^{10}$
6	1	720	518400	$3.73 \cdot 10^8$	$2.69 \cdot 10^{11}$	$1.93 \cdot 10^{14}$

Table 2.1: The number of possibilities to associate m signals between n receivers is $(m!)^{n-1}$. Even with only three receivers a test of all possible associations is time consuming and hardly feasible.

Proof: For the test of a constant subset of c_m signals we have $c_m!$ possible ways of association. From the total set of m signals we can choose $\binom{m}{c_m}$ different subsets. These tests are repeated for each of $c_n + 1$ receivers, a constant subset of n receivers, which raises a power of c_n. We obtain

$$\begin{aligned}\left(\binom{m}{c_m} c_m!\right)^{c_n} &= \left(\frac{m!}{c_m!(m-c_m)!}\, c_m!\right)^{c_n} \\ &= \left(m \cdot (m-1) \cdot \ldots \cdot (m-c_m+1)\right)^{c_n} \\ &< \left(m^{c_m}\right)^{c_n} = m^{c_m c_n}\end{aligned}$$

associations to test, which is a polynomial search space with the constant exponent $c_m c_n$. If a TDoA algorithm yields a plausible solution with an association subset, then the association is possibly the correct one. □

As discussed in the next section, the required constants c_m and c_n are rather large, so the feasibility of an algorithm with such a runtime is questionable. However, there is hope that a smaller exponent can be found, for instance by testing incremental sets of signals.

In practice, a windowing technique may be used, where one assumes that a received signal can not belong to another received signal, if the time difference is larger than the distance equivalent of the receivers. Therefore, if $|T_i - T_k| > \frac{1}{c}\|\mathbf{m}_i - \mathbf{m}_k\|$ no association of timestamps T_i and T_k is possible. As the receiver range is usually not known in calibration-free TDoA, a generous constant is chosen as a threshold. In any case, the number of timestamps to compare is reduced to a constant by windowing, if a finite number of timestamps is produced per time unit. For our practical experiments we choose a signal rate so that the interval between two successive signals t_j and t_{j+1} is larger than the longest distance equivalent the signal has to travel, making the signals unique to distinguish.

signal sources	receivers 1	2	3	4	5	6	7	8	9	10	11	12
1	1	2	3	4	5	6	7	8	9	10	11	12
2	3	3	3	3	3	3	3	3	3	3	3	3
3	5	4	3	2	1	0	-1	-2	-3	-4	-5	-6
4	7	5	3	1	-1	-3	-5	-7	-9	-11	-13	-15
5	9	6	3	0	-3	-6	-9	-12	-15	-18	-21	-24
6	11	7	3	-1	-5	-9	-13	-17	-21	-25	-29	-33
7	13	8	3	-2	-7	-12	-17	-22	-27	-32	-37	-42
8	15	9	3	-3	-9	-15	-21	-27	-33	-39	-45	-51
9	17	10	3	-4	-11	-18	-25	-32	-39	-46	-53	-60
10	19	11	3	-5	-13	-21	-29	-37	-45	-53	-61	-69
11	21	12	3	-6	-15	-24	-33	-42	-51	-60	-69	-78
12	23	13	3	-7	-17	-27	-37	-47	-57	-67	-77	-87

Table 2.2: Degrees of freedom for the *two*-dimensional case. Non-positive values indicate potentially solvable problem instances. [54]

An interesting concept is to provide additional information with the timestamp. By using pattern recognition, a similarity index of two received signals may be calculated. Considering the similarities to other received signals and the temporal difference, one may calculate a probability that two signals belong together. This results in a problem of likelihood maximization, depending on the association. In experiments we saw, however, that discrete acoustically similar signals, like a series of clapping, are hard to distinguish by standard signal processing techniques, such as cross correlation of signals [52]. By adding specific markers to synthetic signals one may overcome this problem, which is described in Chapter 8. If such additional information is given, iterative techniques such as clustering or the Iterative Closest Point algorithm [53] may be worth an attempt, however they need to be extended to consider the permuted order of timestamps and to return a set of possible solutions, instead of only one solution after reaching the abort condition.

2.4 The Minimum Cases

We discuss the degrees of freedom and the theoretical bounds on how many receivers n and signal sources m are necessary to find a unique solution. The minimal solutions have also been appealed to in [4] and [5].

We start the discussion for the two-dimensional case. Since the locations of all receivers and signals are unknown we face $2n + 2m$ variables. Furthermore, we do not know when a signal has been created which adds m variables. Since we have no anchor points the number of variables reduces by two variables for translation, for example setting one node as origin, and one variable for rotation, therefore setting another node to the x-axis.

signal sources	receivers 1	2	3	4	5	6	7	8	9	10	11	12
1	0	2	4	6	8	10	12	14	16	18	20	22
2	3	4	5	6	7	8	9	10	11	12	13	14
3	6	6	6	6	6	6	6	6	6	6	6	6
4	9	8	7	6	5	4	3	2	1	0	-1	-2
5	12	10	8	6	4	2	0	-2	-4	-6	-8	-10
6	15	12	9	6	3	0	-3	-6	-9	-12	-15	-18
7	18	14	10	6	2	-2	-6	-10	-14	-18	-22	-26
8	21	16	11	6	1	-4	-9	-14	-19	-24	-29	-34
9	24	18	12	6	0	-6	-12	-18	-24	-30	-36	-42
10	27	20	13	6	-1	-8	-15	-22	-29	-36	-43	-50
11	30	22	14	6	-2	-10	-18	-26	-34	-42	-50	-58
12	33	24	15	6	-3	-12	-21	-30	-39	-48	-57	-66

Table 2.3: Degrees of freedom for the *three*-dimensional case. Non-positive values indicate potentially solvable problem instances. [54]

We assume that all m signals are received at all n receivers which results in the following equation for the degrees of freedom $\mathcal{G}_2$ presented by the problem size:

$$\mathcal{G}_2(n, m) \;=\; 2n + 3m - nm - 3 \tag{2.5}$$

If $\mathcal{G}_2(n, m) > 0$ then there is no unique solution for the problem, i.e. it is under-determined. There is a chance of a unique solution if $\mathcal{G}_2(n, m)$ equals zero. For negative values the problem is over-determined, which might allow the compensation of inaccuracies. See Table 2.2 for the two-dimensional case.

For the three-dimensional case the number of location variables is increased by $n+m$. Here, three variables can be set to a constant for the symmetry induced by translation and three variables for the rotation symmetry which leads to the following degrees of freedom, see Table 2.3:

$$\mathcal{G}_3(n, m) \;=\; 3n + 4m - nm - 6 \tag{2.6}$$

Note that point and mirror symmetry is not covered by this discussion. If we assume that there is abundant supply of ambient signals we can summarize that at least four receivers might allow the solution in the two-dimensional case when at least five signals are available. For the three-dimensional case of the problem five receivers for at least nine signals might be sufficient. However, in our simulations we saw that ambiguities remain which cannot be explained by symmetries, see Fig. 2.2 for an example. Stewénius [4] found a maximum of 344 solutions to the problem of four receivers and five signals in the plane. In fact, six signal sources seem to be the minimum case for the problem.

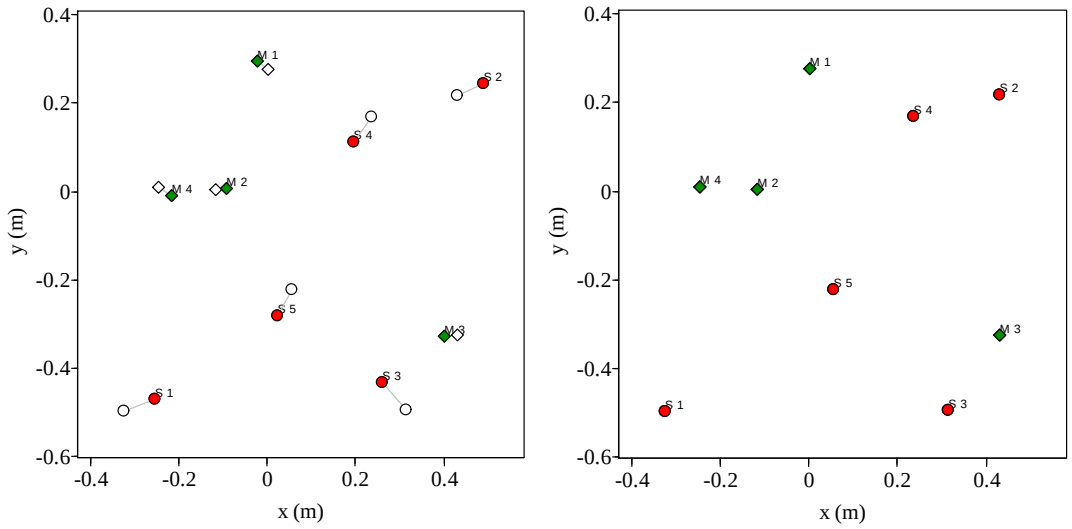


Figure 2.2: Two discrete solutions for the case of four receivers (green) and five signals (red). The true positions are marked in white.

2.5 Estimation of the Convex Hull

One can use the arriving signals for topological considerations of the receiver positions. Given n synchronized receivers in space $\mathbb{R}^p$, $p \in \{2, 3\}$, as defined in Section 2.1, a subset of the convex hull of the receivers can be determined by the timestamp of every signal detected last. Depending on the availability of signals it may also be possible to detect the complete convex hull.

For definition of the convex hull we use the Hesse normal form which introduces a distance measure of a point to a line, respectively a plane, based on the normal vector of the line.

Lemma 3 *For points* $\mathbf{p} \in Z$ *and a receiver* $\mathbf{m} \in Z$ *that is element of the convex hull of* Z*, holds*

$$\exists \mathbf{n} \, \forall \mathbf{p} : \; (\mathbf{m} - \mathbf{p})^T \, \mathbf{n} \geq 0 \, , \qquad \mathbf{n} \in \mathbb{R}^p \, . \tag{2.7}$$

In this definition of Eq. (2.7) is $\mathbf{n}$ the normal vector of a border line through $\mathbf{m}$ if $p = 2$, where all points $\mathbf{p}$ are on the same side of the line. If $p = 3$ then $\mathbf{m}$ and $\mathbf{n}$ define a border plane.

Proof: For a proof of Lemma 3 consider the lines through $\mathbf{m}$ and the two neighbor points on the convex hull, $\mathbf{m}'$ and $\mathbf{m}'' \in Z$. If $\mathbf{m}$ is part of the convex hull, the *internal* angle $\angle_{\mathbf{m}'\mathbf{m}\mathbf{m}''}$ is $\alpha_{\mathbf{m}} \leq \pi$ by definition of convexity. Then all other points $\mathbf{p}$ reside inside the area bounded by the convex hull, therefore in the sector enclosed by $\alpha_{\mathbf{m}}$. For the *external* angle holds $\beta_{\mathbf{m}} = \pi - \alpha_{\mathbf{m}} \geq 0$. Therefore, in the sector of $\beta_{\mathbf{m}}$ exists an array of lines with normal vector $\hat{\mathbf{n}}$, where all points $\mathbf{p}$ are on the same side, which is equivalent to satisfaction of Eq. (2.7). □

For synchronized receivers holds that a signal originating from $\mathbf{s}$ arrives last at a receiver i, iff $\mathbf{m}_i$ is further from $\mathbf{s}$ than all other $\mathbf{m}_k$, therefore

$$\forall k: \; T_i \geq T_k \;\Leftrightarrow\; \|\mathbf{m}_i - \mathbf{s}\| \geq \|\mathbf{m}_k - \mathbf{s}\| \qquad (1 \leq k \leq n) \;, \tag{2.8}$$

which follows from the propagation of sound, Eq. (2.1). Using Lemma 3, we derive the following theorem.

Theorem 1 *A receiver i among synchronized receivers k is element of the convex hull, if a signal exists that arrives last at i such that for all k holds $T_i \geq T_k$.*

Proof: We choose the normal vector $\hat{\mathbf{n}}_u = \mathbf{m}_i - \mathbf{s}$, where we know from the definition of the Euclidean distance that

$$\|\mathbf{m}_i - \mathbf{s}\|^2 = (\mathbf{m}_i - \mathbf{s})^T(\mathbf{m}_i - \mathbf{s}) = (\mathbf{m}_i - \mathbf{s})^T\hat{\mathbf{n}}_u \;. \tag{2.9}$$

Normalizing $\hat{\mathbf{n}}_u$ yields the normalized normal vector $\hat{\mathbf{n}} = \hat{\mathbf{n}}_u/\|\hat{\mathbf{n}}_u\|$. Therefore if $\exists i\, \forall k$: $T_i \geq T_k$ then according to Eqns. (2.8) and (2.9) holds

$$(\mathbf{m}_i - \mathbf{s})^T\hat{\mathbf{n}} \;=\; \|\mathbf{m}_i - \mathbf{s}\| \;\geq\; \|\mathbf{m}_k - \mathbf{s}\| \;\geq\; (\mathbf{m}_k - \mathbf{s})^T\hat{\mathbf{n}} \;. \tag{2.10}$$

We see that the distance between $\mathbf{s}$ and $\mathbf{m}_k$ along $\hat{\mathbf{n}}$ is smaller than the distance between $\mathbf{s}$ and $\mathbf{m}_i$. Eq. (2.10) can be transformed to

$$\begin{aligned}
& (\mathbf{m}_i - \mathbf{s})^T\hat{\mathbf{n}} \;\geq\; (\mathbf{m}_k - \mathbf{s})^T\hat{\mathbf{n}} \\
\Leftrightarrow\quad & (\mathbf{m}_i - \mathbf{s})^T\hat{\mathbf{n}} \;-\; (\mathbf{m}_k - \mathbf{s})^T\hat{\mathbf{n}} \;\geq 0 \\
\Leftrightarrow\quad & \Big((\mathbf{m}_i - \mathbf{s})^T - (\mathbf{m}_k - \mathbf{s})^T\Big)\,\hat{\mathbf{n}} \;\geq 0 \\
\Leftrightarrow\quad & (\mathbf{m}_i - \mathbf{m}_k)^T\hat{\mathbf{n}} \;\geq 0 \;.
\end{aligned}$$

According to Lemma 3 are all receivers $\mathbf{m}_k$ on the same side of a border line defined by $\mathbf{m}_i$ and $\hat{\mathbf{n}}$, therefore is $\mathbf{m}_i$ element of the convex hull of a point set of all other receivers $\mathbf{m}_k$. □

If only few signals are available then we probably cannot detect all elements of the convex hull, as the inverse direction does not hold – not for all receivers that are part of the convex hull does one of m signals necessarily arrive last. See Fig. 2.3, where a subset of the convex hull is detected by signals in the vicinity of the receivers.

However, if signals arrive from all directions from a large distance then the distance between the signals $\mathbf{s}$ and all receivers $\mathbf{m}_k$ is reduced to the distance along a direction vector. This is called the *far-field* case, which is topic of the next Chapter 3. Then, the choice of $\mathbf{n}$ is not limited, therefore one can find appropriate normal vectors and determine all elements of the convex hull.

Once the receivers are known after execution of some TDoA algorithm, the elements of the convex hull of n points can be calculated by the test in Lemma 3, yet at

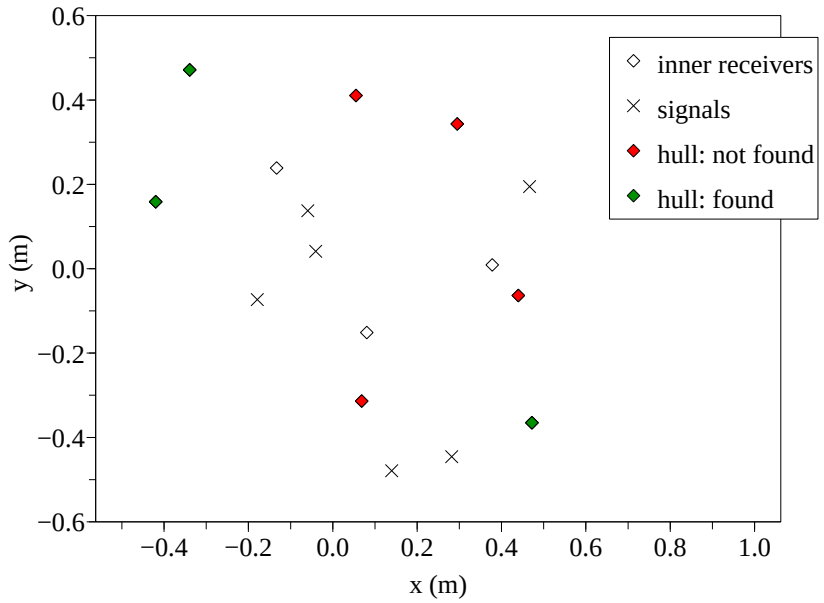


Figure 2.3: Determination of a subset of the convex hull for ten receivers (diamonds). Of seven hull elements could three elements be found (green) by using the last timestamps of six signals.

high costs. For a point, $n-1$ normal vectors need to be tested in the worst case, of which each vector is tested with $n-1$ points, which must be repeated for n points, resulting in a runtime of $\mathcal{O}(n^3)$. There exist faster algorithms to calculate the convex hull of a finite set of points, such as the Graham scan algorithm [55] and the Quickhull algorithm [56], both requiring asymptotic time $\mathcal{O}(n \log(n))$.

2.6 Evaluation of Anchor-Free TDoA

Since there are no anchor points we cannot directly compare the estimated positions of our algorithms to the true positions, also called "ground truth". As no positions are known to the algorithm, the final translation and rotation of the signal source and receiver network are not determined. For an evaluation of the quality of the algorithms we follow the approach of Arun et al. [57] and use singular value decomposition (SVD) to generate a rotation matrix $\mathbf{R}$ aligning the estimated positions to the real-world positions. The alignment algorithm that we use [5] is described in the following.

Let $G = \{\mathbf{g}\}$ and $H = \{\mathbf{h}\}$ be a set of points of equal sizes n in $\mathbb{R}^p$ ($p \in \{2, 3\}$), where G is the ground truth and H is our experimental data. We calculate the arithmetic centers

$$\boldsymbol{\mu}_g = \frac{1}{n}\sum_{i=1}^{n} \mathbf{g}_i \qquad \text{and} \qquad \boldsymbol{\mu}_h = \frac{1}{n}\sum_{i=1}^{n} \mathbf{h}_i \,.$$

The covariance matrix $\mathbf{W}$ is the sum of the dyadic products of G and H:

$$\mathbf{W} = \sum_{i=1}^{n} \Big((\mathbf{g}_i - \boldsymbol{\mu}_g)(\mathbf{h}_i - \boldsymbol{\mu}_h)^T \Big)$$

By subtracting the arithmetic centers we eliminate the translation. The SVD of $\mathbf{W}$ is defined as

$$\mathbf{W} \overset{\text{(SVD)}}{=} \mathbf{UDV}^T$$

where $\mathbf{D}$ is a diagonal matrix of singular values σ_i $(1 \leq i \leq p)$ of $\mathbf{W}$. $\mathbf{U}$ and $\mathbf{V}$ are unitary matrices. $\mathbf{R} = \mathbf{UV}^T$ creates a rotation with a least-squares-optimal mapping of H to G:

$$\hat{H} = \{\hat{\mathbf{h}}\} = \{\mathbf{R}(\mathbf{h} - \boldsymbol{\mu}_h) + \boldsymbol{\mu}_g\} \approx \{\mathbf{g}\} = G$$

The remaining localization error is retrieved by calculating the root mean square (RMS) distance between G and $\hat{H}$.

2.7 The Velocity of Sound

Localization based on the measurement of signal times relies on the signal velocity. In contrast to measurement of the signal amplitude, the velocity of signals is reliable, as it is mostly constant for radio signals and sound.

2.7.1 Sound Velocity in Air

For the velocity of sound in an ideal gas the linear theory [58] yields $c^2 = \frac{\gamma p}{\rho}$, where γ is the adiabatic index, i.e. the ratio of heats, p is the pressure, and ρ is the density. Using the ideal gas laws we replace

$$c = \sqrt{\frac{\gamma p}{\rho}} = \sqrt{\frac{\gamma R T}{M}} = \sqrt{\frac{\gamma k T}{m}} \,,$$

where R is the molar gas constant, T is the absolute temperature, M is the molar mass, k is the Boltzmann constant, and m is the mass of a molecule. Therefore, in an ideal gas, the sound velocity depends only on the temperature, not on pressure.

Depending on the type of gas, the masses m of gas molecules vary, and therefore the sound velocities. In reality there is also a small influence of humidity on the sound velocity, as an effect of mixing of lighter water molecules into air. According to [58], the sound velocity is approximately 0.1% to 0.6% higher in saturated air compared to dry air, depending on the temperature.

For dry air at 20 °C, the sound velocity is approximately $c_{\text{air}} = 343\,\frac{\text{m}}{\text{s}}$. For comparison, in oxygen is the sound velocity $c_{\text{oxygen}} = 318\,\frac{\text{m}}{\text{s}}$, in nitrogen $c_{\text{nitrogen}} = 333\,\frac{\text{m}}{\text{s}}$, in helium $c_{\text{helium}} = 981\,\frac{\text{m}}{\text{s}}$. For air, one can use the temperature-dependent rule of thumb

$$c_{\text{air}} = 331.3\frac{\text{m}}{\text{s}} + 0.6\,\vartheta\,\frac{\text{m}}{\text{s}} \,, \tag{2.11}$$

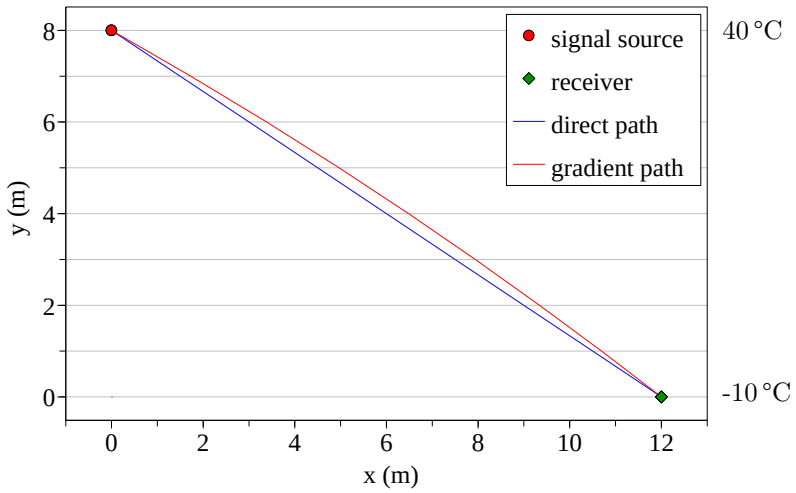


Figure 2.4: In a temperature gradient the sound velocity decreases while propagating from the warm ceiling to the cold floor. The propagation path is no longer a straight line. In the picture the gradient is exaggerated by values of -10 °C to 40 °C.

where ϑ is the temperature in degrees Celsius, ignoring the rather small influence of humidity. As the influence of temperature is small in indoor environments, we use in our experiments the standard velocity $c_{\text{air}} = 343\,\frac{\text{m}}{\text{s}}$.

2.7.2 Sound Propagation in a Temperature Gradient

In an application-specific scenario, a signal emitter **s** is placed at the ceiling of a large hall, sending a signal in a periodic interval to a receiver **m**, which may be a robot driving on the floor. If the air is not perturbed by air conditioning or by open doors or windows, a smooth gradient of heat distribution may be assumed, with the coldest air on the floor and the warmest air just below the ceiling. Therefore, the temperature is assumed to be linear in the interval $[\vartheta_0, \vartheta_y]$, depending on the height $h \in [0, y]$. According to Eq. (2.11) the velocity of sound is linear in the temperature with $c(\vartheta) = 331.3 + 0.6\,\vartheta$.

The signal is obtained by **m** after the propagation time t of any path between **s** and **m**. In uniform air this path equals a straight line, as assumed in Section 2.1. In the following, we analyze the form of the propagation path in a uniform temperature gradient.

Considering the linear temperature gradient, we express the sound velocity as a function of the height, so we obtain $c(h) = (1 - \frac{h}{y})\, c(\vartheta_0) + \frac{h}{y}\, c(\vartheta_y)$, which we abbreviate by

$$c_h = \left(1 - \frac{h}{y}\right) c_0 + \frac{h}{y}\, c_y \;, \tag{2.12}$$

where c_y is the initial sound velocity at the ceiling and c_0 is the final velocity close to the floor.

Denote the angle between the normal vector $(0,1)^T \in \mathbb{R}^2$ of the temperature gradient and a vector of sound wave propagation as α. According to Snell's Law of Refraction the angle of incidence α_1 in a medium with signal velocity c_1 and the angle of the refracted beam α_2 in a medium with signal velocity c_2 behave as

$$\frac{\sin(\alpha_1)}{\sin(\alpha_2)} = \frac{c_1}{c_2}\,, \tag{2.13}$$

which can be reformulated to calculate the incidence, such that

$$\sin(\alpha_h) = \frac{c_h}{c_0}\sin(\alpha_0)$$

is the incidence at height h, if α_0 is the initial incidence at height 0. We obtain the horizontal propagation component x by integration over h. To create the antiderivative we replace and collect the constant terms as follows:

$$\begin{aligned}
x &= \int_{h=0}^{y} \tan(\alpha_h)\,\mathrm{d}h \\
&= \int_{h=0}^{y} \frac{\sin(\alpha_h)}{\sqrt{1-\sin(\alpha_h)^2}}\,\mathrm{d}h \\
&= \int_{h=0}^{y} \frac{\sin(\alpha_0) + \frac{(c_y-c_0)\sin(\alpha_0)}{y\,c_0}h}{\sqrt{1-\left(\sin(\alpha_0) + \frac{(c_y-c_0)\sin(\alpha_0)}{y\,c_0}h\right)^2}}\,\mathrm{d}h \\
&= \left[-\frac{y\,c_0}{(c_y-c_0)\sin(\alpha_0)}\sqrt{1-\left(\sin(\alpha_0) + \frac{(c_y-c_0)\sin(\alpha_0)}{y\,c_0}h\right)^2}\right]_0^y
\end{aligned} \tag{2.14}$$

Now, the initial angle α_0 can be calculated by solving the antiderivative in Eq. (2.14) for

$$\sin(\alpha_0) = \frac{2\,c_0\,xy}{\sqrt{(x^2+y^2)\Big((c_y-c_0)^2\,x^2 + (c_y+c_0)^2\,y^2\Big)}}\,,$$

which we use to calculate the travel time t_{grad} and the trajectory length d_{grad} from the integrals

$$t_{\text{grad}} = \int_{h=0}^{y} \frac{\mathrm{d}h}{c_h\cos(\alpha_h)} = \int_{h=0}^{y} \frac{1}{c_h\sqrt{1-\sin(\alpha_h)^2}}\,\mathrm{d}h \tag{2.15}$$

$$d_{\text{grad}} = \int_{h=0}^{y} \frac{\mathrm{d}h}{\cos(\alpha_h)} = \int_{h=0}^{y} \frac{1}{\sqrt{1-\sin(\alpha_h)^2}}\,\mathrm{d}h\,. \tag{2.16}$$

reference velocity	travel time	error
t_{grad}	0.042395 s	
t_{lin} with c_0	0.044325 s	4.552 %
t_{lin} with c_y	0.040632 s	−4.158 %
t_{lin} with c_{avg}	0.042398 s	0.008 %

Table 2.4: Sound travel times under assumption of different velocities and of the linear propagation model Eq. (2.1). The average velocity $c_{\text{avg}} = \frac{1}{2}(c_0 + c_y)$ is a viable assumption even for a temperature gradient.

In an exaggerated numerical example we use values of -10 °C close to the floor to 40 °C below the ceiling. The respective sound velocities range from 325 $\frac{\text{m}}{\text{s}}$ to 355 $\frac{\text{m}}{\text{s}}$. According to Eq. (2.15) we calculate the travel time t_{grad} to traverse in the example from $\mathbf{s} = (0, 8)^T$ to $\mathbf{m} = (12, 0)^T$. The propagation path that yields the duration t_{grad} is a curve, as illustrated in Fig. 2.4.

Using only one of the initial or the final temperature and the linear propagation assumption in Eq. (2.1) to calculate the expected travel time $t_{\text{lin}} = \frac{1}{c}\|\mathbf{s} - \mathbf{m}\|$ induces an error of almost 5 % for the chosen gradient and positions. However, the average of both velocities $c_{\text{avg}} = \frac{1}{2}(c_0 + c_y)$ yields an expected travel time where the deviation from the real travel time t_{grad} is below 0.01 %. See Table 2.4 for the expected runtimes. Therefore, the linear propagation model is a trustable approximation for such experiment settings, even in a temperature gradient, if both the initial and the final sound velocities are considered.

3 Far-Field Approximation

The direct solution to the calibration-free TDoA problem is a difficult road due to the non-linearity and large problem complexity. To date, approaches are computationally intensive, or do not consider the minimum case, or are numerically fragile. Furthermore, if a specific minimum number of receivers and signals is not provided, depending on the algorithm and the considerations in Section 2.4, typical approaches, such as optimization, are helpless, as the problem is under-determined.

However, when receivers or signals are not arbitrarily distributed but are subject to assumptions, statistical and approximative algorithms may be applied. One of these constraints is the *far-field assumption*, where a remote origin of signals is assumed. Then, the direction vector of a signal is identical for all receiving units, which simplifies the problem immensely. In many cases the computational complexity of approximative methods is small, compared to approaches to the general problem, so they are capable to support more elaborate techniques without notable computational effort.

The original approaches of Thrun et al. [44] consider the term *far-field* with respect to TDoA as signals of which the distance is infinite. We follow this definition, but we include also algorithms into this chapter for which their model assumes a distribution of signals closer than far-field. These still work when the signals are directly beneath receivers, if at least some of them are approximately aligned with the receivers, for example the Min/max method, which is described in Section 3.1.

In this chapter we consider localization of unknown receivers by the time differences of signals from unknown senders, which are assumed far away. Once the locations of receivers are known, estimating the direction of the signals is straightforward. Let the known distance vector between two receivers be $\mathbf{x}_i = \mathbf{m}_1 - \mathbf{m}_i$, where $2 \leq i \leq n$. According to the cosine law, in space $\mathbb{R}^2$ the time difference $\tau_i = T_1 - T_i$ depends only on the angle γ_i of the signal [37]. In this way, a direction vector $\mathbf{u}$ of the signal is obtained by

$$\cos(\gamma_i) \;=\; \frac{c\tau_i}{\|\mathbf{x}_i\|} \;=\; \frac{\mathbf{x}_i^T\mathbf{u}}{\|\mathbf{x}_i\| \cdot \|\mathbf{u}\|} \quad \Rightarrow \quad \mathbf{x}_i^T\mathbf{u} = c\tau_i \;. \tag{3.1}$$

Here, we assume the direction vector normalized by $\|\mathbf{u}\| = 1$. Calculating vector $\mathbf{u}$ in space $\mathbb{R}^p$ requires at least $p-1$ time differences, due to normalization of $\mathbf{u}$. In a simple linear approach at least p time differences are required to solve the equation system $\mathbf{x}_i^T\mathbf{u} = c\tau_i$, ignoring the non-linear normalization.

Calculating the direction of signals is robust in the far-field case and allows for neat applications such as auto-tracking of cameras, as demonstrated in [37]. In contrast, estimating the range of remote signals is imprecise, due to the missing parallax of

the distance vectors. In the following methods we do not calculate the signals at all, and are satisfied with estimating the receivers. Once their positions are known, the directions of signals may be calculated as described.

In the next sections we discuss two classes of approaches to the far-field approximation problem. First, we present statistical methods that estimate the distance between a pair of receivers. They are based on simple evaluation of the observed time differences, or on statistical analysis of the distribution of timestamps. These methods can be applied in many cases if at least two receivers and some signals are available. The quality of the estimation, however, depends heavily on the number of signals and their distribution.

Second, we present the Ellipsoid TDoA method, which is a more advanced algorithm to apply to three receivers in the plane and to four receivers in three-dimensional space. This one receiver less than the required minimum number for the general problem. As we assume far-field distribution of signals, the equation system is reduced, which compensates for the missing receiver. The Ellipsoid method uses the fact that signals that originate from a distance characterize an ellipsoid between three, respectively four receivers. From the shape of the ellipsoid one can estimate the distances and angles between the receivers that caused such a distribution of time differences.

All of the described approximation approaches result in determination of the distances and angles between receivers, not their Cartesian coordinates. Calculating these is an independent problem widely covered in literature. Due to its significance for us, we give a short survey of two popular algorithms, which are Multidimensional Scaling and iterative construction of distance graphs, followed by a discussion of the bipartite graph problem.

3.1 Min/Max Method

Even if the signals do not actually originate from an infinite distance, an estimation of the receiver configuration is feasible. We start with a simple approach, where we relax the far-field assumption of signals to an assumption that a sufficient number of signals resides in the unit disk with the receivers. Then, we suppose that at least some of them reside on a line with two receivers, and beyond both receivers.

We have presented a theoretical analysis of the feasibility of distance estimation from signals distributed in the unit disk in [59, 60]. We defined the *ϵ-critical area* of two nodes $\mathbf{m}_1$ and $\mathbf{m}_2$ as the set of points $\mathbf{p}$ in the plane where holds

$$\|\mathbf{m}_1 - \mathbf{m}_2\| - (\|\mathbf{p} - \mathbf{m}_1\| - \|\mathbf{p} - \mathbf{m}_2\|) \leq \epsilon \,. \tag{3.2}$$

The convex area that is described by this inequality is bounded by a hyperbola, see Fig. 3.1. If in this critical area signals are produced, then the distance estimation is accurate up to an absolute error of ϵ. We showed that these regions are surprisingly large, which makes distance estimation with signals in the unit disk realistic, given a sufficient number of signals.

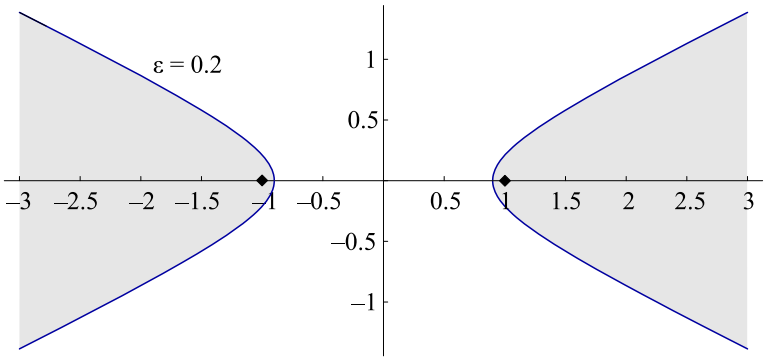


Figure 3.1: The 0.2-critical areas of two nodes at $(-1, 0)$ and $(1, 0)$ are on the left and right side of the hyperbolas. [60]

We derive the estimator for the distance of receivers in the following. It is based on the observation that the maximum time difference between two receivers i and k is bounded by their distance $d_{ik} = \|\mathbf{m}_i - \mathbf{m}_k\|$. Assume a number of m signals originating from the distance in two- or three-dimensional space. The signals $j = 1, \ldots, m$ arrive at the receivers at times T_{ij}, respectively T_{kj}. Every receiver maintains its own clock, so the time difference is shifted by an unknown offset δ_{ik}. Due to uncertainty in the measurement process, the observed reception times contain an error, corrupting the true reception times. We assume that these errors are bounded by $\epsilon_{ij}, \epsilon_{kj} \in [-\epsilon, \epsilon]$. The observed time difference is

$$\tau_{ikj} \;=\; (T_{ij} + \epsilon_{ij}) - (T_{kj} + \epsilon_{kj}) \;. \tag{3.3}$$

The maximum time difference will occur for a signal that is aligned with the axis through $\mathbf{m}_i$ and $\mathbf{m}_k$, and that resides beyond both receivers. The minimum time is analogous.

$$\tau_{\max} \;=\; \frac{d_{ik}}{c} + \delta_{ik} \;, \qquad \tau_{\min} \;=\; -\frac{d_{ik}}{c} + \delta_{ik}$$

Ignoring the fact, that the reception times and the distribution of signals contain errors, we propose a simple estimator for the distance, based on the convex hull of the observed measurements:

$$\tilde{d}_{ik} \;=\; \frac{c}{2}(\hat{\tau}_{\max} - \hat{\tau}_{\min}) \;. \tag{3.4}$$

The offset between the receiver clocks is obtained by

$$\tilde{\delta}_{ik} \;=\; \frac{1}{2}(\hat{\tau}_{\max} + \hat{\tau}_{\min}) \;. \tag{3.5}$$

The estimator will underestimate the true distance d_{ik}, if no measurements are observed from the receiver axis, outside the receivers, and overestimate the distance in case of measurement errors.

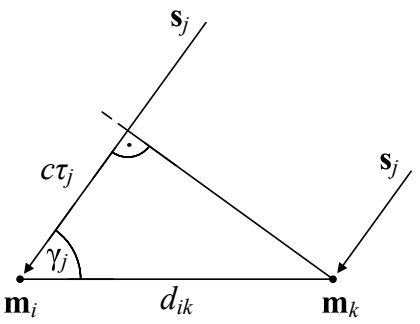


Figure 3.2: For a distant signal j the time difference of arrival τ depends only on the angle of origin γ of the signal according to $\cos(\gamma_j) = \frac{c\tau_j}{d_{ik}}$.

Approaches based on the range of time differences were demonstrated by Biswas and Thrun [47] and by Pertila et al. [38] in practice, where the distances between sensor motes and mobile phones were estimated. In [48] a distance estimator similar to the Min/max-method is presented that explicitly considers measurement errors. In the following we present two distance estimators that are more robust to measurement uncertainty.

3.2 Statistical Approach

In a typical scenario we refer to two receivers surrounded by a large number of signals originating from the horizon [51]. We assume that the angles of origin of the signals are equally distributed, which may be approximately the case for a person standing in the middle of a thunderstorm, or a pedestrian in a forest surrounded by birdsong. Given this equal distribution of signal origins we propose a robust estimator for the distance between a pair of receivers.

Recall that under the assumption of a large distance, the time difference of a signal j arriving at two receivers i and k in the plane depends only on the angle of the signal γ_j with respect to the receivers. Given a signal arriving at the receivers at time $\tau_j = T_{ij} - T_{kj}$ the cosine of the angle is $\cos(\gamma_j) = \frac{c\tau_j}{d_{ik}}$, where d_{ik} is the distance between i and k, see Fig. 3.2.

We consider the circle sector $[-\phi, \phi]$ for arbitrary $\phi \in [0, \pi]$. The probability that the angle γ_j of any signal is inside the sector is

$$p(|\gamma_j| < |\phi|) \;=\; \frac{\phi}{2\pi} - \frac{-\phi}{2\pi} \;=\; \frac{\phi}{\pi}\,. \tag{3.6}$$

Analogously, the probability that the time difference of a signal τ_j is *smaller* than any $\tau \in \left[-\frac{d}{c}, \frac{d}{c}\right]$ is $p(\tau_j < \tau) = 1 - \frac{\phi}{\pi}$. We obtain the cumulative distribution function

$$F(\tau) \;=\; 1 - \frac{\phi}{\pi} \;\overset{(3.1)}{=}\; 1 - \frac{1}{\pi}\arccos\left(\frac{c\tau}{d}\right)\,. \tag{3.7}$$

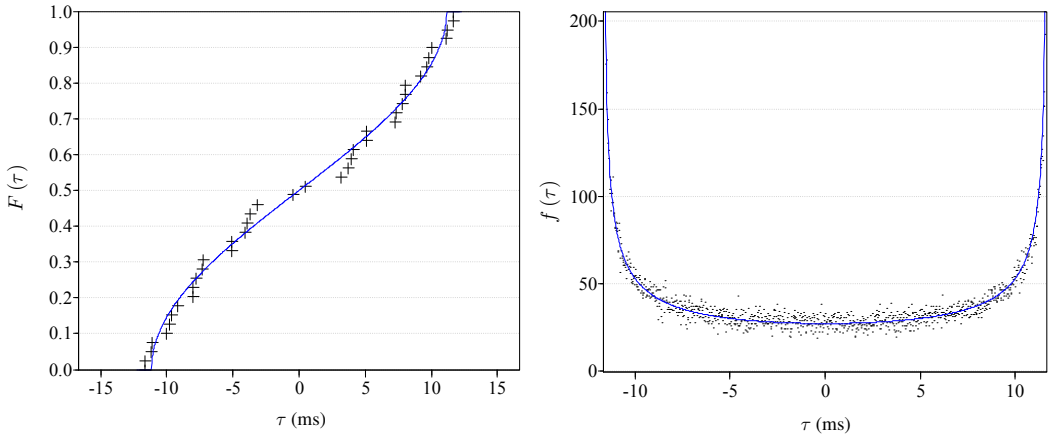


Figure 3.3: *Left*: The cumulative distribution function $F(\tau)$. *Right*: The probability density function $f(\tau)$ reminds of a bathtub. [51]

The derivative of F is the probability density function

$$f(\tau) \;=\; \frac{1}{\pi\sqrt{\frac{d^2}{c^2}-\tau^2}}\;, \tag{3.8}$$

which is defined in the domain $\left(-\frac{d}{c},\frac{d}{c}\right)$.

The limit of the density function $f(\tau)$ is $\lim_{\tau\to\pm\frac{d}{c}} = +\infty$, reminding of the shape of a bathtub, see Fig. 3.3. This also suggests that the proportion of signals with maximum or minimum time difference is comparatively high, which supports the Min/max approach from the previous section.

3.2.1 Variance Estimator

The variance of the signal distribution can be used as an estimator for the distance between the two receivers. For a random variable X with continuous probability function $h(x)$ the variance is defined as

$$\mathrm{Var}(X) = \int (x-\mu)^2\, h(x)\;\mathrm{d}x\;.$$

As we assume that the signals are equally distributed and the receivers are synchronized, i.e. the offset of their clocks is zero, the mean μ is zero. So, the variance of the signals in the domain $\left(-\frac{d}{c},\frac{d}{c}\right)$ is

$$\sigma_\tau^2 \;=\; \int_{-\frac{d}{c}}^{\frac{d}{c}} \tau^2 f(\tau)\;\mathrm{d}\tau \;=\; \frac{d^2}{2c^2}\;, \qquad (c>0,\; d>0)\;, \tag{3.9}$$

from which we derive an estimator for the distance $\tilde{d} = c\sqrt{2\,\sigma_\tau^2}$.

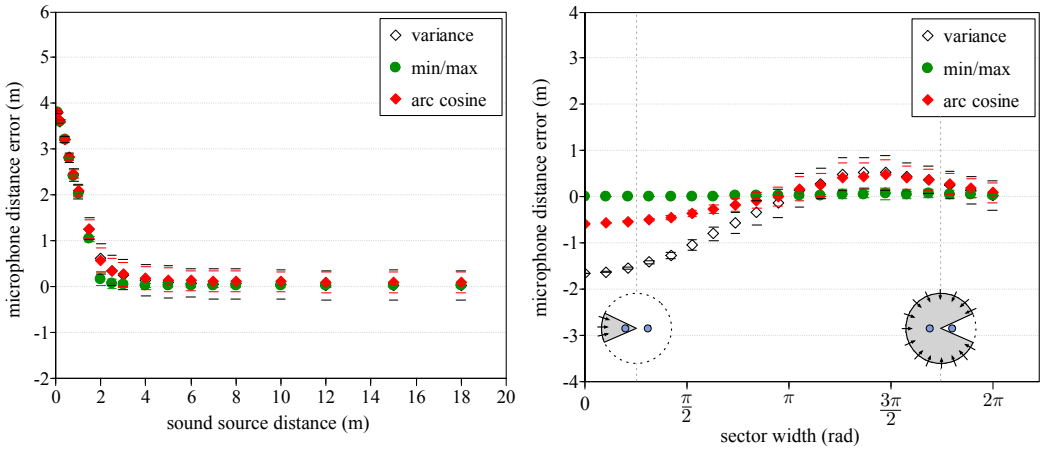


Figure 3.4: Numerical evaluation of the *Min/max*, the *Variance*, and the *Arc Cosine* estimators. The distance between the two receivers is 4 m. *Left*: Violation of the assumption of distant signals. *Right*: Violation of the assumption of equal distribution of the signals. Note that the Min/max estimator is not robust against TDoA errors (not in picture). [51]

The statistical estimator based on the variance of the observed time differences is robust, given a sufficient number of signals. However, when the assumption of equal distribution of the signals at the horizon is violated, the variance method is prone to misestimation. The variance estimator tends to overestimation if the signals are cumulated at the elongation of the axis of both receivers, and to underestimation if they are cumulated perpendicular to the receivers.

3.2.2 Arc Cosine Estimator

We propose an estimator to reconstruct the distribution from the observed time differences, reducing the susceptibility to misestimation. By sorting the time differences τ_j by ascending value, and assigning an index I_j $(1 \leq j \leq m)$ we obtain coordinates $(\tau_j, \frac{1}{m} I_j)$, which correspond to the cumulated distribution $F(x)$. Knowing that $\sin(\arccos(x)) = \sqrt{1-x^2}$, we reformulate $F(x)$ in the following way:

$$I_j = 1 - \frac{1}{\pi} \arccos\left(\frac{c}{d}\tau_j\right) \quad \Rightarrow \quad \underbrace{\left(1 - \sin\left((1-I_j)\pi\right)^2\right)}_{\mathbf{q}} \underbrace{d^2}_{\mathbf{u}} = \underbrace{\tau_j^2 c^2}_{\mathbf{b}}$$

Using linear regression we obtain the best fit of $F(x)$ to the observed signals, with respect to minimizing the residuals, cf. Fig. 3.3. Given $m \geq 1$ signals we obtain a system of equations. Using the least squares method we solve

$$\left(\mathbf{q}^T\mathbf{q}\right)\mathbf{u} = \mathbf{q}^T\mathbf{b} \quad \Rightarrow \quad \mathbf{u} = \left(\mathbf{q}^T\mathbf{q}\right)^{-1}\left(\mathbf{q}^T\mathbf{b}\right) \ ,$$

where the parameter vector $u_1 = \mathbf{u}$ yields the solution $\tilde{d} = \sqrt{u_1}$.

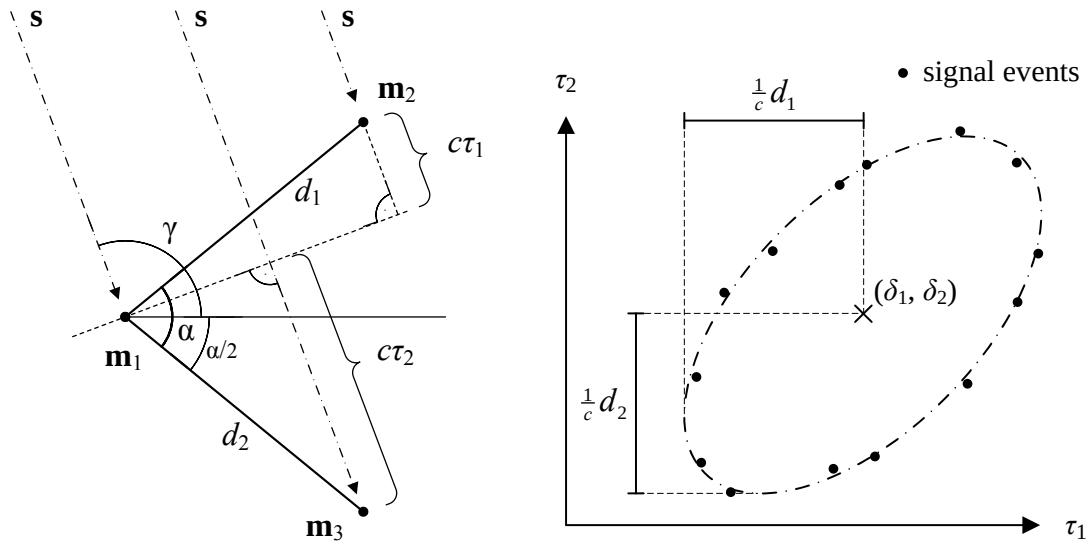


Figure 3.5: Scheme of the Ellipsoid TDoA method. The time differences τ_1 and τ_2 of distant signal events $\mathbf{s}$ for three receivers describe an ellipse in space $\mathbb{R}^2$. From this ellipse the receiver distances d_1 and d_2 and the angle α can be calculated. [61]

Using numeric simulation we have evaluated the precision and robustness of the three estimators, violating the presupposed conditions individually, which are the assumption of distant signals, the number of signals, the precision of the observations, and the assumption of equal distribution of the signals [51].

According to the evaluation, distance approximation of distant signals is computationally fast and comparatively precise, even if the assumptions are only roughly satisfied. The Arc Cosine estimator is superior to the other approaches, as it is robust to timing errors and to adverse distribution of the signals, cf. Fig. 3.4.

3.3 The Ellipsoid TDoA Method

The scenario of receivers surrounded by signals from a very large distance is particular, as a solution to the problem can be calculated in closed form. Some approaches exist in literature where the problem has been considered. Thrun proposed one of the earliest approaches to the problem [44]. In the paper, affine geometry is used to find the distances of n receivers in the plane, however they are restricted to synchronized receivers. Based on the approach of Thrun, an analysis of the problem and its failure modes in three dimensions has been published in a recent paper [45]. In another recent publication the authors calculate the positions of unsynchronized antennas that receive signals from distant transmitters [46]. This algorithm tends towards a complete solution of the far-field approximation problem, but five receivers in three dimensions are required, which is more than the minimum case.

In our approach, the *Ellipsoid TDoA method* [61, 62], we propose a solution to the minimum case of three synchronized receivers and three signals in two dimensions, and to three dimensions, where we require the minimum numbers of four receivers and six signals. In addition, our approach is feasible for unsynchronized receivers, which is straightforward by regression of a general ellipsoid equation. Then, the required number of signals increases to five in the plane, and nine in three dimensions, which is the theoretical minimum case.

In the following, we present a geometrical description of the TDoA distribution of distant signals in the plane to calculate the distances between three receivers.

3.3.1 The Receiver Triangle

Consider three receivers in a triangle $\triangle \mathbf{m}_1\mathbf{m}_2\mathbf{m}_3$ in $\mathbb{R}^2$, where $d_1 = \|\mathbf{m}_1 - \mathbf{m}_2\|$ and $d_2 = \|\mathbf{m}_1 - \mathbf{m}_3\|$ denote the distances, and α denotes the angle $\angle_{\mathbf{m}_1\mathbf{m}_2\mathbf{m}_3}$ between the line segments $\mathbf{m}_1\mathbf{m}_2$ and $\mathbf{m}_1\mathbf{m}_3$.

A distant signal is obtained by the receivers at absolute times T_1, T_2, and T_3, that yield the time differences $\tau_1 = T_1 - T_2$ and $\tau_2 = T_1 - T_3$. As the signal origins are far away, we describe the signal angle as

$$\cos(\gamma_1) = \frac{c\tau_1}{d_1} \qquad \text{and} \qquad \cos(\gamma_2) = \frac{c\tau_2}{d_2} \,.$$

We define $\gamma = \gamma_1 + \alpha/2 = \gamma_2 - \alpha/2$ as the angle between $\mathbf{s}$ and the bisection of $\mathbf{m}_1\mathbf{m}_2$ and $\mathbf{m}_1\mathbf{m}_3$, cf. Fig. 3.2. We combine tuples of time differences to coordinates (x, y) as follows:

$$\begin{aligned} x &= c\tau_1 = d_1\cos(\gamma_1) = d_1\cos(\gamma - \alpha/2) \\ y &= c\tau_2 = d_2\cos(\gamma_2) = d_2\cos(\gamma + \alpha/2) \end{aligned} \tag{3.10}$$

By combining the equations we obtain an ellipse equation, which we write in the normalized form

$$\underbrace{\frac{1}{d_1^2\sin(\alpha)^2}}_{a} x^2 \;+\; \underbrace{\frac{1}{d_2^2\sin(\alpha)^2}}_{b} y^2 \;+\; \underbrace{\frac{-2\cos(\alpha)}{d_1 d_2\sin(\alpha)^2}}_{c} xy \;=\; 1 \,. \tag{3.11}$$

In order to allow the performed transformations we assume that $\alpha \notin \{0, \pi\}$, i.e. the receivers do not reside on a line.

Equation (3.11) can be reformulated such that, given the three coefficients a, b, c, we can solve for the parameters d_1, d_2, and α, which uniquely determine the triangle of receivers $\triangle \mathbf{m}_1\mathbf{m}_2\mathbf{m}_3$. We obtain the following solutions:

$$d_1 = 2\sqrt{\frac{b}{4ab - c^2}} \,, \quad d_2 = 2\sqrt{\frac{a}{4ab - c^2}} \,, \quad \cos(\alpha) = \frac{-c}{2\sqrt{ab}} \tag{3.12}$$

The next step is to solve the ellipse equation for the unknown coefficients a, b, and c.

3.3.2 Solving the Equation System

To find a unique solution to three unknown values d_1, d_2, and α we require at least three signals from different directions, if the receivers are synchronized, forming tuples (x_j, y_j), where $1 \leq j \leq m$. The measurements form an equation system with $m \geq 3$ equations of the form

$$a\,x_j + b\,y_j^2 + c\,x_j y_j = 1\ . \tag{3.13}$$

We use linear regression to calculate the three parameters. The Eqns. (3.13) can be written in matrix notation as

$$\underbrace{\begin{bmatrix} x_1^2 & y_1^2 & x_1 y_1 \\ \vdots & \vdots & \vdots \\ x_m^2 & y_m^2 & x_m y_m \end{bmatrix}}_{\mathbf{Q}} \underbrace{\begin{pmatrix} a \\ b \\ c \end{pmatrix}}_{\mathbf{u}} \quad = \quad \vec{1}$$

If $m > 3$ we use the least squares method to solve for vector $\mathbf{u}$:

$$\left(\mathbf{Q}^T\mathbf{Q}\right)\mathbf{u} = \mathbf{Q}^T\vec{1} \quad \Rightarrow \quad \mathbf{u} = \left(\mathbf{Q}^T\mathbf{Q}\right)^{-1}\left(\mathbf{Q}^T\vec{1}\right) \tag{3.14}$$

If $m = 3$ we solve directly by $\mathbf{u} = \mathbf{Q}^{-1}\vec{1}$. If $m < 3$ the system is under-determined and cannot be solved by this method. Now, we use the parameters a, b, and c in Eqns. (3.12) to determine the distances in the receiver triangle.

3.3.3 Unsynchronized Receivers

If the receivers have no common time base then the relative times τ_1 and τ_2 are shifted by constant offsets δ_1 and δ_2. This corresponds to the center of the ellipse being shifted by a vector $(\delta_1, \delta_2)^T$. The shifted ellipse is described by a general ellipse equation

$$a\,x_j^2 + b\,y_j^2 + c\,x_j y_j + d\,x_j + e\,y_j = 1 \tag{3.15}$$

We use regression of at least $m \geq 5$ signals to solve for the equation system

$$\underbrace{\begin{bmatrix} x_1^2 & y_1^2 & x_1 y_1 & x_1 & y_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_m^2 & y_m^2 & x_m y_m & x_m & y_m \end{bmatrix}}_{\mathbf{Q}} \underbrace{\begin{pmatrix} a \\ b \\ c \\ d \\ e \end{pmatrix}}_{\mathbf{u}} \quad = \quad \vec{1}$$

We obtain the parameter vector $\mathbf{u}$ using Eq. (3.14) as in the synchronous case. We apply a transformation to the general description of the ellipse into a translation invariant representation, so we can extract coefficients $\hat{a}, \hat{b}, \hat{c}$, corresponding to an ellipse of the form

$$\hat{a}\,(x_j - \delta_1)^2 + \hat{b}\,(y_j - \delta_2)^2 + \hat{c}\,(x_j - \delta_1)(y_j - \delta_2) = 1\ . \tag{3.16}$$

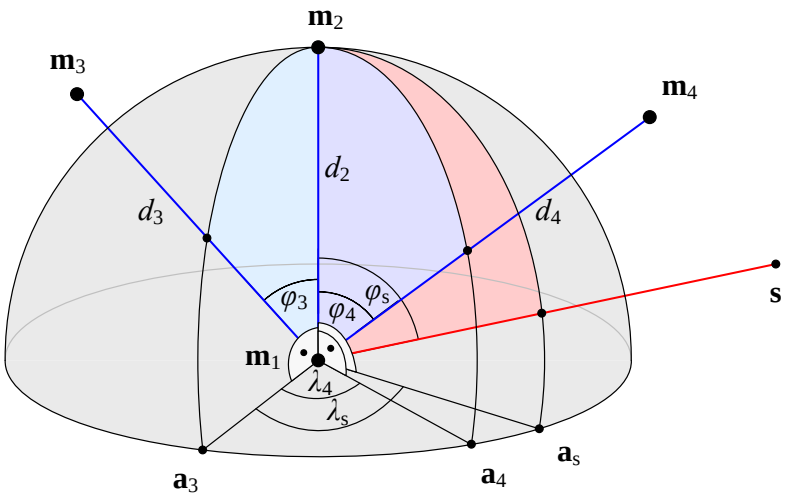


Figure 3.6: Scheme of the Ellipsoid method in three dimensions. Three distances d_2, d_3, d_4, and three angles φ_3, φ_4, and λ_4 define a tetrahedron of four receivers $\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3, \mathbf{m}_4$. A signal arrives from the angles φ_s, λ_s. [63]

This transformation to translation invariant coefficients is calculated by

$$\hat{a} = a\frac{K}{L}\,, \quad \hat{b} = b\frac{K}{L}\,, \quad \hat{c} = c\frac{K}{L}\,, \qquad \text{where}$$

$$K = 4ab - c^2 \quad \text{and} \quad L = ae^2 + bd^2 - c^2 + 4ab - cde\,, \quad \text{if } L \neq 0.$$

As in the synchronous case, we use the Equations (3.12) to compute the receiver triangle by inserting the parameters $\hat{a}, \hat{b}, \hat{c}$. Additionally, we can calculate the synchronization offsets between the receivers $\mathbf{m}_1$ and $\mathbf{m}_2$ by $\delta_1 = -(2bd - ce)\frac{1}{K}$ and between $\mathbf{m}_1$ and $\mathbf{m}_3$ by $\delta_2 = -(2ae - cd)\frac{1}{K}$, if $K \neq 0$.

Once the angle α and the lengths of the adjacent edges are known, we calculate the Cartesian receiver coordinates as described in Section 3.4. If we want, we can calculate the length of the remaining edge by the the cosine rule in the arbitrary triangle:

$$d_3 \;=\; \|\mathbf{m}_1 - \mathbf{m}_3\| = \sqrt{d_1^2 + d_2^2 - 2d_1 d_2 \cos(\alpha)}$$

Now, we can determine the direction of the signals from the triangle by multilateration as described in Eq. (3.1).

3.3.4 A Solution in Three Dimensions

The Ellipsoid TDoA method can be extended to three dimensions [63]. Consider four receivers $\mathbf{m}_1$, $\mathbf{m}_2$, $\mathbf{m}_3$, $\mathbf{m}_4$ that form a tetrahedron and distant emitters that send signals at discrete times. The tetrahedron is defined by three distances $d_2 = \|\mathbf{m}_1 - \mathbf{m}_2\|$, $d_3 = \|\mathbf{m}_1 - \mathbf{m}_3\|$, and $d_4 = \|\mathbf{m}_1 - \mathbf{m}_4\|$, two height angles $\varphi_3 = \angle_{\mathbf{m}_2\mathbf{m}_1\mathbf{m}_3}$, $\varphi_4 =$

$\angle_{\mathbf{m}_2\mathbf{m}_1\mathbf{m}_4}$, and the azimuth angle $\lambda_4 = \angle_{\mathbf{a}_3\mathbf{m}_1\mathbf{a}_4}$, see Fig. 3.6. Signals arrive from the angles $\varphi_s = \angle_{\mathbf{m}_2\mathbf{m}_1\mathbf{s}}$ and $\lambda_s = \angle_{\mathbf{a}_3\mathbf{m}_1\mathbf{a}_s}$, uniquely determining the direction vector. Furthermore we define $\alpha = \angle_{\mathbf{m}_3\mathbf{m}_1\mathbf{m}_4}$ which we need later. The auxiliary points $\mathbf{a}_3$, $\mathbf{a}_4$, and $\mathbf{a}_s$ are projections of $\mathbf{m}_3$, $\mathbf{m}_4$, and $\mathbf{s}$ respectively, onto the plane orthogonal to $\mathbf{m}_1\mathbf{m}_2$ through $\mathbf{m}_1$.

Similar to the two-dimensional case we define the time differences $\tau_i = T_1 - T_i$ and the angles $\gamma_2 = \angle_{\mathbf{m}_2\mathbf{m}_1\mathbf{s}}$, $\gamma_3 = \angle_{\mathbf{m}_3\mathbf{m}_1\mathbf{s}}$, and $\gamma_4 = \angle_{\mathbf{m}_4\mathbf{m}_1\mathbf{s}}$. We obtain

$$\begin{aligned} x &= c\tau_2 = d_2\cos(\gamma_2) \\ y &= c\tau_3 = d_3\cos(\gamma_3) \\ z &= c\tau_4 = d_4\cos(\gamma_4) \end{aligned} \tag{3.17}$$

These triples (x, y, z) describe an ellipsoid in space $\mathbb{R}^3$. Combining the equations we derive the ellipsoid equation

$$(ax)^2 + (by + cy)^2 + (dx + ey + fz)^2 = 1\,, \tag{3.18}$$

where

$$\begin{aligned} a &= \frac{1}{d_2} \\ b &= -\frac{\cos(\varphi_3)}{d_2\sin(\varphi_3)} \\ c &= \frac{1}{d_3\sin(\varphi_3)} \\ d &= -\frac{\cos(\varphi_4)}{d_2\sin(\varphi_4)\sin(\lambda_4)} - \frac{\cos(\varphi_3)\cos(\lambda_4)}{d_2\sin(\varphi_3)\sin(\lambda_4)} \\ e &= \frac{\cos(\lambda_4)}{d_3\sin(\varphi_3)\sin(\lambda_4)} \\ f &= \frac{1}{d_4\sin(\varphi_4)\sin(\lambda_4)}\,. \end{aligned}$$

We compute the ellipsoid parameters by regression analogous to the two-dimensional case. For three dimensions a minimum of six signal sources is required for a unique solution. For higher numbers of signals the least squares method is used as described in Section 3.3.2. In case the receivers are not synchronized the minimum number of signals is increased to nine.

3.3.5 Covariance Representation

An elegant representation can be derived from the knowledge that an ellipsoid, which is formed by time differences between three receivers, respectively four receivers, corresponds to a covariance matrix [64]. If we know the covariance matrix, we can directly

extract the parameters that generate the ellipsoid. In the following we describe the procedure for space $\mathbb{R}^3$ with the motivation to replace the inconvenient notation in Section 3.3.4. The two-dimensional case is simpler and can be obtained by setting irrelevant terms to zero.

Recall that the mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$ of a distribution X in $\mathbb{R}^3$ are defined as

$$\boldsymbol{\mu} \;=\; \begin{pmatrix}\mu_1\\ \mu_2\\ \mu_3\end{pmatrix} = \mathrm{E}(X)\;, \qquad \boldsymbol{\Sigma} \;=\; \begin{bmatrix}\sigma_1^2 & v_1 & v_2\\ v_1 & \sigma_2^2 & v_3\\ v_2 & v_3 & \sigma_3^2\end{bmatrix} = \mathrm{E}(X^2) - \mathrm{E}(X)^2$$

For matrix $\boldsymbol{\Sigma}$ holds that $\mathbf{x}^T\boldsymbol{\Sigma}\mathbf{x} \geq 0$ for all $\mathbf{x} \in \mathbb{R}^3$, i.e. the matrix is positive semidefinite. The matrix characterizes a covariance ellipsoid, where all points on the ellipse have the same constant probability density [65]. We derive the following definition.

Definition 2 *The $k\boldsymbol{\Sigma}$-ellipsoid for covariance matrix $\boldsymbol{\Sigma}$ is the ellipsoid with center $\boldsymbol{\mu}$ where for all points $\mathbf{x}$ holds*

$$d_{Mah}(\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \;=\; \sqrt{(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})} \;=\; k\;.$$

The measure $d_{\mathrm{Mah}}(\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ is known as the Mahalanobis distance. The $\boldsymbol{\Sigma}$-ellipsoid is often used in probabilistic robotics for Gaussian distributed filters to visualize the uncertainty of an estimation. We propose the following theorem.

Theorem 2 *The time differences of distant signals arriving at four receivers $\mathbf{m}_1$, $\mathbf{m}_2$, $\mathbf{m}_3$, $\mathbf{m}_4$ in space $\mathbb{R}^3$ form a $3\hat{\boldsymbol{\Sigma}}$-ellipsoid with covariance matrix*

$$\hat{\boldsymbol{\Sigma}} \;=\; \frac{1}{3}\begin{bmatrix} d_2^2 & d_2 d_3\cos(\varphi_3) & d_2 d_4\cos(\varphi_4)\\ d_2 d_3\cos(\varphi_3) & d_3^2 & d_3 d_4\cos(\alpha)\\ d_2 d_4\cos(\varphi_4) & d_3 d_4\cos(\alpha) & d_4^2\end{bmatrix}\;.$$

Therefore, according to Theorem 2, the distances and angles in the tetrahedron of receivers are directly characterized by the coefficients of the covariance matrix. The distances are calculated by

$$d_2 = \sqrt{3}\,\sigma_1\;, \qquad d_3 = \sqrt{3}\,\sigma_2\;, \qquad d_4 = \sqrt{3}\,\sigma_3\;.$$

The angles of the tetrahedron are calculated from the correlation coefficients by

$$\cos(\varphi_3) = \frac{v_1}{\sigma_1\sigma_2}\;, \qquad \cos(\varphi_4) = \frac{v_2}{\sigma_1\sigma_3}\;, \qquad \cos(\alpha) = \frac{v_3}{\sigma_2\sigma_3}\;.$$

In order to prove Theorem 2 we require the following piece of information about covariance ellipsoids.

Lemma 4 *The covariance of a $\boldsymbol{\Sigma}$-ellipsoid in $\mathbb{R}^3$ is $\hat{\boldsymbol{\Sigma}} = \frac{1}{3}\boldsymbol{\Sigma}$. In the two-dimensional case is $\hat{\boldsymbol{\Sigma}} = \frac{1}{2}\boldsymbol{\Sigma}$ the covariance of a $\boldsymbol{\Sigma}$-ellipse.*

The lemma can be verified by integration over all points of the $\mathbf{\Sigma}$-ellipsoid. In the following we prove Theorem 2 and calculate the covariance of the ellipsoid of time differences.

Proof: According to the Eqns (3.17), the time differences in the Ellipsoid method depend on the distance between two receivers and on the angle of a signal. For simplicity, we assume that the receivers are synchronized. If they are not, i.e. the ellipse center is displaced from the origin, use the technique described in the next paragraph to synchronize the receivers and to center the ellipse, therefore eliminate the mean of the ellipsoid distribution of time differences.

Now, consider the continuous distribution of the synchronized time differences $X = \{(x, y, z)^T\}$ over uniformly distributed directions of origin. A uniform distribution of signal origins $\hat{\mathbf{s}}$ in space $\mathbb{R}^3$ with basis axes $(\hat{x}, \hat{y}, \hat{z})$ can be created by points

$$\hat{\mathbf{s}} \;=\; \begin{pmatrix} r\cos(\lambda) \\ r\sin(\lambda) \\ k \end{pmatrix} \;, \tag{3.19}$$

where $\lambda \in [0, 2\pi]$ and $k \in [-1, 1]$ are uniformly distributed random variables, and $r = \sqrt{1-k^2}$. By such a choice of λ and k holds that the points $\hat{\mathbf{s}}$ are uniformly distributed on the surface of the unit sphere. The density functions of the uniform distributions are $f(k) = \frac{1}{2}$ and $g(\lambda) = \frac{1}{2\pi}$, which can be combined to $h(\lambda, k) = (g \circ f)(\lambda, k) = \frac{1}{4\pi}$.

Without loss of generality, we align the tetrahedron such that $\mathbf{m}_1$ is the origin, $\mathbf{m}_1\mathbf{m}_2$ is parallel to the $\hat{z}$-axis, and $\mathbf{m}_3$ resides on the $\hat{x}/\hat{z}$-plane. Assuming that the sphere is large, i.e. the signals $\hat{\mathbf{s}}$ originate from far away, the incidence angles of the signals are

$$\lambda_\mathrm{s} = \lambda \qquad \text{and} \qquad \cos(\varphi_\mathrm{s}) = k \;. \tag{3.20}$$

By using spherical trigonometry we calculate the time differences with respect to the tetrahedron angles as follows

$$\begin{aligned} \frac{x}{d_2} \;&=\; \cos(\gamma_2) \;=\; \cos(\varphi_\mathrm{s}) \\ \frac{y}{d_3} \;&=\; \cos(\gamma_3) \;=\; \cos(\varphi_3)\cos(\varphi_\mathrm{s}) + \sin(\varphi_3)\sin(\varphi_\mathrm{s})\cos(\lambda_\mathrm{s}) \\ \frac{z}{d_4} \;&=\; \cos(\gamma_4) \;=\; \cos(\varphi_4)\cos(\varphi_\mathrm{s}) + \sin(\varphi_4)\sin(\varphi_\mathrm{s})\cos(\lambda_\mathrm{s} - \lambda_4) \;. \end{aligned} \tag{3.21}$$

Note that the incidence angles γ_2, γ_3, γ_4, are not uniformly distributed in the three-dimensional case, in contrast to the planar case. Furthermore, we express α in terms of the given angles as

$$\cos(\alpha) \;=\; \sin(\varphi_3)\sin(\varphi_4)\cos(\lambda_4) + \cos(\varphi_3)\cos(\varphi_4) \;. \tag{3.22}$$

Using the uniform distribution of signals (3.20) and the time differences (x, y, z) from Eqns. (3.21) that follow, we calculate the covariance matrix by integration as follows:

$$\begin{aligned}
\hat{\mathbf{\Sigma}} &= \int_0^{2\pi}\int_{-1}^{1} X^2\, h(\gamma,k)\, \mathrm{d}k\, \mathrm{d}\lambda \\
&= h(\gamma,k) \int_0^{2\pi}\int_{-1}^{1} \begin{pmatrix} x \\ y \\ z \end{pmatrix}^2 \mathrm{d}k\, \mathrm{d}\lambda \\
&= h(\gamma,k) \int_0^{2\pi}\int_{-1}^{1} \begin{bmatrix} x^2 & xy & xz \\ xy & y^2 & yz \\ xz & yz & z^2 \end{bmatrix} \mathrm{d}k\, \mathrm{d}\lambda \\
&\overset{(3.20)-(3.22)}{=} \frac{1}{3} \begin{bmatrix} d_2^2 & d_2 d_3 \cos(\varphi_3) & d_2 d_4 \cos(\varphi_4) \\ d_2 d_3 \cos(\varphi_3) & d_3^2 & d_3 d_4 \cos(\alpha) \\ d_2 d_4 \cos(\varphi_4) & d_3 d_4 \cos(\alpha) & d_4^2 \end{bmatrix} . \qquad (3.23)
\end{aligned}$$

It remains to show that the distribution of time differences is actually an ellipsoid. From Eqns. (3.11) and (3.18) we see that the time difference tuples describe a quadric hypersurface in space $\mathbb{R}^2$, respectively $\mathbb{R}^3$. As time differences are bounded by the distance of the receivers the type of quadric is restricted to ellipsoids. Other quadric shapes that are described by such polynomials, such as hyperboloids and paraboloids, are not bounded, therefore do not apply. We can affirm this fact by verifying that $d_{\mathrm{Mah}}(\mathbf{x}, \vec{0}, \hat{\mathbf{\Sigma}}) = 1$ as defined in Def. 2 for all time differences $\mathbf{x} = (x, y, z)^T$. □

3.3.6 Transformation of the Covariance Matrix

We describe now the transformation of parameters from the regression polynomial to the parameters of the covariance matrix, which is done by basic math [64]. We show the transformation exemplarily in three dimensions.

In space $\mathbb{R}^3$ an ellipsoid can be represented by the matrix form

$$\mathbf{x}^T \mathbf{\Sigma}^{-1} \mathbf{x} \;=\; 1\,, \qquad (3.24)$$

where $\mathbf{x} = (x_1, x_2, x_3)^T$ is a vector and $\mathbf{\Sigma}$ is a symmetric positive definite matrix

$$\mathbf{\Sigma} \;=\; \begin{bmatrix} \sigma_1^2 & v_1 & v_2 \\ v_1 & \sigma_2^2 & v_3 \\ v_2 & v_3 & \sigma_3^2 \end{bmatrix} .$$

Under the assumption of a zero-mean ellipsoid, i.e. the receivers are synchronized, the regression of a system of ellipsoid polynomial equations

$$a x_1^2 + b x_2^2 + c x_3^2 + d x_1 x_2 + e x_1 x_3 + f x_2 x_3 = 1 \qquad (3.25)$$

yields ellipsoid parameters a to f. We convert the parameter set and extract the parameters of the covariance matrix by substitution in Eq. (3.24) by

$$\begin{aligned} Z &= be^2 + cd^2 + af^2 - 4abc - def \\ \sigma_1^2 &= (f^2 - 4bc) \,/\, Z \\ \sigma_2^2 &= (e^2 - 4ac) \,/\, Z \\ \sigma_3^2 &= (d^2 - 4ab) \,/\, Z \\ v_1 &= (2cd - ef) \,/\, Z \\ v_2 &= (2be - df) \,/\, Z \\ v_3 &= (2af - de) \,/\, Z \end{aligned} \tag{3.26}$$

In case the receivers are not synchronized, the ellipsoid is shifted to zero-mean by converting the general ellipsoid polynomial equation to a translation-invariant form. In three dimensions the general form is

$$ax_1^2 + bx_2^2 + cx_3^2 + dx_1x_2 + ex_1x_3 + fx_2x_3 + gx_1 + hx_2 + jx_3 = 1 \,. \tag{3.27}$$

Convert to the following translation-invariant form

$$\begin{aligned} &\hat{a}(x_1 - \hat{u})^2 + \hat{b}(x_2 - \hat{v})^2 + \hat{c}(x_3 - \hat{w})^2 \\ &+ \hat{d}(x_1 - \hat{u})(x_2 - \hat{v}) + \hat{e}(x_1 - \hat{u})(x_3 - \hat{w}) + \hat{f}(x_2 - \hat{v})(x_3 - \hat{w}) \;=\; 1 \,. \end{aligned} \tag{3.28}$$

Calculating $\hat{a}$ to $\hat{f}$ and $\hat{u}$, $\hat{v}$, $\hat{w}$ from the coefficients of Eq. (3.27) is a lengthy equation. It can be calculated using a computer algebra software by expansion of Eq. (3.28) and substitution of the constant term. The coefficients $\hat{a}$ to $\hat{f}$ are converted for the covariance matrix using Eqns. (3.26). The coefficient vector $(\hat{u}, \hat{v}, \hat{w})^T$ equals the center point of the ellipse and the synchronization offset of the receivers.

Once the coefficients of the covariance matrix are known, the distances and angles of the tetrahedron of receivers can be calculated as described. As the lengths of the three edges and the three angles are known, the tetrahedron is rigid. The construction of a triangle in two-dimensional space is a simpler subproblem where the z-term of the transformation is omitted.

3.3.7 Issues of Regression

The direct computation of the ellipsoid using least squares regression is a simple and fast approach, yet it is not trouble-free in terms of robustness. In some cases the regression fails to return an elliptic curve, which is due to the fact that the polynomial Eq. (3.25), respectively Eq. (3.13) in $\mathbb{R}^2$, does not only represent an ellipsoid but, depending on the parameters a to f, may represent a different quadric hypersurface.

In [66] an iterative scheme for the two-dimensional case is proposed to minimize the Euclidean distance between points and the ellipse, however this is computationally

intensive. From Eq. (3.11) one can derive a side condition to test if the result of the regression is successful. Eq. (3.13) describes an ellipse in $\mathbb{R}^2$ if $\mathcal{K} = \frac{1}{4ab-c^2} > 0$. One can incorporate such a side condition into a linear regression scheme, as described in [67] for ellipses. In three dimensions the case is more complex. Inspired by the approach in Section 3.3.5 one may calculate the eigenvalues of the covariance matrix, which must all be greater or equal to zero, however this is a complicated side condition to include into regression. A closed form approach for three dimensions is described in [68] for ellipsoids. This may solve the regression issue in most cases.

As an alternative we characterize the ellipsoid by calculating the mean and the covariance of the time difference measures. In contrast to a regression matrix, such an actual covariance matrix is dependent on the distribution of signals, and therefore only an approximation. Consider a distribution of m time differences $X = \{\mathbf{x}_j\} = \{(x_j, y_j, z_j)^T\}$ that approximately describe an ellipsoid. We calculate the mean $\tilde{\boldsymbol{\mu}}$ and covariance $\tilde{\boldsymbol{\Sigma}}$ of the set of time differences by

$$\tilde{\boldsymbol{\mu}} \;=\; \frac{1}{m}\sum_{j=1}^{m}\mathbf{x}_j \;, \qquad \tilde{\boldsymbol{\Sigma}} \;=\; \frac{1}{m}\sum_{j=1}^{m}\Big(\mathbf{x}_j - \tilde{\boldsymbol{\mu}}\Big)\Big(\mathbf{x}_j - \tilde{\boldsymbol{\mu}}\Big)^T \;.$$

The $3\tilde{\boldsymbol{\Sigma}}$-ellipsoid is an approximation to the ellipsoid of time differences comparable to the $3\hat{\boldsymbol{\Sigma}}$-ellipsoid from the previous section. Again, we extract the distances and angles as described. The time offsets equal the mean of the time differences $(\delta_1, \delta_2, \delta_3) = \tilde{\boldsymbol{\mu}}^T$. In case the receivers are synchronized we set $\tilde{\boldsymbol{\mu}} = (0, 0, 0)^T$. Alternatively, we obtain from $\tilde{\boldsymbol{\Sigma}}$ the parameter vector that describes the shape of the ellipsoid for the "old" approach by an inverse transformation of the coefficients.

3.3.8 Numerical Evaluation

For the Ellipsoid TDoA method we assume that signals originate from so far away that the direction vectors of arrival at different receivers become identical. If this assumption is violated, one may suppose that errors are induced when calculating the receivers. Another source of error is timing uncertainty during detection of the signals by the receivers, the *TDoA error* that we assume Gaussian distributed. We have run numerical simulations in space $\mathbb{R}^2$ in an computer algebra system to quantify the behavior of the algorithm [61]. In these tests, we vary parameters such as the distance of signal sources and the error of timing.

For every test cycle we randomly placed three receivers on a circle around the origin with a radius of 2.3 m. By this setting, we obtain a large number of different triangle shapes, in particular they are not necessarily equal-sided or equilateral.

Now we simulate m senders located on a circle with a specific radius, emitting a signal in a straight line towards every receiver $\mathbf{m}_i$. The signals arrive at times T_{ij} according to the signal propagation equation

$$T_{ij} = \tfrac{1}{c}\|\mathbf{m}_i - \mathbf{s}_j\| + \delta_i + \epsilon_i \;, \tag{3.29}$$

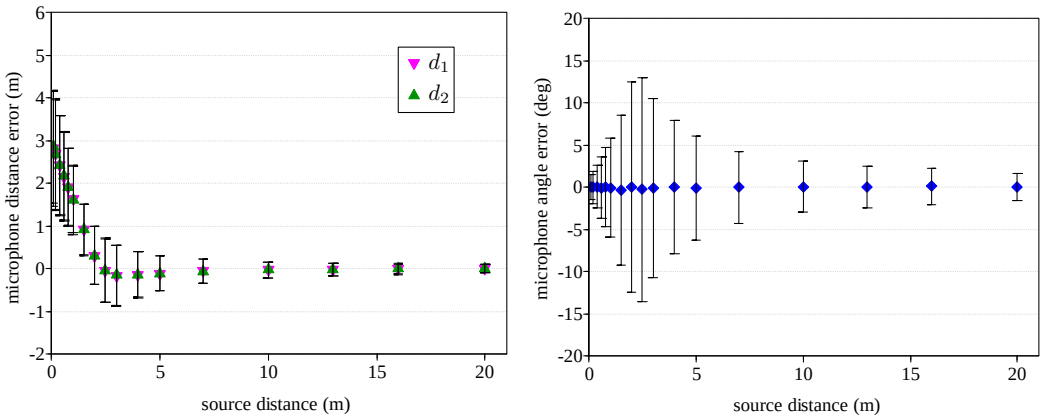


Figure 3.7: Evaluation of 12 signals arriving at three unsynchronized receivers in the plane. The mean errors of the distance estimation d_1 and d_2 (*left*) and of the angle α (*right*) diminish, if the signals are just about twice as far away as the receivers from each other. The black bars denote the standard deviation. [61]

where $i = \{1, 2, 3\}$ and $1 \leq j \leq m$. Here, $\mathbf{s}_j$ is the location of the sender j on the circle, δ_i is the offset of the receiver clock, simulating unsynchronized receivers, and ϵ_i is a Gaussian error variable, simulating timing uncertainty of receivers. From the absolute values we calculate tuples of time differences $(\tau_1, \tau_2)_j = (T_{2j} - T_{1j}, T_{3j} - T_{1j})$. As we assume unsynchronized receivers, the ellipse described by the m tuples is shifted from the origin.

We ran a set of 1000 cycles with different distances of 12 signal sources and analyzed the errors of the estimated distances d_1, d_2, and angle α, compared to the true values. In the evaluation we see that if the signals occur from a distance of about 8 m on, which is only about twice the distance of the receivers, the errors of d_1 and d_2 diminish (Fig. 3.7). Similar behavior was observed for the angle α where the standard deviation of errors clearly decreases.

However, if the signals occur from very close, far-field assumption of the signals is violated, and the distances d_1 and d_2 are clearly underestimated. Interesting is also the case when the signal radius is only slightly larger than the circle on which the receivers reside. Then, the shape described by the signals is deformed, as shown in Fig. 3.8.

Furthermore, we have run experiments with simulated Gaussian timing error, now with the signals originating from very far away. Due to the regression of m signal measurements the algorithm is comparatively robust against these errors, in contrast to for instance the min/max-method from the previous section. Given a sufficient number of at least 8 signals, and reasonable TDoA errors, we obtain estimation errors is in the range of centimeters. For details on the TDoA error evaluation see [61].

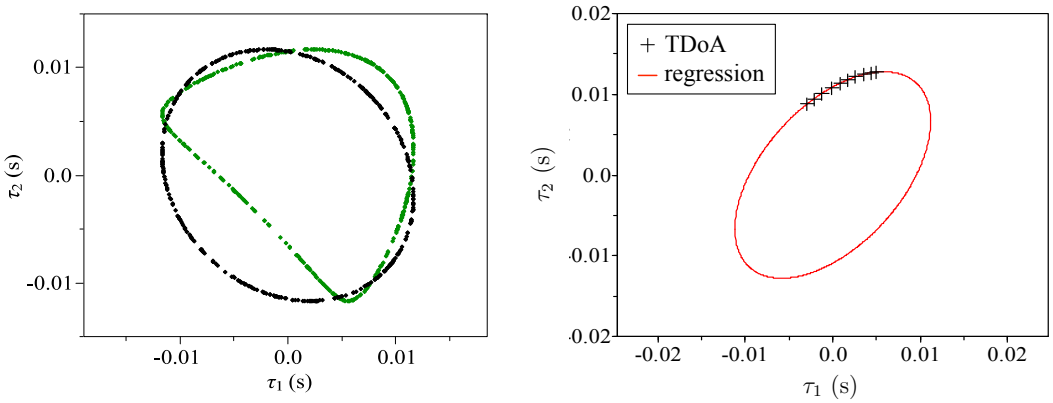


Figure 3.8: *Left*: If the signals originate from close distance (here, a radius of 5 m), the shape is deformed (green), compared to the correct ellipse from distant signals (black). *Right*: Signals in a real-world experiment arriving only from a small sector, from a distance of 13 m. The correct ellipse can still be calculated. [61]

3.3.9 Real-World Evaluation

Using a network of connected laptops and Apple iPhones we have executed some experiments in an indoor and an outdoor setting. In these experiments we recorded short noisy audio events via the built-in microphones of the laptops, and detect the time of the signals. The software we used is described in Section 7.1.

In the indoor experiment we demonstrated that the Ellipsoid method does not rely on well-distributed signal origins, but is feasible even if signals originate only from a small sector, see Fig. 3.8. For more details of the indoor experiment refer to [61].

In a second experiment we distributed four laptops and four iPhones randomly on a green field on our campus and measured their positions by hand as a reference, see Fig. 3.9. A pedestrian walked around the group of devices, starting in the vicinity, up to a distance of 80 m, generating noisy clap signals with two wooden bars every few meters. These were recorded by the devices and exchanged in the network. With the obtained TDoA data the algorithm calculated the distances of the eight devices in groups of three using the Ellipsoid TDoA method. For $n = 8$ devices we selected $n(n-1)(n-2) = 336$ subsets and calculated the edges, which includes of course many symmetric cases. Given these distances, we calculated the coordinates of the devices by optimization of

$$\arg\min_{\mathbf{m}_i, \mathbf{m}_k} \sum_{i,k}^{n} \left(\|\mathbf{m}_i - \mathbf{m}_k\| - \tilde{d}_{ik} \right)^2 ,$$

where $\tilde{d}_{ik}$ is the estimated distance between receiver i and k. We obtained an error of the estimated receiver positions of only 0.38 m ($\sigma = 0.14$ m).

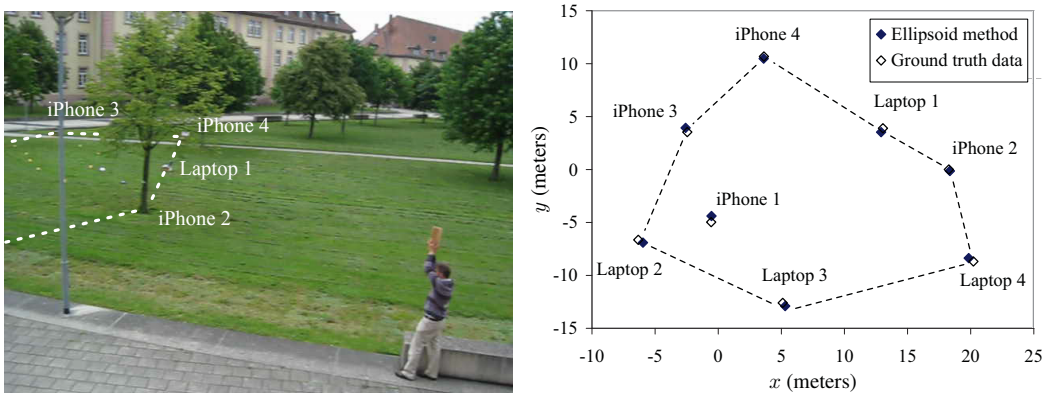


Figure 3.9: Outdoor experiments. *Left*: A pedestrian walks around a group of four iPhones and four randomly placed laptops, clapping two wooden bars. *Right*: The error of the estimated receiver positions is only 0.38 m ($\sigma = 0.14$ m). [61]

3.4 Coordinates From Distances

A common characteristic of the previously presented approximation approaches and of the Ellipsoid TDoA method is that only distances are calculated, not the Cartesian coordinates between receivers in the plane or in three-dimensional space. Similarly, in localization approaches based on the time of arrival (ToA) or the round-trip-time of a signal, the absolute distance to a target is calculated by the signal velocity. Or in case of using the received signal strength indication (RSSI) of radio signals, the distance between two antennas is estimated by a path loss model based on the Friis transmission equation.

Consider the distance problem as a problem of multidimensional scaling (MDS), where the Euclidean norm is used as a metric. Given is a graph of n nodes, and distances d_{ij} between nodes i and j, where $1 \leq i, j \leq n$. We define the MDS problem as a system of equations

$$\|\mathbf{x}_i - \mathbf{x}_j\| \approx d_{ij} \ , \tag{3.30}$$

where node positions $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^p$ are an approximation to the equation system. In our case we choose $p \in \{2, 3\}$ for the dimension.

3.4.1 Classical Multidimensional Scaling

In the classical MDS approach a solution of point vectors $\mathbf{x}_i$ is calculated by matrix decomposition of a matrix of distances $\mathbf{D} = [d_{ij}]$ $(i, j \leq n)$, which is self-similar $(d_{ii} = 0)$ and symmetric $(d_{ij} = d_{ji})$. By choosing the $p < n$ largest vectors of eigenfactorization, an embedding of $\mathbf{x}_i$ into p-dimensional space is obtained.

We follow the description of MDS given in [19]. First, calculate the Gramian matrix of $\mathbf{D}$, for which the row mean and column mean of $\mathbf{D}$ is subtracted, and the overall

mean is added. This is done by a matrix $\mathbf{H} = \mathbf{I}_{n\times n} - \frac{1}{n}\vec{1}_{n\times n}$, where $\mathbf{I}$ is the identity matrix and $\vec{1}$ is a matrix of values 1, and the operation $\mathbf{A} = -\frac{1}{2}\mathbf{H}\mathbf{D}^2\mathbf{H}$. Now, calculate the spectral decomposition $\mathbf{A} \overset{\text{(spec)}}{=} \mathbf{U}\mathbf{\Sigma}\mathbf{U}^T$, which yields an orthogonal matrix of eigenvectors $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_n)$, and the diagonal matrix $\mathbf{\Sigma}$, containing the eigenvalues λ_i. Choose the p largest eigenvalues $\lambda_{k_1} > \ldots > \lambda_{k_p}$, and their according eigenvectors, and assemble a matrix

$$\mathbf{X}_{p\times n} = \left(\sqrt{\lambda_{k_1}}\,\mathbf{u}_{k_1}, \ldots, \sqrt{\lambda_{k_p}}\,\mathbf{u}_{k_p}\right)^T .$$

The point vectors $\mathbf{x}_i$ that we obtain from $\mathbf{X}$, are the optimal linear approximation to (3.30) in the least squares sense [19]. The MDS method is fast, compared to iterative methods, as it can be calculated at most in time $\mathcal{O}(n^3)$, which is required for the matrix decomposition. However, the algorithm does not yield a solution if the graph of measurements is incomplete, as the case of missing measurements d_{ij} cannot be handled in the distance matrix $\mathbf{D}$. In cases of incomplete measurement graphs we require an alternative.

3.4.2 Iterative Construction

In a naive approach we iteratively build the network of n nodes from given distances d_{ij} between nodes i and j [60]. Such an algorithm is feasible for *trilateration graphs* [69], i.e. rigid graphs with minimum degree $p+1$ in space $\mathbb{R}^p$. For simplicity we demonstrate and discuss the approach in two-dimensional space, however it can be extended to three dimensions.

We start by choosing two arbitrary nodes i and j, probably the nodes with the largest distance, and connect them. We place the first node $\mathbf{x}_i$ at the origin and the second node $\mathbf{x}_j$ at $(d_{ij}, 0)$. Now, we choose a third node k and calculate the two solutions $\mathbf{x}_k = (\cos(\alpha)\, d_{ij}, \pm\sin(\alpha)\, d_{ij})$, where $\cos(\alpha) = \frac{d_{ij}^2 + d_{ik}^2 - d_{jk}^2}{2 d_{ij} d_{ik}}$. Without loss of generality, we choose the node in the positive quadrant. We know three nodes now, so any rotation and mirror symmetries of the graph are eliminated, if the nodes are in general position.

To calculate the next node l, we choose any two of the previous nodes $\hat{i}$ and $\hat{j}$ and solve the equation system $\|\mathbf{x}_l - \mathbf{x}_{\hat{i}}\| = d_{l\hat{i}}$ and $\|\mathbf{x}_l - \mathbf{x}_{\hat{j}}\| = d_{l\hat{j}}$. By transforming the coordinates, such that $\mathbf{x}_{\hat{i}}$ is the origin and $\mathbf{x}_{\hat{i}}$ resides on the x-axis, we can directly calculate the two solutions $\mathbf{x}_l^{(1)}$ and $\mathbf{x}_l^{(2)}$ of the quadratic system. By comparing the distance errors $\|\mathbf{x}_l^{(1)} - \mathbf{x}_{\hat{k}}\| - d_{l\hat{k}}$ and $\|\mathbf{x}_l^{(2)} - \mathbf{x}_{\hat{k}}\| - d_{l\hat{k}}$ of another already known node $\hat{k}$ we can decide, which of the two solutions is correct, if the error of distance measures is sufficiently small. In this way, we can iteratively calculate all other nodes and build the complete network in time $\mathcal{O}(n)$. Besides from the fast calculation, one may calculate an upper bound of the error by describing the loss of localization quality in every iteration [60].

3.4.3 Bipartite Distance Graphs

Unfortunately, the iterative construction algorithm is not feasible in the time-of-arrival problem of distinct senders and receivers. Here, the graph structure is a bipartite graph $\mathcal{G}_{n,m} = (M, S, E)$, where the vertices are disjoint sets $M = \{\mathbf{m}_i\}$ and $S = \{\mathbf{s}_j\}$. The vertices are connected by edges $(\mathbf{m}_i, \mathbf{s}_j)$, iff we know the distance d_{ij} $(1 \leq i \leq n, 1 \leq j \leq m)$.

Bipartite graphs can be rigid and unique, then only one solution exists to equation system (3.30), except for congruent transformations. According to [4] and [70] they are rigid for general positions of vertices if $n, m \geq 3$. However, as a general property of bipartite graphs, they do not contain cycles of odd length, so no initial triangle can exist for iterative construction, from which we continue to the other nodes.

One may resort to iterative optimization as an alternative. In case of the times-of-arrival problem, which yields a bipartite graph of distances, neither classical multidimensional scaling is feasible, as the graph is not complete, nor can we use iterative construction of the node positions. Instead, optimization is feasible for a problem of the form

$$\arg\min_{\mathbf{m}_i, \mathbf{s}_j} \sum_{i=1}^{n} \sum_{j=1}^{m} \left(\|\mathbf{m}_i - \mathbf{s}_j\| - d_{ij} \right)^2 ,$$

where d_{ij} are distance measurements between a sender $\mathbf{s}_j$ and a receiver $\mathbf{m}_i$.

For optimization methods may be used that rely on the first-order derivative such as gradient descent or spring relaxation, where d_{ij} represents the target length of a spring. Both approaches are discussed in detail for the problem of hyperbolic localization in the next chapter, which is a related problem. A framework for general graph optimization is presented in [71].

In both problems, distance multilateration and hyperbolic localization, the existence of a unique solution and the probability to miss the global minimum of the error function are an issue. For the problem of distance graphs the conditions for rigidity [4, 69, 70] and the problem of a solution in large graphs [25, 26] have been thoroughly discussed in literature.

A closed-form approach was suggested by Pollefeys [42], who proposed an approach based on matrix factorization to calculate the receiver and signal positions, once the signal times are known, which renders the problem effectively a bipartite graph problem. Yet the algorithm seems to induce additional error of a factor of ten into the calculated positions. Another approach was presented by Crocco et al. [72, 73] where they assume a signal position to coincide with a receiver position.

In a recent approach [74] the ToA bipartite graph problem is analyzed in detail and the minimum problem is solved in closed form. Such linear approaches tend to induce further error into the graph, so it is advisable to refine the results by non-linear least-squares optimization.

3.5 Summary

Approximative approaches provide a fast answer to the problem of time differences of arrival. In all of the proposed algorithms receiver positions are calculated in polynomial time, without the need for cumbersome iteration or search algorithms that require vast amounts of CPU time and energy. Once the receivers are determined, either the directions or the positions of signals can be calculated, using trigonometry or ordinary TDoA multilateration, respectively. Furthermore, a statistical approximation may be available for only two receivers, a far-field approximation requires only three, respectively four receivers, depending on the dimension. With small effort approximation algorithms can be applied when no synchronization between receivers is available.

When we can rely on certain assumptions, such as the distance and distributions of signals, or even just a high number of arbitrary signals, we have demonstrated that approximation algorithms achieve high precision of localization. In a real-world experiment using the Ellipsoid TDoA method we have obtained the positions of laptops up to an error of 38 cm, which is less than 2 % in an area of 25×20 m. Even if the assumptions on the signals are not perfectly satisfied, we obtain reasonable results.

And still, approximation is not a solution to the general TDoA problem. For instance, the minimum case of four receivers and only few signals at general positions, as described in Section 2.4, is not well-treated by the proposed algorithms. In the next chapter we discuss iterative optimization as an approach that yields a solution to general settings of the TDoA problem in most cases.

4 Local Optimization

The quality of approximative solutions under the far-field assumption degrades if signals are rare or in the vicinity of receivers. In the following we address the problem of finding an exact solution to the TDoA equation system. Because of the high number of at least eighteen quadratic equations that are required for unique determination of the equation system, and the large number of quadratic terms, the existence of an algebraic formulation of the solution is doubtful. Due to the non-linearity and non-convexity of the equation system we use non-linear iterative optimization to approximate a numerical solution up to floating point precision.

We present two algorithm classes of local optimization to conquer the problem. The first two algorithms are based on evaluation of the gradient of the error function, where we derive the Jacobian matrix for anchor-free TDoA localization. The gradient descent method and the Gauss-Newton algorithm that rely on the Jacobian are the standard approaches to non-linear optimization. Many sophisticated optimization algorithms are based on these, such as the Levenberg-Marquardt [75] and the Gelfand algorithm [76]. Second, we propose an iterative algorithm tailored to the TDoA problem which is based on a physical mass-spring simulation. The *Iterative Cone Alignment* optimizes a TDoA error function by relaxation of springs that represent the time constraints of the error function.

4.1 First-Order Gradient Methods

An common approach of local optimization to the TDoA problem is iterative update based on the gradient of an error function. We introduce gradient descent and the Gauss-Newton method, two first-order methods that use the derivative of the system of hyperbolic error equations, the Jacobian matrix.

4.1.1 The Gradient Descent Method

Such methods have been considered in literature, regarding this problem. Biswas & Thrun represent the problem of signal and receiver localization using TDoA as a Bayesian network [41]. They assume a Gaussian model of measurement errors and optimize the signals and receivers as parameters in a likelihood maximization manner. Because of monotony considerations, and because of the assumption of Gaussian probability, the probabilistic model effectively reduces to a problem of minimization of

quadratic error, cf. Eq. 8 in the paper [41][1]. However, we were not satisfied with the comment of the authors that "gradient descent may get stuck in local minima", and we felt that deeper insight into the problem is required.

Biswas & Thrun consider the quadratic objective

$$\underset{\mathbf{m}_i,\mathbf{s}_j,t_j}{\arg\min} \sum_{i=1}^{n}\sum_{j=1}^{m} \left(\|\mathbf{m}_i - \mathbf{s}_j\| - c\,(T_{ij} - t_j)\right)^2 \ ,$$

where they pursue to find solutions for receivers $\mathbf{m}_i$, senders $\mathbf{s}_j$ in space $\mathbb{R}^p$, where $p = \{2, 3\}$, and send times of the signals t_j. They obtain an equation system of $(p+1)m + pn$ unknown variables and mn equations or less, if signals were missed by one or more receivers.

We use a hyperbolic system of equations instead, which we obtain by combining the i-th and the first equation, see [5, 54] and Section 2.2. In this way the signal time is eliminated, reducing the system by m variables and m equations, so we obtain $p(m+n)$ unknowns and a maximum of $m(n-1)$ equations. We obtain a system of hyperbolic equations of the form

$$f_{ij} \ = \ \|\mathbf{m}_i - \mathbf{s}_j\| - \|\mathbf{m}_1 - \mathbf{s}_j\| - c\,(T_{ij} - T_{1j}) \ , \tag{4.1}$$

where $2 \leq i \leq n$. Reduction of the equation system is an important step to increase the probability of correct convergence, as some false minima of misleading send times disappear, see Section 4.3.4. We combine the equations and obtain the quadratic objective

$$\underset{\mathbf{m}_i,\mathbf{s}_j}{\arg\min} \sum_{i=2}^{n}\sum_{j=1}^{m} \left(f_{ij}\right)^2 \ ,$$

which is proportional to the objective in vector notation $w = \frac{1}{2}\mathbf{b}^T\mathbf{b}$, where $\mathbf{b} = (f_{21}, f_{22}, \ldots, f_{nm})^T$.

We define a state vector of the form $\mathbf{u} = \left(\mathbf{s}_{1,1}^T, \ldots, \mathbf{s}_{m,p}, \mathbf{m}_{2,1}^T, \ldots, \mathbf{m}_{n,p}^T\right)^T$, where $\mathbf{m}_{i,\ell}, \mathbf{s}_{j,\ell}$ denotes the ℓ-th scalar of $\mathbf{m}_i, \mathbf{s}_j$. The operator $(\,\cdot\,)^T$ denotes the transposition. We initialize $\mathbf{u}^{(0)}$ with randomized values around the origin. Alternatively, the result of an approximation algorithm from the previous chapter could be used.

In every iteration step k we calculate the first-order derivative of the equation system, the Jacobian matrix $\mathbf{Q}$ and the function vector $\mathbf{b}$ from the input values of $\mathbf{u}^{(k)}$. The Jacobian is defined as

$$\mathbf{Q} = \begin{bmatrix} \frac{\partial f_{21}}{\partial \mathbf{s}_{1,1}} & \cdots & \frac{\partial f_{21}}{\partial \mathbf{s}_{m,p}} & \frac{\partial f_{21}}{\partial \mathbf{m}_{2,1}} & \cdots & \frac{\partial f_{21}}{\partial \mathbf{m}_{n,p}} \\ \frac{\partial f_{22}}{\partial \mathbf{s}_{1,1}} & \cdots & \frac{\partial f_{22}}{\partial \mathbf{s}_{m,p}} & \frac{\partial f_{22}}{\partial \mathbf{m}_{2,1}} & \cdots & \frac{\partial f_{22}}{\partial \mathbf{m}_{n,p}} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_{nm}}{\partial \mathbf{s}_{1,1}} & \cdots & \frac{\partial f_{nm}}{\partial \mathbf{s}_{m,p}} & \frac{\partial f_{nm}}{\partial \mathbf{m}_{2,1}} & \cdots & \frac{\partial f_{nm}}{\partial \mathbf{m}_{n,p}} \end{bmatrix} \ ,$$

[1] Apparently, a minus sign is missing in Eq. (8) and in the previous Gaussian density function

where the entries of $\mathbf{Q}$ are the partial derivatives of Eq. (4.1):

$$\begin{aligned} \frac{\partial f_{ij}}{\partial \mathbf{s}_{j,\ell}} &= -\frac{\mathbf{m}_{i,\ell} - \mathbf{s}_{j,\ell}}{\|\mathbf{m}_i - \mathbf{s}_j\|} + \frac{\mathbf{m}_{1,\ell} - \mathbf{s}_{j,\ell}}{\|\mathbf{m}_1 - \mathbf{s}_j\|} \\ \frac{\partial f_{ij}}{\partial \mathbf{m}_{i,\ell}} &= \frac{\mathbf{m}_{i,\ell} - \mathbf{s}_{j,\ell}}{\|\mathbf{m}_i - \mathbf{s}_j\|} \,. \end{aligned} \tag{4.2}$$

We calculate the direction of the steepest *ascent*

$$\begin{aligned} \nabla w &= \nabla\left(\frac{1}{2}\mathbf{b}^T\mathbf{b}\right) = \nabla\left(\frac{1}{2}\sum_{i,j}(f_{ij})^2\right) \\ &= \begin{pmatrix} \vdots \\ \frac{\partial\left(\frac{1}{2}\sum_{i,j}(f_{ij})^2\right)}{\partial u_h} \\ \vdots \end{pmatrix} = \begin{pmatrix} \vdots \\ \sum_{i,j}\frac{\partial\left(\frac{1}{2}(f_{ij})^2\right)}{\partial u_h} \\ \vdots \end{pmatrix} = \begin{pmatrix} \vdots \\ \sum_{i,j}\left(\frac{\partial f_{ij}}{\partial u_h} f_{ij}\right) \\ \vdots \end{pmatrix} \\ &= \begin{bmatrix} \vdots & \\ \cdots & \frac{\partial f_{ij}}{\partial u_h} \quad \cdots \\ \vdots & \end{bmatrix}^T \begin{pmatrix} \vdots \\ f_{ij} \\ \vdots \end{pmatrix} = \mathbf{Q}^T\mathbf{b} \,, \end{aligned} \tag{4.3}$$

where u_h is the h-th element of $\mathbf{u}^{(k)}$. The update vector is $\hat{\mathbf{u}} = \gamma\,\nabla w = \gamma\,\mathbf{Q}^T\mathbf{b}$, where γ is an adaptive factor for the step width. We proceed one step in direction of the steepest *descent* by updating the state $\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} - \hat{\mathbf{u}}$. The algorithm is repeated until $\|\hat{\mathbf{u}}\| < \epsilon_{\text{target}}$.

4.1.2 The Gauss-Newton Algorithm

For speedup we have added an extension to the optimization algorithm, and switch to the fast Gauss-Newton algorithm after some time, reducing the number of iterations when the gradient has become small during gradient descent optimization. The scheme of the Gauss-Newton algorithm is identical to the one of gradient descent, except for the calculation of the update. For the Gauss-Newton update we write the system of non-linear equations of the form (4.1) as a least squares equation system in matrix notation, and solve in every iteration step for for $\hat{\mathbf{u}}$:

$$\mathbf{Q}^T\mathbf{Q}\hat{\mathbf{u}} = \mathbf{Q}^T\mathbf{b} \quad \Rightarrow \quad \hat{\mathbf{u}} = (\mathbf{Q}^T\mathbf{Q})^{-1}(\mathbf{Q}^T\mathbf{b}) \tag{4.4}$$

The equation system (4.4) is usually solved by calculating the inverse $(\mathbf{Q}^T\mathbf{Q})^{-1}$. However, we recommend using LU or QR decomposition for higher numerical stability.

As the Gauss-Newton algorithm is very prone to divergence when applied to random initial positions, we do not use this method from the beginning. Instead, we rely on the robust gradient descent until the error function has become steady. However, when the

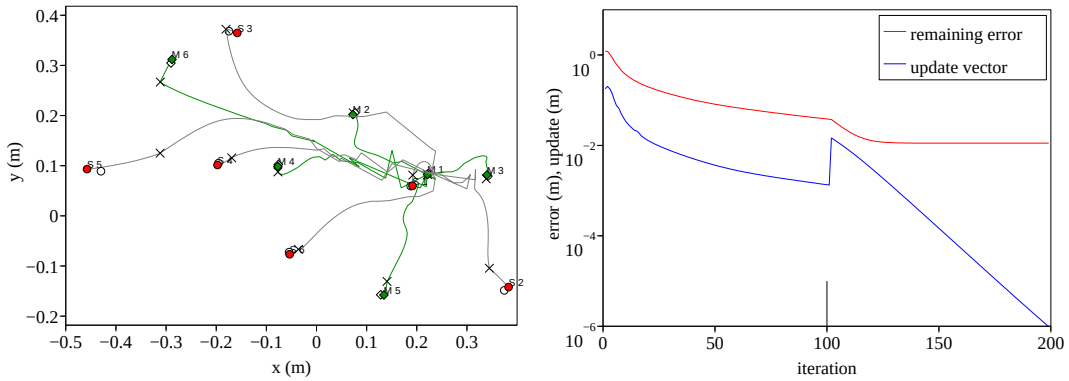


Figure 4.1: Runtime example of gradient descent and the Gauss-Newton method with six receivers (green) and six signals (red). The true positions are marked in white. After 100 iterations (marked as "×" in the left plot) the algorithm switches to Gauss-Newton. Because of random TDoA error of 1 % the remaining error does not reduce to zero.

gradient is shallow later during optimization, convergence in gradient descent tends to be very slow. Then, we switch to the fast the Gauss-Newton algorithm, making use of the quadratic convergence and reducing the number of iterations notably. As described later, we benefit from the Gauss-Newton algorithm not only after the gradient descent method, but also for finalization of the Cone Alignment algorithm. In particular, this leads to higher probability of convergence in case of badly scaled scenarios.

In experimental runs of the gradient method we saw that we could solve many scenarios with this combination of algorithms, see Fig. 4.1 for an example. However, some scenarios that run into local minima could not be solved even with repeated attempts. In the next section we present an algorithm based on the relaxation of springs which can achieve higher probability of convergence, and which we compare empirically to the gradient based method.

4.2 Iterative Cone Alignment

Many iterative approaches to the problem of TDoA use gradient descent, which is prone to local minima, or the Gauss-Newton method, which is prone to divergence. Modifications of the gradient method with "momentum" require scenario-dependent adjustments of parameters. Furthermore, the calculation and inversion of the Jacobian matrix is an expensive operation, for which high numeric precision is required in implementations.

We present an iterative approach to local optimization, the Cone Alignment algorithm [5, 54, 77], which uses a geometric representation of the error function and a physical mass-spring simulation to optimize the function for the general case.

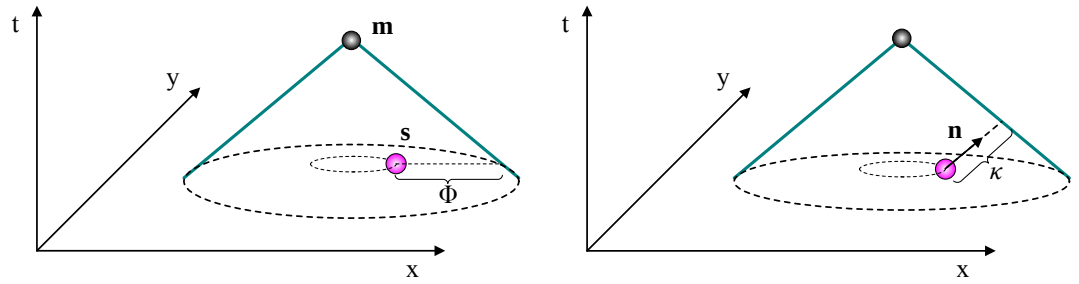


Figure 4.2: Cone representation of Eq. (4.5) for $p = 2$. *Left*: Signal source $\mathbf{s}_j$ resides offside the cone surface of receiver $\mathbf{m}_i$ and therefore it is not *valid* and $\Phi \neq 0$. *Right*: In order to restore validity $\mathbf{s}_j$ has to be moved in direction of $\mathbf{n}_{ij}$. [5]

Consider a receiver $\mathbf{m}_i$ and a signal origin $\mathbf{s}_j$ in $p \in \{2, 3\}$-dimensional space. From the problem setting we know that

$$T_{ij} = t_j + \tfrac{1}{c}\|\mathbf{m}_i - \mathbf{s}_j\| \,, \tag{4.5}$$

where T_{ij} is the time of reception and t_j is the time of emission of a signal.

This equation describes a cone in p+1-dimensional space where the signal time t_j is added as an extra variable, see Fig. 4.2. The vector $(\mathbf{m}_i; T_{ij})$ is the apex of the cone. The vector $(\mathbf{s}_j; t_j)$ describes a signal that occurred at position $\mathbf{s}_j$ at time point t_j. The expression $(\bullet\,;\circ)$ denotes a column vector of the form $(\bullet^T, \circ^T)^T$ where $(\bullet)$ is a column vector and $(\circ)$ is a scalar.

If for all receivers $\mathbf{m}_1, \ldots, \mathbf{m}_n$ and signal sources $\mathbf{s}_1, \ldots, \mathbf{s}_m$ these equations are satisfied we receive a possible solution of the given problem. Of course, this does not necessarily imply we found the correct solution as the problem might be underdetermined. Recall that there is no absolute solution since we obtain only a relative localization.

4.2.1 Error Function

Starting from an initial setting for all positions and time points our iterative approach greedily decreases an error function. We describe now this error function which corresponds to the potential energy of springs. We define the function in p+1-dimensional space as an expression of the violation of constraint (4.5) by

$$\Phi((\mathbf{d}; d)) = c\,d + \|\mathbf{d}\| \,, \tag{4.6}$$

where $\mathbf{d}$ is a difference vector in space and d is the time of flight. $\Phi((\mathbf{s}_j; t_j) - (\mathbf{m}_i; T_{ij}))$ describes the relation of a signal $\mathbf{s}_j$ emitted at time t_j relative to a receiver $\mathbf{m}_i$ which obtains the signal at time T_{ij}. See Fig. 4.2, illustrating the Φ-function.

If Φ yields a non-zero value, which we call an *invalid* cone constraint, one can change both the position and time $(\mathbf{s}_j; t_j)$ of the signal source and the position vector $\mathbf{m}_i$ of

the receiver by moving it in space in order to recover a valid position. According to Section 2.1, the receiver time T_{ij} is fixed, so $(\mathbf{m}_i; T_{ij})$ is restricted to movements in p-dimensional space, whereas $(\mathbf{s}_j; t_j)$ can be moved in p+1 dimensions.

We define the direction vector pointing towards the surface of the cone

$$\hat{\mathbf{n}}_{ij} = \left(\frac{\mathbf{s}_j - \mathbf{m}_i}{\|\mathbf{s}_j - \mathbf{m}_i\|}; \frac{1}{c} \right) , \tag{4.7}$$

which we normalize to $\mathbf{n}_{ij} = \frac{\hat{\mathbf{n}}_{ij}}{\|\hat{\mathbf{n}}_{ij}\|}$. The normalized direction vector $\mathbf{n}_{ij}$ describes the shortest path from $\mathbf{s}_j$ to the cone surface of $\mathbf{m}_i$, cf. Fig. 4.2. Note that the time axis is compressed by the signal velocity c. Scaling by c reveals that $\mathbf{n}_{ij}$ is in fact orthogonal to the cone surface.

For the case that $t_j > T_{ij} + \frac{1}{c}\|\mathbf{m}_i - \mathbf{s}_j\|$ and thus $\mathbf{n}_{ij}$ does not intersect the cone, we choose $\mathbf{n}_{ij} = (\vec{0}; -1)$ pointing along the time axis ensuring an intersection.

By construction there is a scalar $\kappa_{ij} \in \mathbb{R}$ such that $\Phi((\mathbf{s}_j; t_j) - (\mathbf{m}_i; T_{ij}) + \kappa_{ij}\, \mathbf{n}_{ij}) = 0$, i.e. an intersection point of $\mathbf{n}_{ij}$ and the cone surface exists. κ_{ij} equals the distance along $\mathbf{n}_{ij}$ between $(\mathbf{s}_j; t_j)$ and the cone surface (Fig. 4.2).

Now, for every receiver i and signal j the distance can be computed by the intersect theorem

$$\kappa_{ij} \;=\; \frac{\Phi((\mathbf{s}_j; t_j) - (\mathbf{m}_i; T_{ij}))}{\Phi((\mathbf{s}_j; t_j) - (\mathbf{m}_i; T_{ij})) - \Phi((\mathbf{s}_j; t_j) - (\mathbf{m}_i; T_{ij}) + \mathbf{n}_{ij})} \,. \tag{4.8}$$

For the goal to minimize the distances κ_{ij} between all signal senders and the cone surfaces of the receivers we require to manipulate their positions in $\mathbb{R}^{p+1}$. We aim to minimize the quadratic objective

$$E_{\text{sum}} \;=\; \min_{\mathbf{m}_i, \mathbf{s}_j, t_j} \sum_{i=1}^{n} \sum_{j=1}^{m} (\kappa_{ij})^2 \,, \tag{4.9}$$

which is proportional to the sum of the potential energy of springs [78]. By minimizing the potential energy of a physical mass-spring system representing the signal senders and receivers we minimize the objective E_{sum}. In case of a value $E_{\text{sum}} = 0$ we have found a scenario of senders and receivers, which is a possible explanation of the observed TDoA measurements.

4.2.2 Mass-Spring Simulation

We optimize the error objective E_{sum} by a simulation of a physical mass-spring system [78], calculating the signal source and receiver positions. The mass-spring system is based on *particles*, which are tuples $(\mathbf{x}_\ell^{(k)}, \mathbf{v}_\ell^{(k)}, m_0)$, representing the receivers and signals in p+1-dimensional space at discrete simulation times $k \in \mathbb{R}$, where $\ell \in \{i, j\}$ is a placeholder. The particles represent the physical properties position $\mathbf{x}_\ell$, velocity $\mathbf{v}_\ell$ and mass m_0. They obey Newton's law of inertia, i.e. velocity changes result only from the influence of forces. For every receiver vector $(\mathbf{m}_i; T_{ij})$ and sender vector $(\mathbf{s}_j; t_j)$ we create a particle $\left((\mathbf{m}_i; T_{ij})^{(k)}, \mathbf{v}_i^{(k)}, m_0\right)$, respectively $\left((\mathbf{s}_j; t_j)^{(k)}, \mathbf{v}_j^{(k)}, m_0\right)$.

For every sender particle we use the distance to the cone surface κ_{ij} of a receiver particle to implement a force based on the spring equation $\mathbf{f}_{ij} = \eta\,\kappa_{ij}\,\mathbf{n}_{ij}$, where η is a constant of the spring stiffness. We apply the force $-\mathbf{f}_{ij}$ to every receiver particle and the opposite force $\mathbf{f}_{ij}$ to the corresponding signal particle, accelerating the sender particles towards the cone surface and the receiver particles in the opposite direction, yet only in space $\mathbb{R}^p$, not in the time dimension.

In cases, when movements are locked in a dimension, the respective component of the force vector is set to zero, thus preventing changes of the position in the respective dimension, as in our case for the time component of the receivers. In more advanced cases, when restrictions of positions exist, for instance all receivers reside on a circle, or signals are fixed to a plane, penalty forces or local constraints [79] can be used.

In addition, we introduce quadratic damping, which is comparable to aerodynamic drag, stabilizing the simulation [78]:

$$\mathbf{w}_\ell^{(k)} \;=\; -\lambda\,\|\mathbf{v}_\ell^{(k)}\|^2 \frac{\mathbf{v}_\ell^{(k)}}{\|\mathbf{v}_\ell^{(k)}\|} \;=\; -\lambda\,\|\mathbf{v}_\ell^{(k)}\|\;\mathbf{v}_\ell^{(k)}\;.$$

$\lambda > 0$ is a damping factor which we choose small to achieve high velocity of spring relaxation, but large enough to avoid the simulation to oscillate and become instable. For proper choice of a damping constant, damping reduces the velocity of a particle, and in this way its kinetic energy. The overall force for every particle is the sum of all spring forces Θ_ℓ and the individual damping force, in total $\hat{\mathbf{f}}_\ell = \Theta_\ell + \mathbf{w}_\ell$.

The temporal integration of the mass-spring simulation is realized by a simple Euler-Cromer scheme [80] with a timestep of h:

$$\mathbf{x}_\ell^{(k+h)} = \mathbf{x}_\ell^{(k)} + h\mathbf{v}_\ell^{(k+h)} \qquad\qquad \mathbf{v}_\ell^{(k+h)} = \mathbf{v}_\ell^{(k)} + \frac{h}{m_0}\hat{\mathbf{f}}_\ell^{(k)}$$

An Euler-Cromer scheme is *stable*, even in the undamped case, i.e. errors do not amplify over time. When using springs the scheme is *conditionally stable*, which means the stability is dependent on the parameters [80]. In practice the choice of spring and damping parameters is not an issue, as the range of stable operation is large.

We initialize the particles close to the origin in p+1-dimensional space, randomized by a small amount to avoid singularities. The initial signal source time is set to the minimum of all associated receiver timestamps. This is the closest guess we can do, as no positions are known a priori.

After the start, forces are calculated. Position and velocity updates are made accordingly to the Euler-Cromer scheme. The simulation runs until a termination condition has been met, where we obtain the receiver positions $\mathbf{m}_i$ and the sender tuples $(\mathbf{s}_j; t_j)$ from the position vectors of the respective particles.

For the termination, either the overall energy function E_{sum} falls below a fixed threshold, or the overall particle velocity falls below a fixed limit, or a certain number of steps have been exceeded. If no TDoA error was presumed the latter two cases are

an indication that the algorithm did not arrive at the zero of the error function. We call this a *local minimum*, an issue which is discussed later in Section 4.3.1.

The spring-mass simulation relies only on basic numerical operations, such as vector addition and scalar multiplication. Same holds for the Euler-Comer integration scheme and the calculation of the distance scalar κ and the normal vector $\mathbf{n}$. Therefore, Cone Alignment requires only low numeric precision, which can be a benefit for implementation on embedded systems that do not provide a hardware floating point unit. This is an advantage over algorithms that use sophisticated operations such as inversion of the gradient matrix, which requires high numerical precision even if numerically robust matrix decomposition algorithms are employed.

4.3 Numerical Evaluation

We have implemented the Cone Alignment algorithm in C++. Simulations were run in both, the two-dimensional and the three-dimensional case, also cf. [5]. For the signal velocity we choose the speed of sound at 20 °C, which is $c = 343\,\mathrm{m/s}$.

For any number of receivers and signal sources $n, m \leq 14$ we created 100 random scenarios. Receivers and signal sources were placed in a two-dimensional, resp. three-dimensional space of 1000 meters edge length. For given randomly distributed signals in space we calculated the timestamps at every receiver. Then, the timestamp information was given to our algorithm and finally we evaluated the quality of the result by applying SVD alignment as described in Section 2.6 and comparing the output of the simulation to the ground truth positions. As an abort condition of the algorithm we chose an error threshold ϵ. In the successful case the remaining RMS error lay clearly below the threshold. If after 20.000 iteration steps the threshold could not be reached, the run was marked as not successful.

The runtime of this algorithm is $\mathcal{O}(mn)$ and it converges after 1,000 to 10,000 iterations for $n, m \leq 14$ which gives an absolute runtime of 0.01 to 0.70 seconds on a standard 2 GHz PC, cf. Table 4.1. Most interestingly, the number of iterations decreases with increasing numbers of signal sources and receivers due to over-determination, i.e. a highly negative degree of freedom, as described in Section 2.4.

For an evaluation of the necessary number of receivers and signals we calculated the remaining error after mapping the experimental positions to the corresponding ground truth positions using SVD alignment. The experiments indicate, that for the two-dimensional case at least four receivers are necessary and sufficient. If the number of signals is fixed to three, deploying a sufficient number of receivers makes the localization error decrease to zero. We found that employing at least four receivers in the 2D case, distance errors between estimated and true positions began to diminish, provided enough signal sources were given. In an analogous manner given three signal sources, which are obtained by enough receivers, RMS distances decrease to zero.

receivers	signals	iterations	duration (s)
2	2	1,815	0.013
14	2	4,359	0.132
2	14	5,086	0.148
5	5	9,087	0.108
7	7	5,620	0.170
10	10	4,636	0.333
14	14	4,259	0.680

Table 4.1: Average number of iterations and average experiment duration on a 2 GHz PC until the abort condition is satisfied in a 2D experiment. 100 runs were executed for every setting of receiver and signal numbers. The number of iterations decreases in the over-determined cases. [5]

In the three-dimensional case we observed similar results, with convergence for five receivers, respectively four signal sources. These observations correspond to our considerations from Section 2.4, where we predicted the reconstructability of all unknown positions for such numbers of signals and receivers.

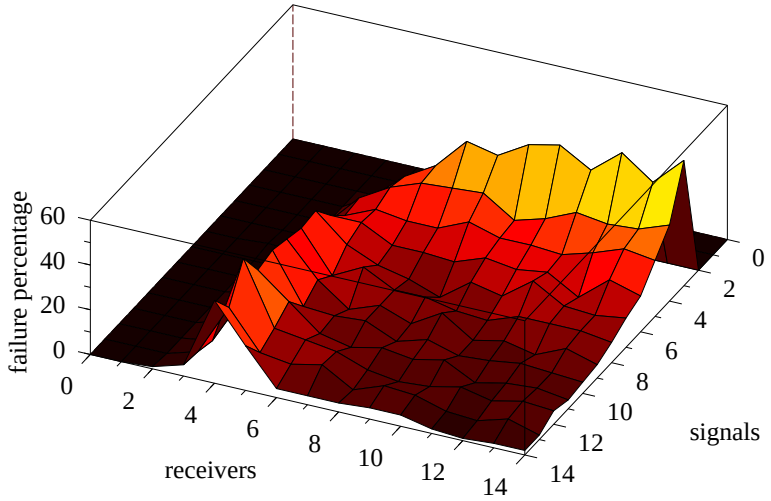
4.3.1 Local Minima

In some cases the localization algorithm failed and got stuck in a local minimum of the error function which is not the global minimum. This opposes reconstruction errors due to under-determined scenarios, where constraints contain too little information and degrees of freedom remain. Local minima occurred mainly in uniquely determined or over-determined scenarios. However, we saw cases where local minima also occurred in under-determined cases.

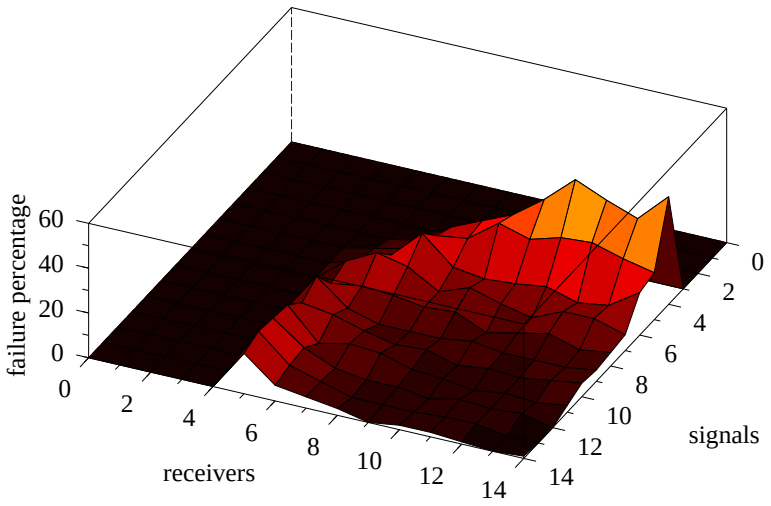
The failure rate converges to zero with increasing number of signals, depicted in Fig. 4.3(a) for the two-dimensional case and in Fig. 4.3(b) for three dimensions. Comparing this observation with Table 2.2 and 2.3 shows that high failure rates correspond to small absolute degrees of freedom.

In a visual representation we saw that items were blocked on the wrong side of a line or a plane. We implemented an algorithm that mirrored them on the other side by way of trial. This successfully resolved local minima in some cases, but not in all. Obviously, some of the local minima are complicated and hardly comprehensible by geometrical considerations. For now, we disable the algorithm and instead use repetitions of Cone Alignment with different random initial positions. By this means, we could improve the rate of success notably, which is described in Section 4.3.4.

Furthermore, we ran experiments with simulated TDoA error. Here, the error in timestamping the signals at the receivers is assumed to be Gaussian distributed. Timing errors may be induced from synchronization errors and from imprecisions in determining the time points of signal events in the recorded audio. Errors of a standard deviation up to 200 ms were tested, which is a spatial equivalent of 70 m.



(a) In two dimensions for four receivers and for three signal sources the risk of ending in a local minimum is exceedingly high.



(b) In three dimensions the risk of local minima is highest for five receivers and for four signal sources.

Figure 4.3: Distribution of local minima for two and for three dimensions for the Cone Alignment algorithm. The risk of ending in a local minimum culminates at the minimum cases and converges to zero in overdetermined scenarios. [5]

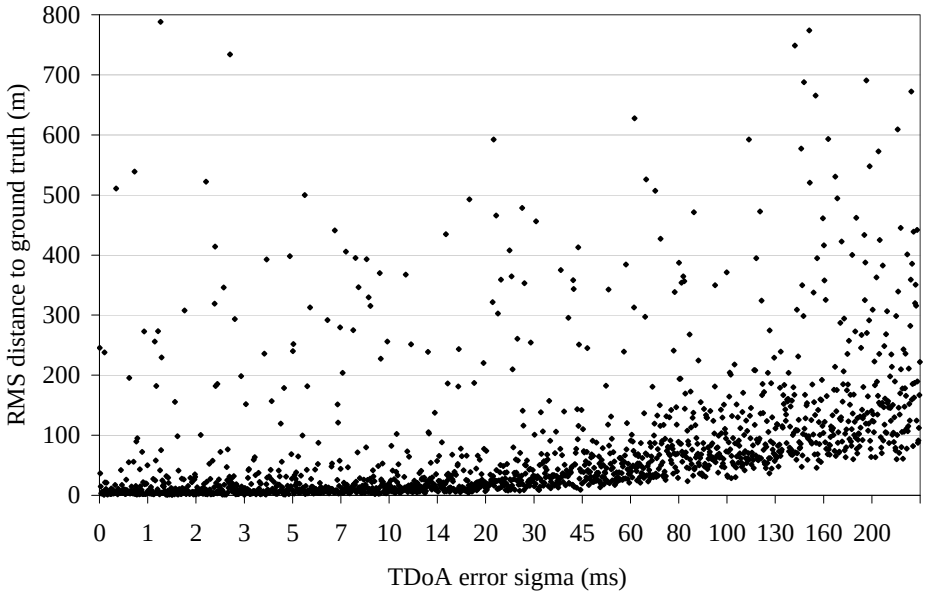


Figure 4.4: TDoA error experiment of the Cone Alignment algorithm for seven receivers and seven signal sources. For TDoA error steps from 0 to 200 ms a total of 1700 experiments were run. With increasing error the average distance from the true positions increases and local minima are harder to distinguish from the correct solution. [5]

With increasing TDoA error both the average distance from the true positions and the tendency of local minima increased. We observed this tendency difficult to quantize as with increasing error a local minimum is hard to distinguish from the global minimum in the least-squares sense. See Fig. 4.4 for the example of seven receivers.

4.3.2 The Scilab `lsqrsolve` Function

We have run numerical tests on the Scilab `lsqrsolve` function[2], which is an implementation of the Levenberg-Marquardt algorithm [81]. For a function $F : \mathbb{R}^n \to \mathbb{R}^m$ a call to $(\hat{\mathbf{x}}, F(\hat{\mathbf{x}})) \leftarrow \texttt{lsqrsolve}(\mathbf{x}_0, F, q, [J])$ calculates a local minimization of the objective $\|F(\mathbf{x})\|^2$, returning a vector $\hat{\mathbf{x}}$ and the residuum $F(\hat{\mathbf{x}})$. Optionally, a function handle for analytical calculation of the Jacobian $J(\mathbf{x})$ can be provided, otherwise the Finite Differences method [82] is used. In Finite Differences is

$$\frac{\partial F_k(\mathbf{x})}{\partial x_i} = \lim_{h \to 0} \frac{F_k(\mathbf{x} + \mathbf{h}_{[i]}) - F_k(\mathbf{x})}{h}, \qquad \mathbf{h}_{[i]} = (0, \ldots, 0, \underbrace{h}_{i\text{-th}}, 0, \ldots, 0)$$

an approximation of the gradient, where F_k is the k-th function of an enumeration of q functions, and $1 \leq i \leq n$.

[2] Scilab 4.1.2, `http://www.scilab.org`

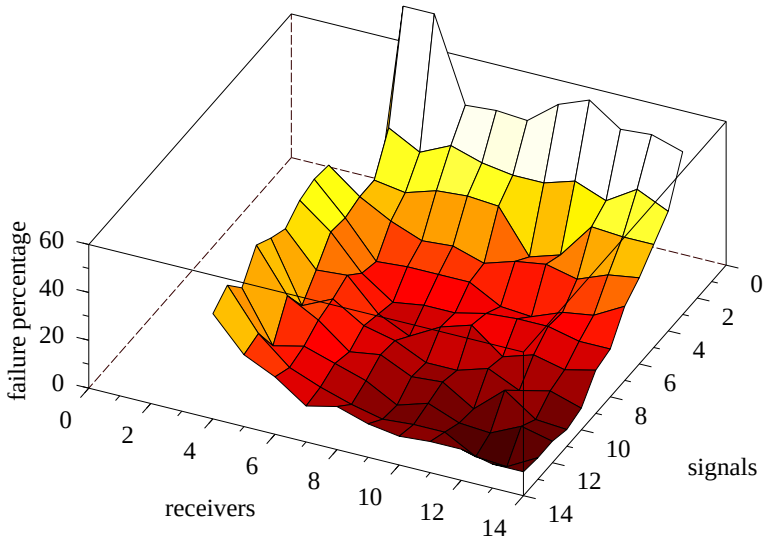


Figure 4.5: Distribution of local minima for two-dimensional settings for the Scilab Levenberg-Marquardt implementation with the Jacobian given. The probability of local minima is higher than for gradient descent and Cone Alignment, especially in the critical cases of four receivers and three signals.

For optimization of TDoA we consider the function f_{ij} and its derivative according to the definitions in Eqns. (4.1) and (4.2). In a simulation we ran 100 randomized cycles for every combination of $m, n \leq 14$ receivers and signal sources in the planar unit square, with and without the analytical derivative provided. As no TDoA measurement error was given we expected the function to return a zero residuum. If after a run the residual is $\|F(\mathbf{x})\| > 10^{-3}$ we marked a cycle as *failure*. In the evaluation we analyzed the proportion of random test cases that can be solved successfully, therefore with zero residuum. We observed an issue in under-determined scenarios, where none or all of the scenarios failed for certain combinations of receivers and signals. We eliminated the under-determined cases and consider only the results for determined and over-determined scenarios. See Fig. 4.5 for an evaluation of the failure cases of the Scilab Levenberg-Marquardt implementation with the analytical Jacobian given.

We compared the results of the Scilab minimizer `lsqrsolve` to test runs of the implementation of gradient descent from Section 4.1, and to Cone Alignment, where we ran similar test settings. For Cone Alignment a test field of 1000 m $\times$ 1000 m was used, instead of the unit square, and an condition of $E_{\text{sum}} > 1$ for the definition of a failed test, see Eq. (4.9), which corresponds to a relative error of 10^{-3}. Furthermore, Cone Alignment uses not a hyperbolic error function, but an estimation of the send time. However, we believe that the experiments are comparable, as the field size is only a scaling factor, and the residual after termination is mostly clearly above or below the decision threshold for local minima for all algorithms.

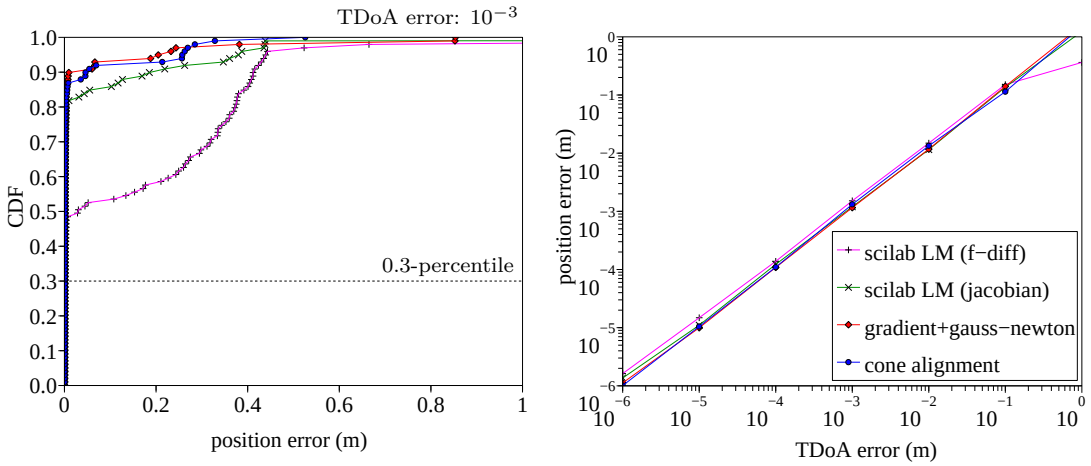


Figure 4.6: Evaluation of the algorithms with varying relative TDoA error in a planar setting of seven receivers and seven signals. *Left*: The position errors increase notably in case of local minima. *Right*: The plot of the 0.3-percentile shows that the position errors are proportional to the TDoA error. The algorithms behave very similar, as they minimize a similar quadratic error function.

In an evaluation of the precision under influence of measurement errors series of 100 test cycles with relative Gaussian error from 10^{-6} to 10^{0} were run in a planar setting of seven receivers and seven signals. In the cumulative distribution we see that position errors, compared to the true positions, are proportional to the input error in the majority of attempts. However, in some runs, depending on the algorithm up to 50 %, the error of position estimation is salient, see Fig. 4.6, *left*. These disproportionalities appear to be a result of local minima. We consider the 0.3-percentile of position errors (the 0.5-percentile is the median), excluding potential local minima. Then, the four algorithms exhibit only small variations, see Fig. 4.6, *right*, which is not surprising, as they minimize the quadratic error of the same conic or hyperbolic TDoA equations. The LM algorithm without Jacobian exhibits higher error, which might be caused by numerical issues of the finite differences method.

In the evaluation of failure cases with no runtime error, we immediately notice that test cycles where no analytical definition of the derivative is given, exhibit a high amount of failures, and no decrease of failures due to local minima for over-determined cases, see Fig. 4.7. We attribute this to the difficulty of choosing a proper value of h in finite differences, leading to a badly chosen secant if h is too large, and numerical rounding errors if h is too small. Small errors in the estimated gradient may have a significant effect on the direction of convergence in the Gauss-Newton component of Levenberg-Marquardt, and the problem is numerically critical due to the high dimensionality of the state vector $\mathbf{x}$. If the Jacobian is provided, the behavior of the algorithms is more effective for over-determined scenarios, see Fig. 4.7. A difference between the Levenberg-Marquardt, the Gauss-Newton and the Cone Alignment is only barely noticeable, probably with small advantages for the gradient method and Cone

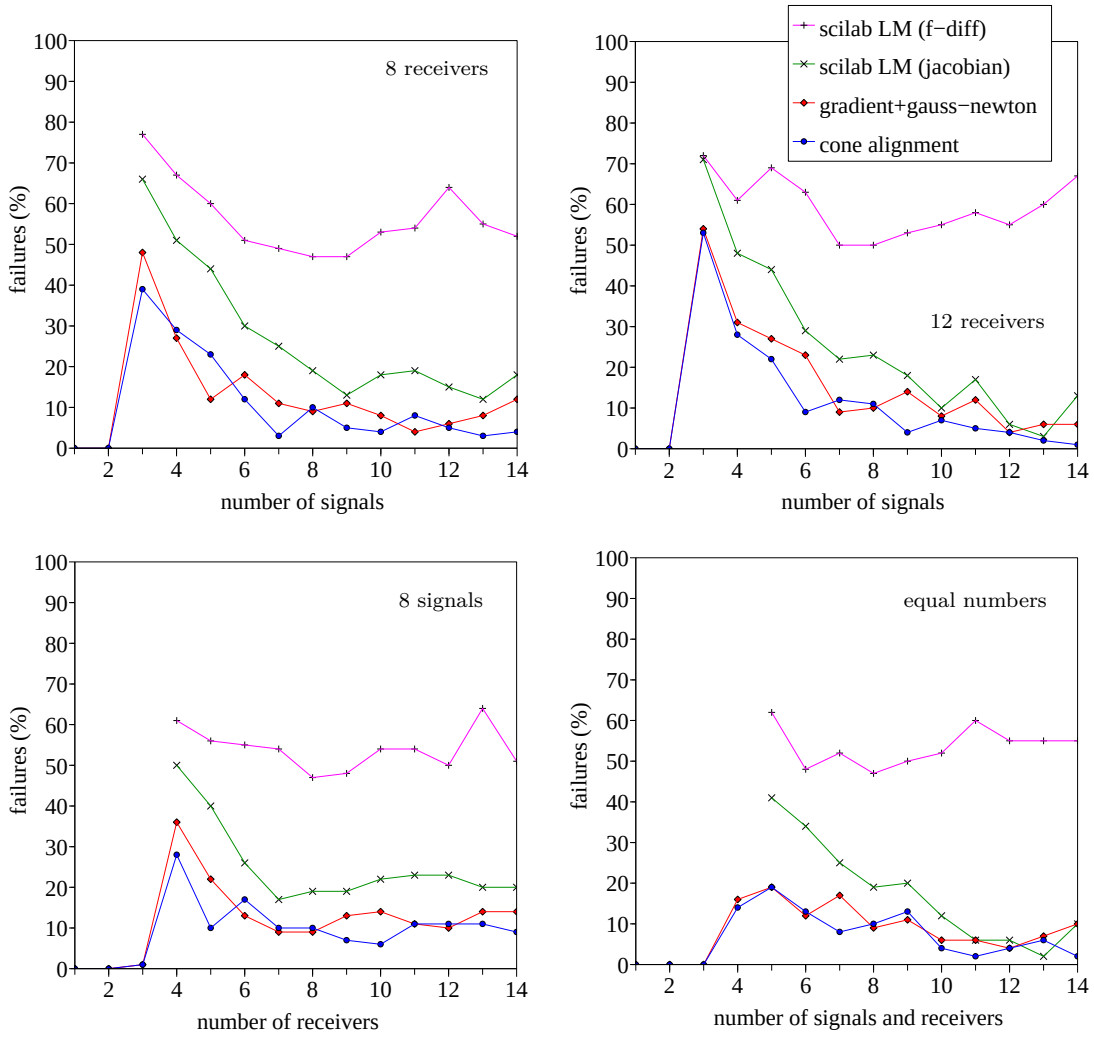


Figure 4.7: Failure cases due to local minima for 100 runs in the planar world for the algorithms: Levenberg-Marquardt (LM) with finite differences, LM with analytical Jacobian (green line, see also Fig. 4.5), combined gradient descent and Gauss-Newton, and Cone Alignment, for varying numbers of signals and receivers.

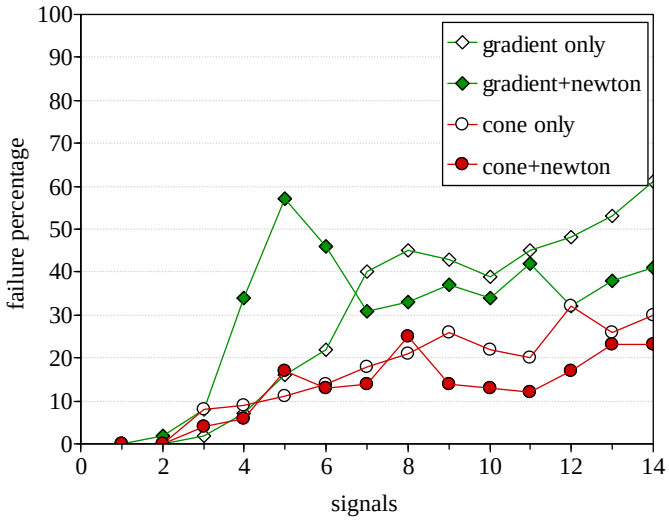


Figure 4.8: Comparison of failure ratio for 100 runs per combination. In the important case of four receivers in the plane the Cone Alignment (red marks) can find the global minimum more frequently than the gradient method as used by Biswas & Thrun [41] (green marks). Using the Gauss-Newton method after both Cone alignment and gradient descent improves the probability to satisfy the abort condition. [5]

Alignment. Levenberg-Marquardt appears to perform weaker in the critical cases when only four receivers or only three signals are given.

In the next section, we consider further comparison of the gradient method and Cone Alignment. Both algorithms were run with and without the Gauss-Newton algorithm used afterwards. Again, we observe regions with higher failure rate for the gradient method, especially in the case of four receivers and in the case of three signal sources.

4.3.3 Minimum Case: Four Receivers

We focus on the case of four receivers in the plane, the smallest case in which positions can be calculated, which is especially interesting for application, as the receivers are usually the costly part of TDoA localization, while signals may be complimentary in a natural environment.

According to Fig. 4.8, Cone Alignment exhibits a lower failure rate for a varying numbers of signals, therefore a lower tendency to get stuck in local minima. We observe the same with a fixed number of three signal sources. As an explanation we suppose that the gradient descent method fails to escape local minima, as it can only decrease in its error function. In contrast, the particles of the Cone Alignment gather momentum while relaxing the spring constraints. In this way, barriers can be overcome towards a smaller minimum. As we implemented particle velocity as an imitation of physical springs we did not have to optimize a momentum parameter.

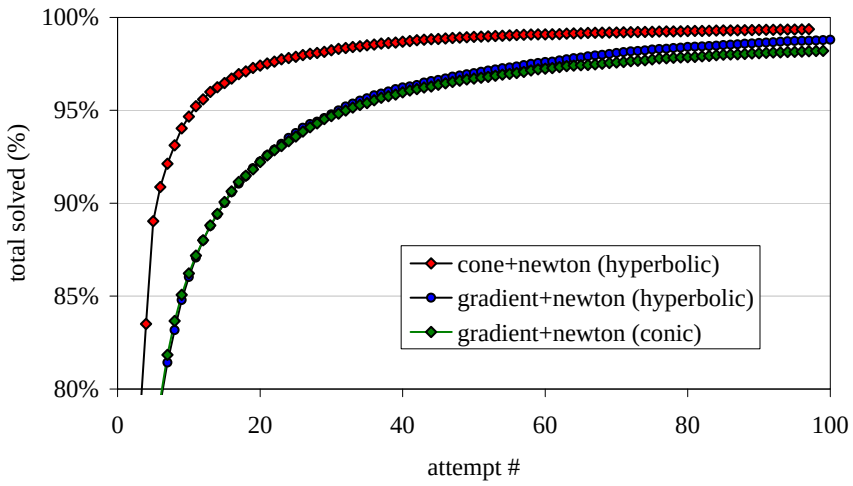


Figure 4.9: Repeated executions with randomized initialization increase the probability to find the global minimum. In the important minimum case of four receivers and six signal sources in the plane we repeated a scenario 100 times. Of 20.000 random scenarios Cone Alignment solved 99.4%. Gradient descent with Gauss-Newton combined is slightly weaker, solving 98.8 % with hyperbolic equations and 98.2 % with conic equations.

Furthermore, we observe an exceedingly high tendency to get stuck if the Gauss-Newton method is used after the gradient descent for four to six signal sources. We could not finally elaborate the reason for that.

Both algorithms, the Cone Alignment and the gradient method, benefit from the combination with the Gauss-Newton algorithm (Fig. 4.8). We observed that scenarios with very shallow gradients were marked as "unsolved" when an error threshold could not be reached after a maximum number of iteration steps. In several cases the threshold could be met when the Gauss-Newton algorithm was executed afterwards. This happens especially in ill-conditioned scenarios, for instance when two of the four receivers are close. In general, the number of iteration steps is immensely reduced for both algorithms when the run is finalized with a subsequent execution of the Gauss-Newton algorithm. With increasing number of both receivers and signals the ratio of local minima decreases. Also, the disparity between both algorithms diminishes.

4.3.4 Towards a 100 Percent Solution

We have analyzed our algorithms to increase the success rate of finding the global minimum by repeated executions. The repeated attempts come with increased computational power for finding the solution, but such calculations might be trivially executed in parallel.

We consider now only gradient descent and Cone Alignment with the Gauss-Newton method executed after the abort condition, increasing the rate of success in ill-condi-

tioned scenarios. However, now we distinguish between gradient descent with hyperbolic equations as introduced in Section 4.1.1 and conic equations, as proposed in [41], where also the send time is estimated. For the three algorithms we ran approximately 20,000 random scenarios. Every scenario was repeated with random initialization until it is solved, up to a maximum of 100 times, where we noted down the required number of repetitions.

In the minimum case of four receivers and six signal sources in the plane, Cone Alignment has already solved 95.6 % of the scenarios after 12 repeats, while both gradient descent with hyperbolic and with conic equations achieve only 88.0 %, see Fig. 4.9. After 100 repeats the algorithm achieves a success rate of 99.4 % with randomized initialization. Only 0.6 % of all scenarios remain stuck and unsolvable. In the case of the gradient descent method we could not achieve such a high success rate. Still 1.2 % of all scenarios fail to be solved when using gradient descent with hyperbolic equations, which is more by a factor of two. For gradient descent with conic equations is noticeable after 100 repeats that even 1,8 % of all random scenarios remain unsolved, which we attribute to the additional complexity of conic equations, where the send time is estimated as an additional parameter.

As we can split larger scenarios into subsets of this size and merge them after solving a subset, we can solve larger scenarios in the same way. This form of repeating should work also for the other minimum cases, for $5\,/\,4$ and for $7\,/\,3$ receivers and signal sources, and for the three-dimensional case.

4.4 Related Work

We have derived the Jacobian and developed gradient based approaches to demonstrate and analyze the application of non-linear iterative first-order optimization methods for problems of calibration-free TDoA. An introduction to non-linear optimization is given in [81, 82, 83, 84]. Optimization with a focus on practices for machine learning is discussed in [85].

A standard approach for local optimization of an unconstrained non-linear function $f(\mathbf{x})$ is the gradient descent method [82], where a function is minimized by following the direction of the gradient. The gradient, i.e. the linear term of a Taylor expansion of $f(\mathbf{x})$, is either calculated by evaluation of the Jacobian matrix $J(f) = \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}$ or numerically by the Finite Differences method [82]. *Conjugate* gradient descent [86] requires a system of linear equations, therefore does not apply to TDoA multilateration which is non-linear. Yet there exist extensions for non-linear optimization [87], such as Fletcher and Reeves [88], and Hager and Zhang [89].

The multivariate Newton method [90] that calculates a zero of $f(\mathbf{x})$ requires the Hessian matrix $H(f) = \frac{\partial^2 f(\mathbf{x})}{\partial \mathbf{x}^2}$ in every iteration, which may be difficult to calculate, depending on the function. We have implemented the Gauss-Newton algorithm [91] instead, a modification that uses the Jacobian matrix to minimize an objective $\|f(\mathbf{x})\|^2$.

The Levenberg-Marquardt algorithm [75, 81] is based on Gauss-Newton, improving convergence by introducing a "trust-region method". The BFGS method [81] is a Quasi-Newton method that uses an approximation of the Hessian.

4.5 Global Optimization

The previously discussed gradient based and spring-mass based algorithms are both approaches to *local* optimization. They are intrinsically prone to not finding the global optimum, which is fundamental to local optimization. For the definition of *completeness* we follow the definition in [92], of which we give an overview in the following.

Algorithms that only perform a local search for the first optimum they encounter do not find the global optimum with probability one. They are considered *incomplete* optimization algorithms.

By infinite number of repetitions with randomized initialization over the entire search space one may suppose that incomplete algorithms eventually initialize with values that end in the global optimum solution. Such algorithms are *asymptotically complete*, as the global optimum is found with probability one. However, in presence of measurement errors the algorithm does not know from the residuum of the objective if the global optimum was approached, or if a better local optimum exists somewhere, as the residuum is always larger than zero. Therefore, asymptotically complete algorithms cannot find the global optimum in definite time.

Algorithms that find the global optimum after finite time with certainty are considered *complete*. They know by exploration of the search space, when the algorithm has terminated that the global optimum was found, and not one of several inferior local optima.

In [92] a survey is given about global optimization. Global optimization approaches such as BARON and OQNLP are analyzed and compared in [93] and [94]. Recent advances are discussed in [95]. Note that global optimization algorithms do not guarantee a solution for *every* random numeric problem in practice. First, finding the global optimum in a multivariate problems of large dimension may take a very long time of minutes or hours. Second, it may be impossible for an algorithm to further proceed towards a solution due to precision limits of floating point operations.

4.6 Summary

In this chapter we have addressed the general problem of calibration-free TDoA localization by iterative local optimization approaches. These are the standard approaches to the problem, as they consider the minimum cases, and no assumptions of the receiver and signal positions are required.

We have derived the Jacobian for gradient descent and Gauss-Newton based algorithms and we have implemented the algorithms. We have furthermore proposed the

Iterative Cone Alignment algorithm, an iterative mass-spring simulation that minimizes the problem of anchor-free TDoA localization in an energy minimization manner. The simulation bases on particles obeying Newton's law of inertia that gather momentum while spring constraints are relaxed.

In extensive numeric experiments we have analyzed the failure cases of the algorithms. We demonstrated that our mass-spring algorithm can find the correct positions of receivers and signals in randomized scenarios with higher probability than the gradient descent approach. With Cone Alignment we could find the correct solution in 99.4 % of all cases of four receivers and six signals in planar space by repeated executions of the algorithm, making the algorithm asymptotically complete.

Yet, the problem of not finding always the correct solution remains unsolved by the discussed optimization algorithms. Obviously, an issue are not only local minima, but also difficulties of numerical nature. First, the iterative algorithms are bound to abort conditions of the error function. Second, the algorithms based on the first-order derivative encounter numerical instability when inverting the Jacobian – even with the most robust numerical methods, such as QR decomposition.

In case of TDoA measurement error remains a residuum of the error function. Then, these local optimization algorithms cannot decide if the residuum originates from a local minimum next to a smaller minimum, or is just the result of measurement error. Global optimization algorithms that explore the search space by non-local random stepping to find the global minimum, such as the Gelfand algorithm [76, 83], are rare, as the exploration of the search space is traded for the lack of an residual-dependent abort condition in definite time. Usually, such algorithms are aborted after a given period of time. Furthermore, in cases of ill-conditioned problems, i.e. when very small changes of the observed time difference result in large changes of the positions, a local minimum is hard to distinguish from the numerical inability to further approximate the correct solution, even if no TDoA measurement error is assumed.

In the next chapter we present a complete exploration algorithm for calibration-free TDoA based on iterative search in a tree of uncertainty test cases. This algorithm can provably converge towards the correct solution, and furthermore enumerate all solutions up to an error bound ϵ.

5 A Branch-and-Bound Algorithm

Local optimization can yield a solution to the TDoA problem in many cases. In a simple TDoA problem, as for example finding the position of a signal source using hyperbolic equations, one can calculate the correct solution often by just a few iterations of the Gauss-Newton algorithm. Unfortunately, such algorithms are not guaranteed to converge towards the correct solution, especially in the high-dimensional case of calibration-free TDoA, which has been discussed in literature [41, 54]. In the following we formulate a limit criterion to describe the ability of an algorithm to find a solution to the TDoA problem.

5.1 Existence of a Solution in TDoA

Consider parameters $\mathbf{x}$ from a set X, which are in our case the true positions of TDoA senders and receivers, and observations $\mathbf{z}$ from a set Z that depend on the parameters. A function $f : X \to Z$ is an injective mapping, iff a different choice of parameters always results in a change of observations. The injectivity of f depends not only on the definition of the function, but also on the choice of $\mathbf{x}$. For the problem of time differences of arrival we assume that the mapping of positions to time measurements is injective, if at least the minimum number of receivers and signals is given and no signals are lost during reception, if $\mathbf{x}$ is normalized in a way that congruencies are eliminated, and if $\mathbf{x}$ is limited to general positions, i.e. receivers or signals are not collinear or coplanar, therefore the resulting equation system is over-determined.

Note, that we do not show a proof of injectivity of the TDoA problem. However, based on our considerations about the solvability in Section 2.4, we believe that no identical vectors of time differences for two different choices of signals and receivers exist if the above conditions are satisfied.

From the injectivity of f follows the existence of an inverse mapping g, such that $g(f(\mathbf{x})) = \mathbf{x}$, if $\mathbf{x}$ is a strict subset of X, which is the case. Concerning the TDoA problem, this implies that a unique solution of positions for given time differences exists. For instance, a unique solution does exist for at least four receivers and six signals in the plane, if they reside in general position, i.e. they are not exactly on a line or even at identical positions. A unique solution does not exist for four receivers and five signals, because the mapping f of positions is not injective, cf. Fig.2.2.

Now, we define defective observations $\mathbf{z}_\epsilon = f(\mathbf{x}) + \mathcal{D}(\epsilon)$, where $\mathcal{D}$ is a distribution of errors bounded by $[-\epsilon, \epsilon]$. Such errors occur for instance from imprecisions in timing or from environmental disturbances.

Definition 3 *Under the assumption of existence of the inverse g, an algorithm $\hat{g}$ is called a solution to the TDoA localization problem if*

$$\lim_{\epsilon \to 0} \hat{g}(\mathbf{z}_\epsilon) = \mathbf{x} \; . \tag{5.1}$$

This definition of convergence is equivalent to

$$\forall \delta > 0 \, \exists \epsilon > 0 : \forall \mathbf{z}_\epsilon \left(\|\mathbf{z}_\epsilon - \mathbf{z}\| < \epsilon \; \Rightarrow \; \|\hat{g}(\mathbf{z}_\epsilon) - \mathbf{x}\| < \delta \right) \; . \tag{5.2}$$

So, $\hat{g}$ is convergent if holds, whenever measurement errors are bounded by a constant ϵ, the resulting position errors are bounded by another constant δ.

Optimization algorithms based on local search do not satisfy the convergence criterion, as the result may be a local minimum of the error function. Then, the true positions $\mathbf{x}$ are not obtained, even if the measurement error is zero. In the following we present a complete algorithm which bounds the error of positions of receivers, given a bounded input error ϵ, and therefore satisfies the criterion of convergence.

5.2 The Error Bound

We present an algorithm which is based on partitioning of a search space into subspaces, and a hypothesis test to decide if a subspace is part of the solution [96]. An inspiration to the algorithm was given in [97], where sender positions are approximated by recursive partitioning of three-dimensional space, assuming known positions of receivers. Such a subdivision and test strategy is commonly referred to as *branch-and-bound* algorithm. According to literature the application of such algorithms is is often in the domain of linear programming and discrete problems [98, 99, 100]. However, the principle of a branch-and-bound algorithm holds also for optimization in a high-dimensional continuous search space.

Consider the setting of n receivers and m signal senders in the unit square in two-dimensional space $\mathbb{R}^2$, according to Section 2.1. The signals are emitted at times t_j and propagate from the sender positions $\mathbf{s}_j$ to the receivers $\mathbf{m}_i$ in a direct line. For simplicity we normalize the speed of signals to $c = 1$.

Assume, that the reception times of signals are subject to imprecisions, which are bounded by ϵ. So the positions of receivers and signals are bounded by the constraint

$$T_{ij} - \epsilon \; \leq \; t_j + \|\mathbf{m}_i - \mathbf{s}_j\| \; \leq \; T_{ij} + \epsilon \; . \tag{5.3}$$

If we can find a set of receivers $\mathbf{m}_i$ and of signal positions $\mathbf{s}_j$ for all mn constraints, then we have found a possible solution up to the error ϵ. We call this an *ϵ-approximation* of the TDoA problem.

Note that in such an approximation the TDoA error of Inequations (5.3) is bounded by ϵ, but the positions of receivers or signals are not. First, depending on the problem, a small error can lead to large position errors. Second, if the problem is underdetermined, the results of the algorithm can be entirely wrong, even if no TDoA error

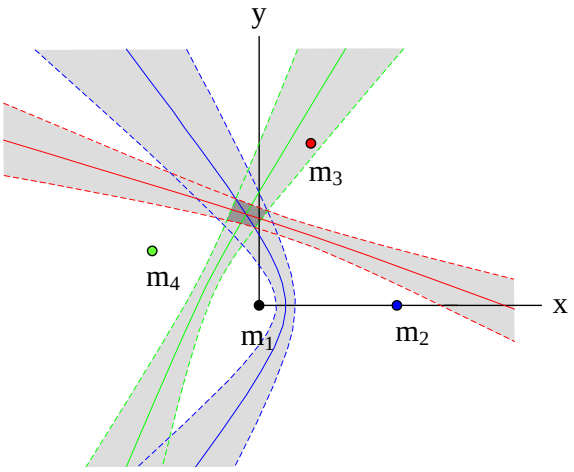


Figure 5.1: The measured time difference τ_{ij} between $\mathbf{m}_1$ and $\mathbf{m}_i$ yields a hyperbola (solid lines). Inequation (5.5) bounds a region of uncertainty where $\mathbf{s}_j$ can reside (gray regions), here depicted for the origin $\mathbf{m}_1$ and three other receivers. [96]

is assumed. However, if a problem is determined, and if the TDoA error is zero, then the positions of senders and receivers converge towards the correct solution, as required in Def. 3, which we demonstrate later on. In the following we describe a constraint condition to test the position of four or more receivers.

5.3 A Test of Feasibility

When the positions of at least four receivers are known, the positions of the senders may be calculated quickly. Then, we need to find signal positions to satisfy Inequations (5.3) for a test of feasibility of a setting. This inspires the following test algorithm [96].

First, we eliminate the sending time by combining two receiver times T_{ij} and $T_{\ell j}$ to $\tau_{i\ell j} = T_{ij} - T_{\ell j}$. Because of redundancy we choose $\ell = 1$ and $i > 1$ without loss of generality. We obtain the hyperbolic equation

$$\mathcal{H}(i, j, \tau_{ij}) \;=\; \|\mathbf{m}_i - \mathbf{s}_j\| - \|\mathbf{m}_1 - \mathbf{s}_j\| \;=\; \tau_{ij}\;. \tag{5.4}$$

The equation describes a hyperbola with $\mathbf{m}_1$ and $\mathbf{m}_i$ as focal points and $\mathbf{s}_j$ as a point on the curve. For the ϵ-approximation we consider the inequality

$$-\epsilon \;\leq\; \|\mathbf{m}_i - \mathbf{s}_j\| - \|\mathbf{m}_1 - \mathbf{s}_j\| - \tau_{ij} \;\leq\; +\epsilon\;, \tag{5.5}$$

which follows from the $\frac{1}{2}\epsilon$-approximation problem of Inequations (5.3). The inequality describes an area of possible positions for $\mathbf{s}_j$. This *uncertainty band* is bounded by the "inner" hyperbola $\mathcal{H}(i, j, \tau_{ij} - \epsilon)$ and the "outer" hyperbola $\mathcal{H}(i, j, \tau_{ij} + \epsilon)$. If we repeat this step for all microphones, we obtain the intersection where $\mathbf{s}_j$ can reside.

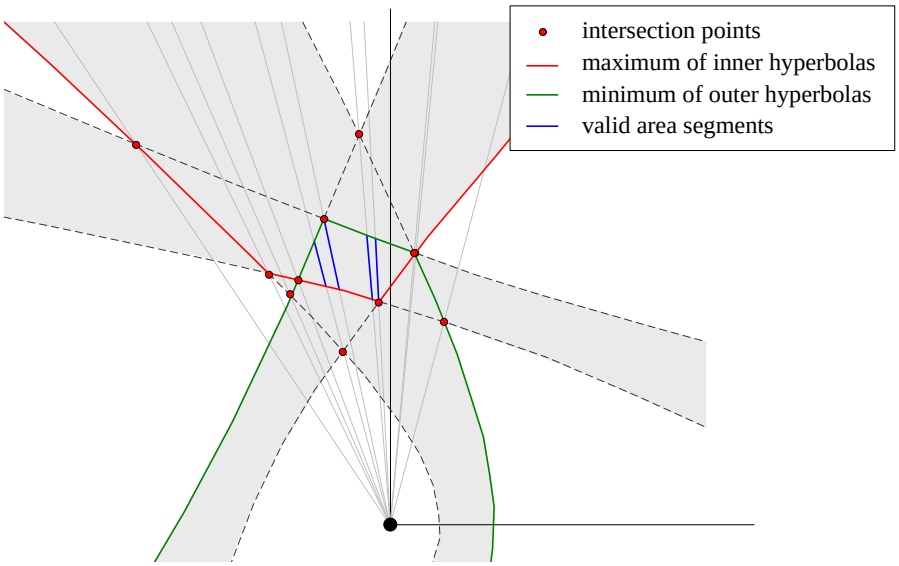


Figure 5.2: Scheme of the line-sweep algorithm. If the curve of maximum radius of the inner hyperbolas (red) and the curve of minimum radius of the outer hyperbolas (green) enclose a region, then the ϵ-test is positive for a scenario. The domain borders and hyperbola vertices are not considered in this schematic figure.

See Fig. 5.1 for an example of four receivers and one signal. If the intersection is non-empty for all signals j, then there is a solution to the approximation problem.

For the test of existence of such an area we use a segment test algorithm [96] in a polar coordinate system. In such a polar representation a hyperbola of the form (5.4) can be described as a function of the angle α,

$$\mathcal{R}(\alpha)_{\tau_i} = \frac{d_i^2 - \tau_i^2}{2\,|d_i \cos(\alpha) - \tau_i|}\,, \tag{5.6}$$

where $d_i = \|\mathbf{m}_i - \mathbf{m}_1\|$, and where the domain is $\alpha \in \left(-\arccos\left(\frac{\tau_i}{d_i}\right), \arccos\left(\frac{\tau_i}{d_i}\right)\right)$. Note that the borders are excluded from the domain.

As every hyperbola is only defined in a certain domain, we can compute an intersection of the domains, where all hyperbolas are defined. Only this domain intersection is considered for a possible intersection area of the uncertainty bands. The domain intersection may consist of multiple disjoint sets. Then, proceed for every set of domain intersections individually.

Proposition 1 *A non-empty intersection area of hyperbolas $\mathcal{H}(i, j, \tau_{ij})$ exists, iff* $\exists\, \alpha_1, \alpha_2 : \alpha_1 < \alpha_2, \forall\, \alpha \in [\alpha_1, \alpha_2] : \max_i \left(\mathcal{R}(\alpha)_{\tau_i - \epsilon}\right) < \min_i \left(\mathcal{R}(\alpha)_{\tau_i + \epsilon}\right)$.

Therefore, the uncertainty bands enclose an area iff an interval exists, where the minimum of the radius of the outer hyperbolas is larger than the maximum of the radius of the inner hyperbolas.

It is sufficient to evaluate the sweep line only at critical points, which are the intersection points of the hyperbolas, the boundaries of the domain intersection, and the *vertices*, the points of minimum radius. Between these critical points no intersection area can begin or end, due to the continuity and the monotony of the hyperbolas in the domain.

To execute the ϵ-test, we first calculate all intersection points of the hyperbolas. The $2(n-1)$ hyperbolas can be considered in $\binom{2(n-1)}{2}$ pairings, with a maximum of two intersections per pair, which is implied by the symmetric radius function, so the number of intersections ψ is bounded by $\psi \leq (2n-2)(2n-3)$. These intersections can be calculated in closed form in time $\mathcal{O}(n^2)$. Now, for every critical point we test if the sweep line of the angle encloses a segment according to Proposition 1, see Fig. 5.2, which can be done in linear time, yielding the total runtime $\mathcal{O}(n^3)$.

By using a sweep line algorithm, where the sweep line starts at the origin, rotating around it, the existence of an area can be tested in smaller asymptotic time. The sweep line algorithm is motivated by the observation that the line intersects such a hyperbola at most once, as of the radius function (5.6) of the hyperbola. With this algorithm we track the curve of maximum radius of the inner hyperbolas and the curve of minimum radius of the outer hyperbolas, therefore searching for the boundary curve of the intersection area of the uncertainty bands, see Fig. 5.2.

Here, the intersection points need to be sorted first, which is done in $\mathcal{O}(n^2 \log(n))$ for the quadratic number of intersection points. Then, beginning after the start of the domain, the minimum radius and the maximum radius is calculated once. If the curves intersect during the sweep at one of the now following intersection points, an intersection area begins or ends.

These tests can be calculated in constant time for every intersection point, by noting the index of the respective hyperbola and by considering the monotony of the hyperbolas, yielding the asymptotic time $\mathcal{O}(n^2)$ after sorting, so the total time is $\mathcal{O}(n^2 \log(n))$. Due to the design of the branch-and-bound algorithm, n is small, hardly larger than four, so a test with small constant overhead should be chosen.

Either test is repeated for all m signal sources, yielding the asymptotic time $\mathcal{O}(mn^3)$ for the segment test, and $\mathcal{O}(mn^2 \log(n))$ in case of the sweep line algorithm. If the test is satisfied for all signals then the setting of receiver and signal positions is an ϵ-approximation of the observed time differences of arrival.

5.4 Recursive Grid Refinement

Previously, we introduced a test to decide the feasibility of a setting of receivers for given TDoA data, with regard to an approximation error ϵ. Now we present an algorithm based on the ϵ-test to determine the positions of receivers [96].

For n receivers, where we eliminate three unknowns for the translation and rotation, the dimensionality of the search space is $2n-3$. Now, to find the positions of n receivers,

Algorithm 1 Breadth-first search for ϵ-environment for n receivers. [96]

Require: Initial guess of receiver positions $\mathcal{M} = \{\mathbf{m}_1, \ldots, \mathbf{m}_n\}$,
receiver times $\mathcal{T} = \{T_{ij}\}_{i\in[n],j\in[m]}$, target ϵ_{target}

queue $Q \leftarrow \emptyset$
$\mathcal{M} \leftarrow (0,0)^n$,
$Q \leftarrow Q \cup (0, \mathcal{M})$
repeat
  $(\hat{d}, \hat{\mathcal{M}}) \leftarrow Q.\text{pop}()$
  for all $b_1, \ldots, b_{2n-3} \in \{-1, 1\}^{2n-3}$ **do**
    $\tilde{\mathcal{M}} = \hat{\mathcal{M}} + \frac{1}{2^{\hat{d}+1}} \cdot ((0,0), (b_1, 0), (b_2, b_3), \ldots, (b_{2n-2}, b_{2n-3}))$
    if ReceiverPositionIsFeasible $\left(\tilde{\mathcal{M}}, \mathcal{T}, \frac{1}{2^{\hat{d}+\frac{1}{2}}}\right)$ **then**
      $Q \leftarrow Q \cup (\hat{d}+1, \tilde{\mathcal{M}})$
    end if
  end for
until $Q = \emptyset$ **or** $\frac{1}{2^{\hat{d}+\frac{1}{2}}} \leq \epsilon_{\text{target}}$
return $Q.\text{pop}()$

in a trivial approach we would enumerate all positions of receivers in a grid of cell size $\frac{\epsilon}{\sqrt{2}}$. This would result in the brute force exploration of $\left(\frac{\sqrt{2}}{\epsilon}\right)^{2n-3}$ grid cells where the ϵ-test is executed for each cell.

Instead we consider the construction of a recursive search tree, as shown in Algorithm 1. A node in this search tree corresponds to a setting with n receiver positions $(\mathbf{m}_1, \ldots, \mathbf{m}_n)$ in the unit square with the edge length $s = 1$. As described, $\mathbf{m}_1$ is normalized to the origin and $\mathbf{m}_2$ to the x-axis. We start with a root node, which we call depth level $d = 0$. Now, for every node we test for the feasibility of these receiver positions using the ϵ-test, as demonstrated in the previous section. The uncertainty ϵ depends on the depth d of the node. In case the setting is feasible, we add child nodes for the next level $d + 1$.

For searching for $2n - 3$ variables we obtain 2^{2n-3} child nodes, which represents all combinations to shift the variables by a vector $s \cdot (\pm 1, \ldots, \pm 1)_{[2n-3]}$. In the case of four receivers we obtain $2^5 = 32$ child nodes.

Lemma 5 *Displacement of the position of a receiver $\mathbf{m}_i$ in an area of edge length s increases the error according to Inequality 5.5 by at most $\sqrt{2}\,s$.*

Proof: Given is the position of a receiver $\mathbf{m}_i$ and an error ϵ caused by measuring the timestamps. According to Inequation (5.5) holds

$$\left| \|\mathbf{m}_i - \mathbf{s}_j\| - \|\mathbf{s}_j\| - \tau_{ij} \right| \;\leq\; \epsilon \,. \tag{5.7}$$

Now, displace the receiver by a vector $\mathbf{x}$, where $\|\mathbf{x}\| \leq \hat{\epsilon}$, which induces further error. By choosing $\mathbf{x}$ in a square of edge length $\hat{s}$ we obtain $\hat{\epsilon} = \sqrt{2}\hat{s}$ according to the circle

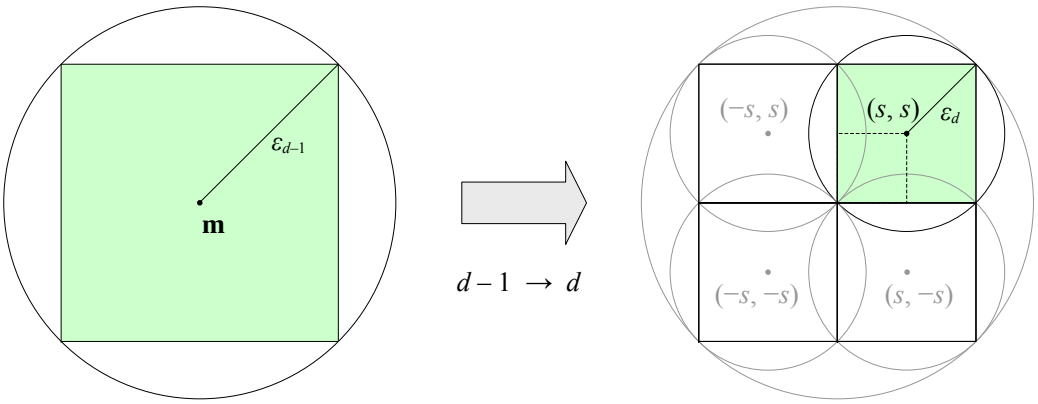


Figure 5.3: In the refinement step in depth d the edge length of a previous cell is partitioned into grid cells of edge length 2^{-d}. Every variable is shifted by $s = \pm 2^{-d-1}$, in this figure depicted for search space $\mathbb{R}^2$. The new cells are tested for $\epsilon = 2^{-d-\frac{1}{2}}$.

equation. Applying the triangle inequality yields

$$\left| \|\mathbf{m}_i + \mathbf{x} - \mathbf{s}_j\| - \|\mathbf{s}_j\| - \tau_{ij} \right| \;\leq\; \Big| \underbrace{\|\mathbf{x}\|}_{\leq \hat{\epsilon}} + \underbrace{\|\mathbf{m}_i - \mathbf{s}_j\| - \|\mathbf{s}_j\| - \tau_{ij}}_{\leq \epsilon} \Big| \;\leq\; \epsilon + \hat{\epsilon}$$

Therefore, the error induced by displacement of the receiver $\mathbf{m}_i$ increases by at most $\hat{\epsilon} = \sqrt{2}\, s$, which ends the proof. □

In the refinement step of the branch-and-bound algorithm a cell in the search space is split and the previous estimations $\hat{\mathbf{m}}_i^{(d-1)}$ of depth $d-1$ are shifted by an amount $s = 2^{-d-1}$, where we obtain $\hat{\mathbf{m}}_i^{(d)} = \hat{\mathbf{m}}_i^{(d-1)} + \binom{\pm s}{\pm s}$, see Fig. 5.3. Assume that no additional error is involved, and for an estimated $\hat{\mathbf{m}}_i^{(d-1)}$ is the real receiver $\mathbf{m}_i$ in a distance of at most $\epsilon = 2^{-d+\frac{1}{2}}$. Then, according to Lemma 5, the ϵ-test holds. When the area is partitioned as described, at least one receiver $\hat{\mathbf{m}}_i^{(d)}$ must exist for which the distance to $\mathbf{m}_i$ is at most $2^{-d-\frac{1}{2}}$, and therefore the $\frac{1}{2}\epsilon$-test holds.

The proof that in the partition step the whole search space is covered, and therefore a solution is always found, is straightforward: The receivers reside in the unit square, which is covered in the first depth level 0 by choosing $\epsilon = \sqrt{2}$. By partitioning the search space of edge length $4s$ into subsequent depth levels of edge length 2, where the receivers are shifted by s for every dimension, the whole search space is covered.

Together with the proof that for a cell that satisfies the test, at least one subcell satisfies the test, which follows from Lemma 5, ensures a solution of the refinement. Lemma 5 also proves the convergence of the algorithm according to Eq. (5.2), as the position error of receivers is now bounded by $s = \frac{1}{\sqrt{2}}\epsilon$ for a measurement error ϵ.

The algorithm ends if a node is reached where $\epsilon \leq \epsilon_{\text{target}}$. If there is additional TDoA error in a setting, the algorithm will discard a cell if the TDoA error ϵ_{tdoa} exceeds the error in the ϵ-test. Then, the ϵ-test will eventually fail for all child cells, so the queue

is emptied and the algorithm will terminate. In this case a solution up to the error bound ϵ is returned.

Once the positions of microphones are determined using the branch-and-bound algorithm, calculating the positions of signal sources is straightforward. Just execute the algorithm until $\epsilon \leq \epsilon_{\text{target}}$ or the queue is depleted. In the function for the ϵ-test return also the positions of intersections that enclose a *valid* segment. These are positions of signal sources bounded by an error proportional to ϵ.

In the minimum case of four receivers, the search space of this refinement algorithm is five, which is the typical execution size in order to keep the runtime small. In case that more receivers are subject to locate, calculate a subset of four receivers first using the proposed algorithm. Once these are known, calculate the positions of signals by the ϵ-test, then locate the other receivers by the signal positions, which can be done in closed form.

5.5 Numerical Evaluation of Four Receivers

We have implemented the algorithm in C++ to evaluate the runtime and the number of cells in practical experiments [96]. Due to the structure of the algorithm cell tests are independent, so we utilize multi-threaded processing of the queue with multiple worker threads. While in our implementation any number of $n \geq 4$ receivers can be used, we focus on four receivers, as the number of child cells in every step is as large as 2^{2n-3}. For problems with more receivers one can select a subset of four receivers, solve it using the algorithm and calculate the remaining receivers afterwards by simple hyperbolic equations.

For our experiments we generate random positions of four receivers in the unit circle in the plane. We normalize the receivers such that the first location is in the origin, the second is on the positive leg of the x-axis, and the third receiver is in the upper two quadrants. Furthermore, we create m random signals in the unit circle. For m we choose a series from 5 to 50 signals with 100 runs for each number of signals. The time differences of arrival are calculated and given to the algorithm, which is the only information available to the algorithm. The algorithm calculates the feasibility, splits and discards cells according to the specification in Algorithm 1. See Fig. 5.4 for a runtime example of four receivers and 20 signals.

We evaluated the algorithm in terms of inspected search nodes, search queue length, duration and correctness of the calculation, i.e. the receiver positions. In some underdetermined cases with few signal sources, or malicious receiver positions, where any two receivers are very close to each other, we observe run-times as indicted by the worst case analysis. Then, the width of the search tree grows rapidly resulting in high run-times. In these cases we abort the algorithm after 10 minutes and mark the attempt failed (although the algorithm would eventually find the solution).

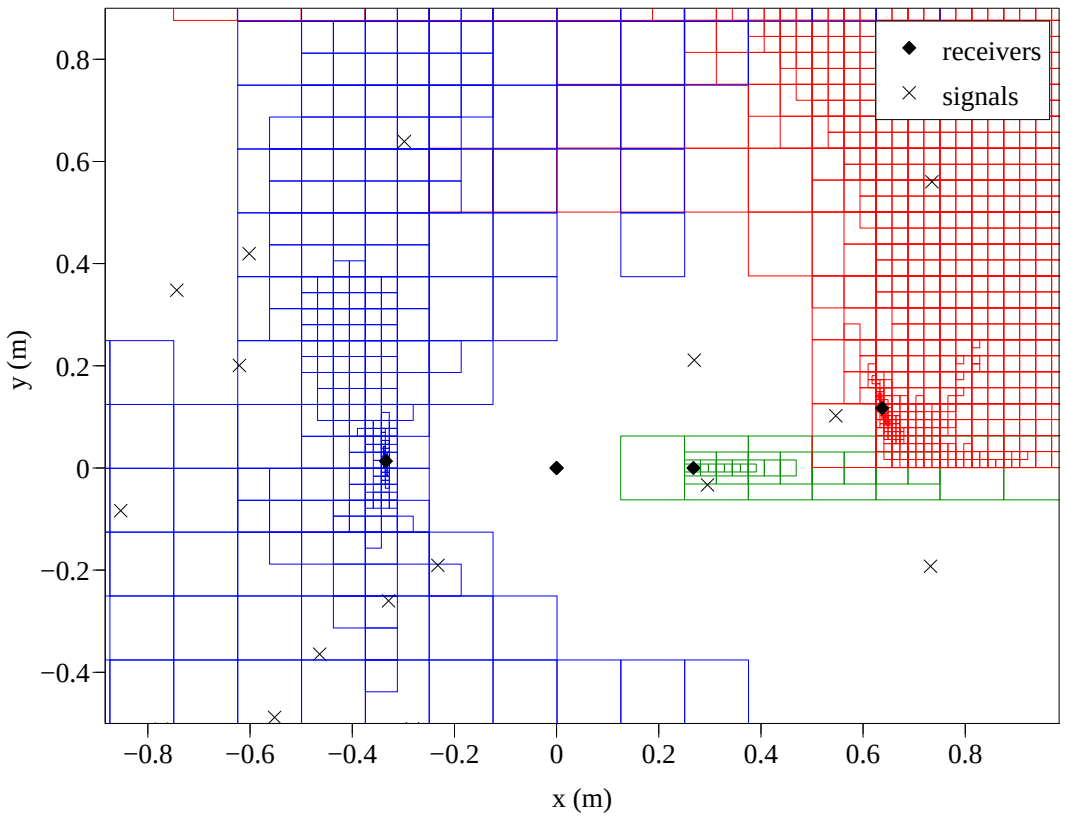


Figure 5.4: Runtime example of the branch-and-bound algorithm with recursive refinement of four receivers, 20 signal sources, and no TDoA error. Although various areas are explored up to a certain depth, the algorithm converges only at the correct receiver positions. For clarity only cells of depth $d \geq 4$ are displayed.

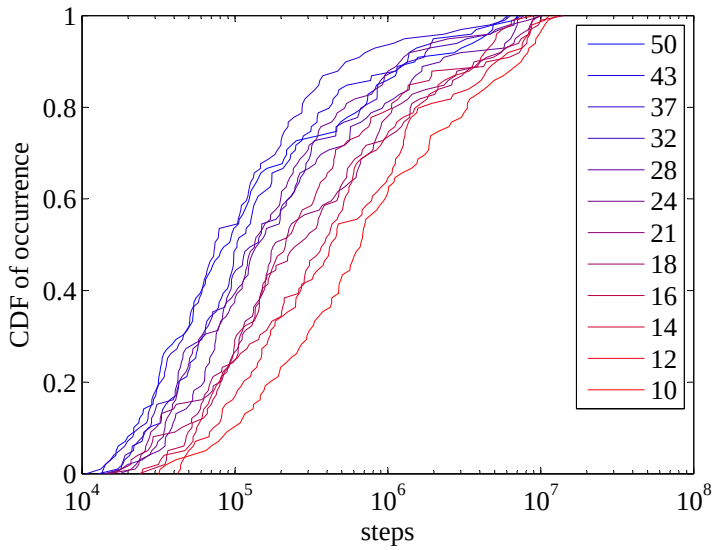


Figure 5.5: Cumulative distribution of step numbers given 10 – 50 signal sources. The mean is $6.7 \cdot 10^5$ steps, averaging to 44 seconds on a dual-core CPU. [96]

The number of required evaluation steps depends on the location characteristics of the receivers, on the traversal strategy through the tree, and on the number of signals. In our simulations we choose breadth-first search, which is the slowest search type, but with deterministic characteristics. Then, given a sufficient number of signals, the number of traversed nodes varies between 10^4 and 10^7 with a cumulation at 10^5. With decreasing number of signals the algorithm has increased difficulty to eliminate possible locations, increasing the number of steps by a factor of 10^1 and more, shifting the cumulation towards higher step numbers (Fig. 5.5).

On an Intel Core-i5 machine we could process a number of 10^5 nodes in about 4 – 8 seconds, running on four processor cores (Fig. 5.6). A typical execution time given 40 signals is 8 seconds, which is the mode of the distribution of runtime, with some ill-conditioned settings raising the mean to 10.5 seconds. Altogether, the typical execution time is 10 seconds with a mean of 44 seconds.

5.6 Summary

To our knowledge the presented branch-and-bound algorithm is the only algorithm by the time of this thesis that can find a solution in the sense of convergence to the minimum problem of calibration-free TDoA. This is not achieved by the discussed local optimization algorithms, not by those in literature, and neither by the factorization approach of Pollefeys which does not cover the minimum problem.

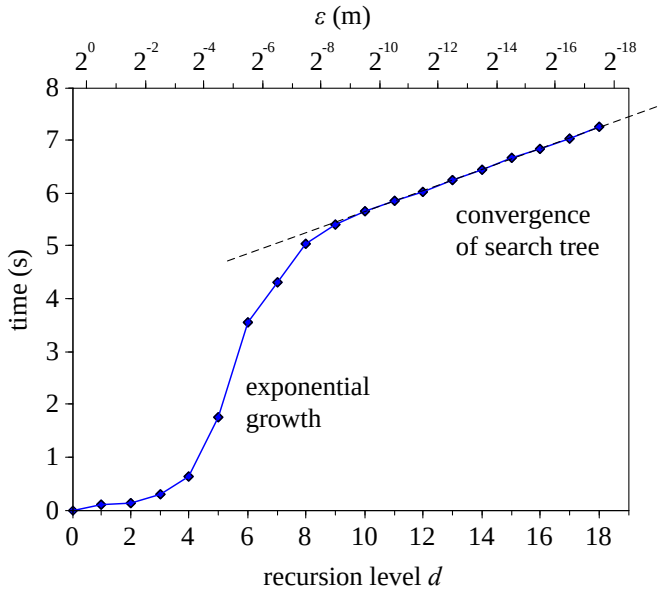


Figure 5.6: Total run-time of an example with four receivers and 20 signals on an Intel Core-i5 quad-core CPU to achieve precision ϵ. The recursion level is $d = \frac{1}{2} - \log_2(\epsilon)$. The run-time is proportional to the number of processed nodes. [96]

However, branch-and-bound search is costly in terms of the runtime in such large search dimensions. While approximation and local search algorithms find a solution that can be tested for correctness in fractions of a second, the runtime of the recursive search space exploration is in the range of seconds to minutes. In some cases our approach suffers from an ill-conditioned configuration of the receiver locations, i.e. some of the four receivers are close to each other or near to a line, rendering the problem close to under-determined. Then, our algorithm is forced to generate a very large search tree, resulting in a long duration for the traversal.

For an extension of the algorithm to three dimensions, one of the challenges is to extend the ϵ-test and calculate the uncertainty regions in three dimensions. The corners of these regions are defined by intersection points of exactly three hyperboloids in space. Many of the closed-form TDoA algorithms use TDoA *differences* to a reference receiver, as for instance [9], which increases the minimum number of hyperboloid equations to four. On the other hand, calculating the intersection points with iterative methods, even with the fast Gauss-Newton method, will horribly affect the runtime of the search algorithm. Fortunately, the hyperbolic intersection can be expressed in closed form, as demonstrated in [30], even though the equation is lengthy. Then, the search space will increase from five variables for estimating four receivers, when eliminating three variables for symmetry, to nine variables for estimating five receivers, eliminating six variables.

6 Probabilistic TDoA Localization

In Chapter 4 we saw that a solution to the general TDoA self-calibration problem is found by iterative optimization in many cases. However, the optimization approach does not consider spatial characteristics of subsequent signals, especially of a single sender that moves in space, emitting signals at a constant interval. We will now focus on the special case of tracking a moving signal beacon under the assumption of a coherent trajectory of the beacon, which is a typical scenario in a localization system. The scenario of a moving beacon has been considered in some recent approaches where a particle filter is used for TDoA localization [101, 102, 103, 104, 105], which is also known as Monte Carlo localization [106, 107, 108]. In the mentioned approaches the receiver positions are known in advance, and only the signal emitter is located, which requires a priori calibration of the receiver positions.

In cooperation with the Chair for Autonomous Intelligent Systems, University of Freiburg, we have developed a probabilistic approach based on a particle filter. We assume that receivers are stationary, yet their positions are unknown in the beginning, as in the general calibration-free TDoA problem [109]. We propose a sensor model which considers the Gaussian characteristics of the TDoA measurements at the receivers and explicitly takes into account measurement outliers, increasing the robustness of the approach to effects such as multipath signal propagation (echoes), which are characteristic for indoor environments. In our motion model we assume a continuous movement of the signal beacon, which enables a quick recovery in case of temporary signal loss. Initialization of the particle filter estimate is achieved by multiple iterative optimization attempts on randomly chosen subsets of the available measurement data. Due to the weighting of the particles according to our TDoA sensor model, the particle filter implicitly selects the correct initialization hypothesis and reliably converges to the true configuration.

6.1 Problem Formulation

We consider the problem of simultaneous estimation of a continuous trajectory $\mathbf{s}_h$ $(h = 1, \ldots, m)$ of a signal beacon and the stationary positions $\mathbf{m}_i$ $(i = 1, \ldots, n)$ of n receivers in Euclidean space $\mathbb{R}^p$, where $p \in \{2, 3\}$. At each discrete time step h, the beacon emits a discrete signal at the unknown send time t_h. Other than in the general setting description in Section 2.1 the signals are assumed to occur in a regular manner. The signal is received and uniquely identified by some or all of the receivers at the times T_{ih}. For estimating the beacon trajectory and the receiver positions only

the arrival times T_{ih} of the signal are available. Up to time h the measurement data $\mathbf{z}_{1:h} = \{\mathbf{z}_j \mid j = 1, \ldots, h\}$ is available. As the range of the sound signal is limited and its propagation may be inhibited by obstacles, at each time step the measurement data $\mathbf{z}_j \subseteq \{T_{ij} \mid i = 1, \ldots, n\}$ is a subset of the possible reception times of all receivers.

We assume that the signal propagates from the beacon to the receivers in a straight line with a constant signal velocity c according to the signal propagation equation

$$\|\mathbf{m}_i - \mathbf{s}_h\| = c\,(T_{ih} - t_h)\,, \tag{6.1}$$

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

Recall that in the general TDoA localization problem [4, 54] the positions of signal beacons $\mathbf{s}_j$ and receivers $\mathbf{m}_i$ are arbitrary in space. As no anchor positions are provided, the position and the orientation of the signal beacon and the receivers relative to global coordinates is undetermined. The estimated solution can be compared to references by aligning it using a congruent transformation.

6.2 Robust State Estimation in the Particle Filter

We consider the estimation of the continuous trajectory of the signal beacon and the positions of the stationary receivers as a recursive state estimation problem. For that, we apply the Bayesian filtering scheme [108], which is a probabilistic approach to recursive state estimation. It is suitable to TDoA localization as the state can be computed online and it is robust to motion and measurement uncertainty. We have published such an approach in [109], from which the following description is adopted.

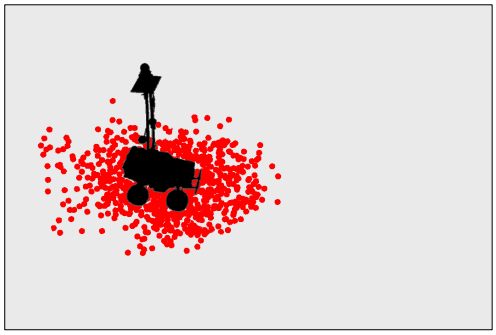
The key idea of the recursive Bayesian filter is to maintain a probability density $p(\mathbf{x}_h \mid \mathbf{z}_{1:h}, \mathbf{u}_{1:h})$ of the state $\mathbf{x}_h$ at time h. The state is conditioned on all sensor data $\mathbf{z}_{1:h}$ and control commands $\mathbf{u}_{1:h}$ up to that time h. This probability density which is also called posterior distribution can be factorized into

$$\begin{aligned} &p(\mathbf{x}_h \mid \mathbf{z}_{1:h}, \mathbf{u}_{1:h}) \\ &\quad = \eta_h\, p(\mathbf{z}_h \mid \mathbf{x}_h) \int p(\mathbf{x}_h \mid \mathbf{u}_h, \mathbf{x}_{h-1})\, p(\mathbf{x}_{h-1} \mid \mathbf{u}_{1:h-1}, \mathbf{z}_{1:h-1})\ \mathrm{d}\mathbf{x}_{h-1}\,, \end{aligned} \tag{6.2}$$

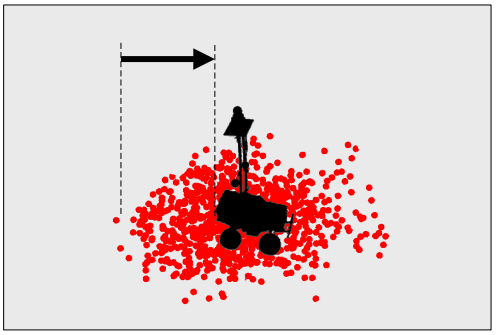
where η_h is a normalizing constant ensuring that $\int p(\mathbf{x}_h \mid \mathbf{z}_{1:h}, \mathbf{u}_{1:h})\ \mathrm{d}\mathbf{x}_h = 1$. The term $p(\mathbf{x}_h \mid \mathbf{u}_h, \mathbf{x}_{h-1})$ is the state transition probability and $p(\mathbf{z}_h \mid \mathbf{x}_h)$ is the measurement probability specified by the motion model and the sensor model, respectively.

Various implementations of the recursive Bayesian filter exist. The Kalman filter assumes Gaussian uncertainty and linear system dynamics. There are various extensions for nonlinear systems [108]. However, their performance degrades with increasing nonlinearities [110], and the Gaussian uncertainty assumption is not valid on TDoA data with additional measurement outliers.

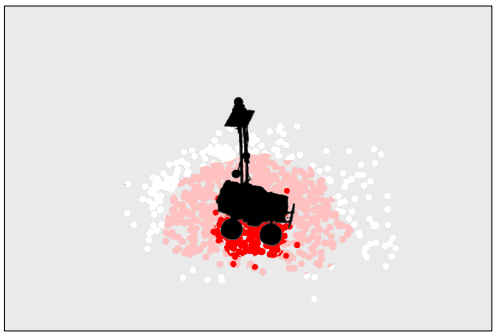
We apply the particle filter, which approximates the current belief $p(\mathbf{x}_h \mid \mathbf{z}_{1:h}, \mathbf{u}_{1:h})$ by a set $\mathcal{M} = \{(\mathbf{x}^{[k]}, w^{[k]})\}_{(k=1,\ldots,N)}$ of N particles. Each particle corresponds to a state



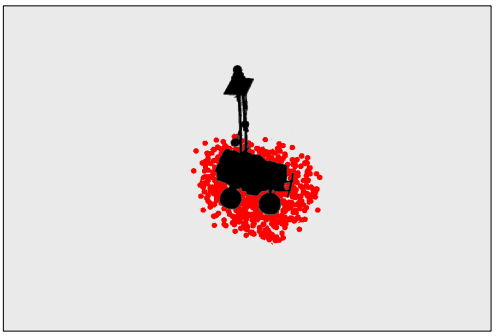
(a) The weighted set of particles represents the posterior distribution, the *belief* of the moving beacon.



(b) The belief is updated in the *prediction* step according to the motion model and to control input.



(c) In the *correction* step the particles are weighted according to the sensor model for the probability of observations.



(d) In the *resampling* step the distribution of the particles is refreshed according to the weights of the particles.

Figure 6.1: Scheme of Monte Carlo Localization using a *particle filter* [108], a probabilistic recursive state estimation algorithm based on *particles*, consisting of the components prediction, correction, and resampling.

hypothesis $\mathbf{x}^{[k]}$ weighted by the so-called importance weight $w^{[k]}$, where the weight is normalized by $\sum_{k=1}^{N} w^{[k]} = 1$. In the particle filter, we perform the recursive belief update given in (6.2) according to the following three steps:

1. In the *prediction step*, we propagate each particle by drawing a successor state based on the proposal distribution $p(\mathbf{x}_h^{[k]} \mid \mathbf{u}_h, \mathbf{x}_{h-1}^{[k]})$ specified by the motion model. Thereby, the control command $\mathbf{u}_h$ and the motion uncertainty is taken into account.
2. In the *correction step*, we integrate a new measurement $\mathbf{z}_h$ by updating the weight $w_h^{[k]} \propto w_{h-1}^{[k]}\, p(\mathbf{z}_h \mid \mathbf{x}_h^{[k]})$ of each particle according to the measurement likelihood given its state hypothesis. The measurement likelihood is defined in the sensor model taking into account the measurement process and the sensor characteristics.
3. In the *resampling step*, we draw a new set of particles from $\mathcal{M}$ with replacement such that each particle in $\mathcal{M}$ is selected with a probability that is proportional to its weight.

We reduce the particle depletion by applying low variance resampling [108] and delay the resampling step until the effective number of particles $N_{\text{eff}} = \left(\sum_{k=1}^{N} \left(w^{[k]}\right)^2\right)^{-1}$ drops below half the number of particles [111]. See Fig. 6.1 for an illustration of the particle filter principle.

The initialization of the filter and the design of the sensor and motion models is substantial for precise and reliable state estimation. In the Sections 6.2.2 to 6.2.4 we describe our approach to robust initialization of the particle filter, a constant velocity motion model suitable for standard beacon tracking, and the TDoA sensor model.

For tracking of a mobile beacon, we define the state $\mathbf{x}_h$ at time h as

$$\mathbf{x}_h = \left(\mathbf{s}_h^T, \mathbf{v}_h^T, \mathbf{m}_{1h}^T, \ldots, \mathbf{m}_{nh}^T\right)^T . \tag{6.3}$$

Here, $\mathbf{s}_h$ is the position and $\mathbf{v}_h$ is the velocity of the beacon. Furthermore, we neglect the fact that the receiver positions $\mathbf{m}_i$ are assumed to be stationary, which will be taken into account in the motion model as described in Section 6.2.3.

6.2.1 Calculation of the Particle Filter Estimate

We estimate both, the position of the signal beacon, as well as the positions of the receivers. For calculation of the filter estimate from the weighted set of particles we approximate the posterior distribution $p(\mathbf{x}_h \mid \mathbf{z}_{1:h}, \mathbf{u}_{1:h})$ by a Gaussian. According to that approximation, the filter estimate $\hat{\mathbf{x}}_h$ is the weighted average of the particles which is the mean of the Gaussian distribution.

In our state estimation problem the TDoA measurement data specifies only the individual distances between the receivers and the signal origins, i.e., the internal configuration of the system. Consequently, the global translation and rotation of the state $\mathbf{x}^{[k]}$ of every particle in the filter is arbitrary. To evaluate the weighted average

of all particles we globally align the particles in a congruent transformation. The constraints inside the configuration estimated by every particle, and the distances between the beacon and the receivers remain unchanged in this procedure.

By following the approach of Arun et al. [57] we use singular value decomposition to identify the transformation parameters between two particle states $\mathbf{x}^{[k]}$ and $\mathbf{x}^{[l]}$, the rotation matrix $\mathbf{R}_{kl}$ and the translation vectors $\boldsymbol{\mu}_k$ and $\boldsymbol{\mu}_l$. As the configuration of the receivers can be mirror-inverted, the determinant of the rotation is $|\mathbf{R}_{kl}| = \pm 1$. See Section 2.6 for the calculation of $\mathbf{R}_{kl}$.

We align all particles to the most likely particle, which we assume to be the first particle in the set. The alignment of every receiver and the signal beacon in a particle is calculated by

$$\begin{aligned}\tilde{\mathbf{m}}_i^{[k]} &= \mathbf{R}_{1k}(\mathbf{m}_i^{[k]} - \boldsymbol{\mu}^{[k]}) + \boldsymbol{\mu}^{[1]} \\ \tilde{\mathbf{s}}^{[k]} &= \mathbf{R}_{1k}(\mathbf{s}^{[k]} - \boldsymbol{\mu}^{[k]}) + \boldsymbol{\mu}^{[1]}\end{aligned}$$

from the rotation matrix and the mean of each estimate of the receiver positions. Analogously, the velocity estimate is transformed by $\tilde{\mathbf{v}}^{[k]} = \mathbf{R}_{1k}\mathbf{v}^{[k]}$.

Now, the weighted mean and the covariance of the particle filter with normalized weights can be calculated by

$$\begin{aligned}\boldsymbol{\mu}_h &= \sum_{k=1}^{N} w^{[k]} \tilde{\mathbf{x}}_h^{[k]} \\ \boldsymbol{\Sigma}_h &= \sum_{k=1}^{N} w^{[k]} (\tilde{\mathbf{x}}_h^{[k]} - \boldsymbol{\mu}_h)(\tilde{\mathbf{x}}_h^{[k]} - \boldsymbol{\mu}_h)^T \ .\end{aligned}$$

6.2.2 Initialization

As long as the particle filter is in the initialization stage, we try to initialize the filter using an iterative non-probabilistic error minimization approach. In this stage, we periodically initialize a certain proportion of the particles according to the result of an iterative minimization on a subset of TDoA measurement data as follows.

In an initialization attempt at time step h, we use gradient descent, followed by the Gauss-Newton method [54] to optimize the signal propagation constraint (6.1). For that, we randomly choose $\tilde{m} < h$ beacon signals including the latest signal $\{\mathbf{z}_{I_1}, \ldots, \mathbf{z}_{I_{\tilde{m}-1}}, \mathbf{z}_h\} \subseteq \mathbf{z}_{1:h}$ as a subset of the available measurement data, where I_j $(j = 1, \ldots, \tilde{m}-1)$ is the random index of the subset. For simplicity in notation, in the following we use j to denote the measurement with the index I_j.

The signal propagation constraint (6.1) yields a maximum of $\tilde{m}n$ equations if all receivers obtain all signals, with $(p+1)\tilde{m} + pn$ unknown variables. By combining two equations for receivers i and k $(1 \leq i, k \leq n)$ and for a signal j the signal emission time t_j is eliminated, reducing the maximum number of equations to $\tilde{m}(n-1)$ equations

and $p\tilde{m} + pn$ variables. We obtain the hyperbolic equation

$$\|\mathbf{m}_i - \mathbf{s}_j\| - \|\mathbf{m}_k - \mathbf{s}_j\| = c\,(T_{ij} - T_{kj})\ .$$

We define the state vector with signal and receiver positions to optimize as

$$\boldsymbol{\psi} \ = \ \left(\tilde{\mathbf{s}}_1^T, \ldots, \tilde{\mathbf{s}}_{\tilde{m}}^T, \tilde{\mathbf{m}}_1^T, \ldots, \tilde{\mathbf{m}}_n^T\right)^T\ .$$

Without loss of generality, we choose $k = 1$ and define the error functional

$$f_{ij}(\boldsymbol{\psi}) \ = \ \|\tilde{\mathbf{m}}_i - \tilde{\mathbf{s}}_j\| - \|\tilde{\mathbf{m}}_k - \tilde{\mathbf{s}}_j\| - c\,(T_{ij} - T_{kj})\ ,$$

which we combine into a hyperbolic error function

$$\mathbf{b}(\boldsymbol{\psi}) \ = \ \left(f_{21}(\boldsymbol{\psi}), f_{22}(\boldsymbol{\psi}), \ldots, f_{n\tilde{m}}(\boldsymbol{\psi})\right)^T\ .$$

We minimize $f_{ij}(\boldsymbol{\psi})$ by using gradient descent and refine the solution with the Gauss-Newton method afterwards, which is described in detail in Section 4.1. The result of the iterative approach is an approximation of the latest position of the beacon $\mathbf{s}_h$ and the stationary positions of the receivers $\mathbf{m}_1$ to $\mathbf{m}_n$:

$$\tilde{\mathbf{x}} \ = \ \left(\tilde{\mathbf{s}}_h^T, \vec{0}^T, \tilde{\mathbf{m}}_1^T, \ldots, \tilde{\mathbf{m}}_n^T\right)^T\ .$$

We inject this approximation as a possible initialization into the particle filter by sampling a certain proportion of the particle set from the multivariate Gaussian distribution $\mathcal{N}(\tilde{\mathbf{x}}, \Sigma_{\text{Newton}})$. Here, the covariance Σ_{Newton} of the Newton optimization results with respect to the true state can be determined straight forward from recorded data.

Some of these attempts of the iterative algorithm may not be successful. In fact, especially in the beginning, when only few signals are given and the signal beacon did not move that far, so that the positions of the signal emissions are not well distributed, most optimization attempts will fail or converge to a local minimum. However, the probability of finding the correct solution is increased by repeated attempts [54]. Furthermore, by using varying subsets of data, we circumvent adverse scenarios where the iterative algorithm never finds a solution.

After some time, the initialization algorithm has provided an estimate which is close to the true configuration, and the particle filter converges towards the correct positions. We detect this by continuously evaluating the internal TDOA residual $\|(f_1, \ldots, f_n)^T\|$ of the particle filter mean $\boldsymbol{\mu}_h$, where

$$f_i(\boldsymbol{\mu}_h) \ = \ \|\tilde{\mathbf{m}}_{1h} - \tilde{\mathbf{s}}\| - \|\tilde{\mathbf{m}}_{ih} - \tilde{\mathbf{s}}\| - c\,(T_{1h} - T_{ih})\ .$$

After a possible initialization was found, a lowpass filtered indicator of the TDOA residual will fall below a threshold ϵ_{init}, which is proportional to the noise of the system. Empirically, we have determined this value to $\epsilon_{\text{init}} = 1$, which is suitable for all experiments that we conducted, reliably indicating proper convergence.

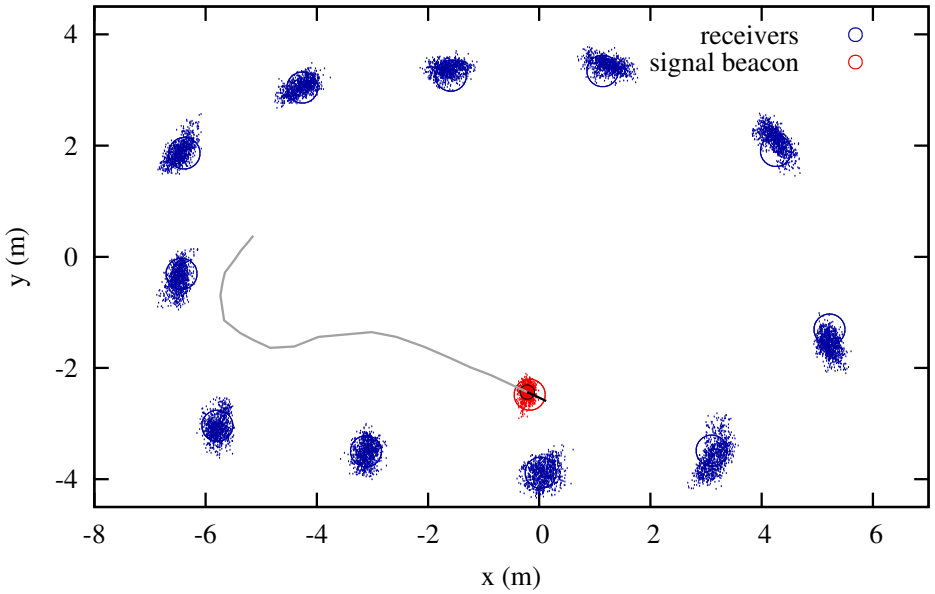


Figure 6.2: A visual representation of the state estimate of the particle filter. The receivers are shown in blue and the signal beacon in red. The ground truth positions are depicted as blue and red circles, respectively. [109]

Naturally, the internal TDOA residual of the position estimate is not a safe guarantee that a possible solution of low TDOA residual is also the correct solution with low deviation from the reference positions. The discrepancy becomes evident when additional measurements are received which are not compatible to the current state, so the TDOA residual will then increase to a high level. We allow the algorithm to re-enable initialization mode if such an inconsistency occurs, which is triggered by the internal error rising above a threshold of $\epsilon_{\text{retry}} = 2$. See Fig. 7.12 in Chapter 7 for an illustration of initialization.

6.2.3 Motion Model

The proper design of the probabilistic motion model is substantial for the efficiency and the accuracy of the state estimation during the prediction step of the particle filter. In the context of mobile beacon tracking with simultaneous calibration of the receiver positions we factorize the probabilistic motion model into

$$p(\mathbf{x}_h^{[k]} \mid \mathbf{u}_h, \mathbf{x}_{h-1}^{[k]}) = p(\mathbf{s}_h^{[k]}, \mathbf{v}_h^{[k]} \mid \mathbf{u}_h, \mathbf{s}_{h-1}^{[k]}, \mathbf{v}_{h-1}^{[k]}) \cdot \prod_{j=1}^{n} p(\mathbf{m}_{jh}^{[k]} \mid \mathbf{u}_h, \mathbf{m}_{jh-1}^{[k]}) \ , \tag{6.4}$$

where we assume independent motion of the beacon and the individual receivers.

In most applications no control of the movement of the beacon is given. Hence, we assume the control command $\mathbf{u}_h = \Delta h$ to contain only the time elapsed since the

last prediction step and apply a constant velocity motion model. According to these assumptions the probabilistic model is

$$\mathbf{s}_{h+1} = \mathbf{s}_h + \Delta h\, \mathbf{v}_h \tag{6.5}$$

$$\mathbf{v}_{h+1} = \mathbf{v}_h + \Delta h\, \boldsymbol{\xi} \qquad \text{with} \quad \boldsymbol{\xi} \sim \mathcal{N}(\vec{0}, \Sigma_{\mathbf{v}})\ . \tag{6.6}$$

The model is inspired by the Euler integration method [80]. It assumes that the beacon moves with a constant velocity, and models changes in the velocity by an Gaussian noise term $\boldsymbol{\xi}$, instead of acceleration. The covariance matrix $\Sigma_{\mathbf{v}}$ can be determined from empirical data.

The positions of the stationary receivers do not have to be modified so that the probability density function of the motion model

$$p(\mathbf{m}_{jh}^{[k]} \mid \mathbf{u}_h, \mathbf{m}_{jh-1}^{[k]}) = \delta(\mathbf{m}_{jh-1}^{[k]}) \tag{6.7}$$

of the receivers is the Dirac delta distribution. This results in a sample degeneracy, or "attrition", which is usually prevented by adding small random Gaussian disturbances to the receiver positions of each sample during resampling. However, this leads to a loss of information in each resampling step, and hence is a suboptimal solution. We use *kernel smoothing* [112] instead, as soon as the particle filter is converged:

$$\mathbf{m}_{jh}^{[k]} \sim \mathcal{N}(\mathbf{m}_{jh}^{[k]}; \alpha\, \mathbf{m}_{jh-1}^{[k]} + (1-\alpha)\, \overline{\mathbf{m}}_{jh-1}, \rho^2 \Sigma_{M_{jh-1}})\ . \tag{6.8}$$

$\overline{\mathbf{m}}_{jh-1}$ and $\Sigma_{M_{jh-1}}$ are the weighted mean and covariance of the j-th receiver position over the particle set at time $h-1$. The constant factors $\alpha = \frac{3\gamma-1}{2\gamma}$ and $\rho^2 = 1-\alpha^2$ only depend on a discount factor, which we set to $\gamma = 0.95$. Compared to adding disturbances, kernel smoothing prevents a loss of information in the receiver positions. Before each kernel smoothing step the particle set is aligned as described in Section 6.2.1.

6.2.4 Sensor Model

The probabilistic sensor model $p(\mathbf{z} \mid \mathbf{x})$ defines the likelihood of the measurement data $\mathbf{z}$ given the state $\mathbf{x}$ of the system. Here, the measurement data $\mathbf{z}_h = \{z_1, \ldots, z_k\} \subseteq \{T_{ih} \mid i = 1, \ldots, n\}$ is a subset of all possible reception times of the beacon signal emitted at time t_h measured by the receivers. We assume the measurements of the individual receivers to be independent, given the current state $\mathbf{x}$ of the system, i.e., the position of the beacon and the receivers:

$$p(\mathbf{z} \mid \mathbf{x}) = \prod_{j=1}^{k} p(z_j \mid \mathbf{x})\ . \tag{6.9}$$

Based on Eq. (6.1) the measurement process of each individual measurement z_j is described by

$$z_j - t_h = \frac{1}{c}\|\mathbf{m}_j - \mathbf{s}_h\| + \epsilon_j\ . \tag{6.10}$$

In our model, we assume that the corresponding error variable ϵ_j is basically Gaussian distributed with an additional small proportion of extreme outliers induced by external influences like reflections of the signal and echoes. Consequently, we apply a mixed model combining a Gaussian and a uniform distribution:

$$p(\epsilon_j) \;=\; (1-\alpha)\frac{1}{2\lambda} + \alpha\,\mathcal{N}(\epsilon_j; 0, \sigma^2)\;. \tag{6.11}$$

Here, λ is the maximum error of measurement outliers. $\mathcal{N}(\circ; \mu, \sigma^2)$ denotes the density function of the Gaussian distribution. The proportion of measurement outliers is defined by the parameter $0 \leq \alpha \leq 1$.

Based on this measurement error, the measurement likelihood is defined as

$$p(z_j \mid \mathbf{x}) \;=\; (1-\alpha)\frac{1}{2\lambda} + \alpha\,\mathcal{N}(z_j; \mu_j, \sigma^2) \tag{6.12}$$

with the mean $\mu_j = t_h + \frac{1}{c}\|\mathbf{m}_j - \mathbf{s}_h\|$.

To maximize the likelihood of the measured data with respect to the expected measurements given the current pose of the beacon, we estimate the most likely send time $\hat{t}_h$:

$$\hat{t}_h \;=\; \arg\max_{\tilde{t}_h} p(\mathbf{z} \mid \mathbf{x}, \tilde{t}_h) \;=\; \arg\max_{\tilde{t}_h} \prod_{j=1}^{k} p\left(z_j \mid \mathbf{x}, \tilde{t}_h\right)\;. \tag{6.13}$$

Maximizing Eq. (6.13) requires to solve

$$\begin{aligned}
&\frac{\mathrm{d}}{\mathrm{d}\hat{t}_h} p(\mathbf{z} \mid \mathbf{x}, \hat{t}_h) \\
&= \frac{\mathrm{d}}{\mathrm{d}\hat{t}_h} \prod_{j=1}^{k} p\left(z_j \mid \mathbf{x}, \hat{t}_h\right) \\
&= \frac{\mathrm{d}}{\mathrm{d}\hat{t}_h} \prod_{j=1}^{k} \left((1-\alpha)\frac{1}{2\lambda} + \alpha\,\mathcal{N}\left(z_j; \hat{t}_h + \frac{\|\mathbf{m}_j - \mathbf{s}_h\|}{c}, \sigma^2\right)\right) \quad \stackrel{!}{=} \quad 0\;.
\end{aligned} \tag{6.14}$$

This product results in a binomial term of degree k, depending on the number of available receiver measurements. For that derivative, we see no way to find a direct solution in a closed form. A practical solution can be found using iterative optimization with only a few iterations. However, as the likelihood maximization is calculated in every observation step and for every particle in the filter, this will eventually be very slow.

Hence, we approximate the most likely send time by neglecting the (comparably small) uniform part of the measurement likelihood and obtain

$$\begin{aligned} 0 &= \frac{\mathrm{d}}{\mathrm{d}\hat{t}_h} \prod_{j=1}^{k} \mathcal{N}\left(z_j; \hat{t}_h + \frac{\|\mathbf{m}_j - \mathbf{s}_h\|}{c}, \sigma^2\right) \\ &= \left(\frac{1}{\sqrt{2\pi}\,\sigma}\right)^k \exp\left(-\frac{\sum_{j=1}^{k}\left(z_j - \hat{t}_h - \frac{\|\mathbf{m}_j - \mathbf{s}_h\|}{c}\right)^2}{2\sigma^2}\right) \frac{1}{\sigma^2} \sum_{j=1}^{k}\left(z_j - \hat{t}_h - \frac{\|\mathbf{m}_j - \mathbf{s}_h\|}{c}\right). \end{aligned}$$

As the first and the second factor are always greater than zero, the equation can be rewritten as

$$0 = \frac{1}{\sigma^2} \sum_{j=1}^{k}\left(z_j - \hat{t}_h - \frac{\|\mathbf{m}_j - \mathbf{s}_h\|}{c}\right),$$

which finally can be solved to

$$\hat{t}_h = \frac{1}{k} \sum_{j=1}^{k}\left(z_j - \frac{\|\mathbf{m}_j - \mathbf{s}_h\|}{c}\right).$$

With that approximation of $\hat{t}_h$, we calculate the likelihood of a measurement $\mathbf{z}_h$ in (6.9) with

$$p(z_j \mid \mathbf{x}) \;=\; (1-\alpha)\frac{1}{2\lambda} + \alpha\,\mathcal{N}(z_j; \hat{t}_h + \frac{1}{c}\|\mathbf{m}_j - \mathbf{s}_h\|, \sigma^2)\,, \tag{6.15}$$

taking into account measurement outliers for robust beacon tracking in the particle filter. Also refer to [109] where we published the motion and sensor model.

6.3 Parameter Estimation of the Particle Filter

An essential component of probabilistic state estimation is the choice of parameters for the sensor and motion model. As described in Section 6.2.4, the sensor model relies on two parameters – the proportion of the linear component α and the sensor noise parameter σ^2, the variance of the Gaussian proportion of the sensor model. In an experiment with different values for the sensor model we saw that smaller values increase the risk of divergence of the particle filter, while much larger values lead to a loss of precision. In the empirical evaluation of the ultrasound system we observed a typical standard deviation for time measurement of $\hat{\sigma} \approx 0.15\,\mathrm{ms}$. For TDoA measurements leads such a small noise to a very highly peaked probability distribution, which is not well represented by discrete particles, especially if the number of particles is small. Therefore, we choose slightly larger values for the estimated sensor noise σ, trading precision for increased robustness of the particle filter. A suitable choice for the given empirical data of our experiments appears to be $0.3 \leq \sigma \leq 0.4$, resulting in minimum overall positioning error of the beacon, see Fig. 6.3. In the evaluation of the filter in

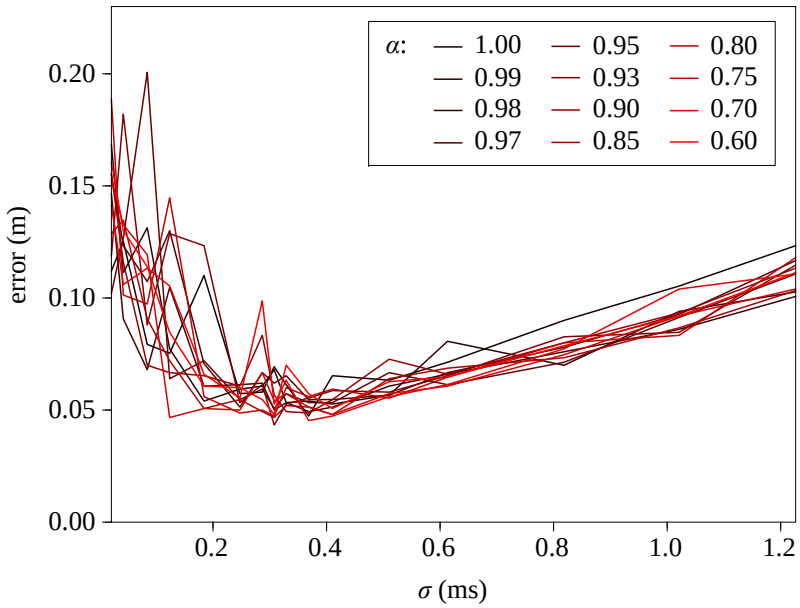


Figure 6.3: Estimation of the sensor model noise parameter in experimental data set *Exp. 1a* with measurement outliers up to 100 ms. The data points represent the median of estimation error of individual runs. The optimum is $\sigma = 0.4$ ms for different linear component parameters α.

Chapter 7 we use $\sigma = 0.3$ ms. For the linear component parameter α the influence is smaller, and a trend for different choices of α is hardly noticeable. Only if we choose α close to one, therefore no uniform noise component is assumed, then the particle filter appears to diverge in some cases. In our experiments we choose $\alpha = 0.8$.

For the constant velocity motion model, which we described in Section 6.2.3, the procedure is rather uncritical. We found a standard deviation of at least $\sigma_v \geq 0.5$ appropriate for elements of the covariance matrix $\Sigma_{\mathbf{v}} = (\sigma_v)^2 \cdot \mathbf{I}$, where $\mathbf{I}$ is the identity matrix. According to Eq. (6.6), we add such a noise term to the velocity estimate in every prediction step.

We also ran an evaluation of the required number of particles using the real-world data sets, which are described later in Chapter 7. We have tested values from less than 100 particles up to 200,000 particles. If at least 100 particles are given then the particle filter can track the beacon most of the time, see Fig. 6.4. If 1000 particles are given the localization becomes reliable, which is remarkably few, considering the high dimensionality of the state space. We attribute this to the fact that we enforce initialization of the particle filter by the iterative algorithm. A high number of more than 100,000 particles does still increase the precision of localization, yet at the cost of high computational load. In the evaluation of the particle filter in the next chapter we use a high number of 40,000 particles to avoid aliasing effects due to sparse distribution of particles.

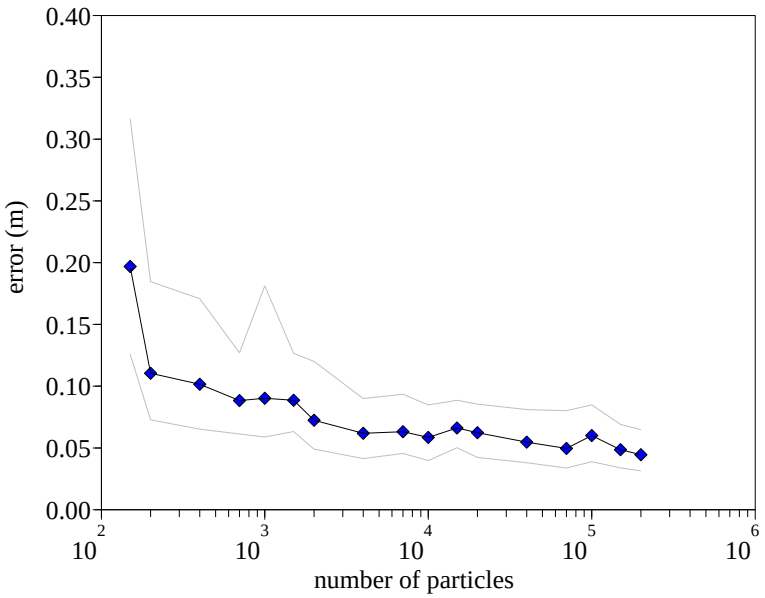


Figure 6.4: Evaluation of the required number of particles in experimental data set *Exp. 1a* with measurement outliers up to 100 ms. The data points represent the median of estimation error of individual runs, and the 30/70 percentiles (gray). A higher number of particles increases the precision of localization, yet also the computational load.

6.4 Summary

In this chapter, we have presented an approach to robust localization of a mobile beacon using TDoA measurement data when no receiver positions are given a priori. We cope with the high-dimensional state estimation problem, which typically has ambiguities in the measurements and several local optima, through a probabilistic formulation in a particle filter. The filter is reliably initialized through multiple attempts of an iterative optimization algorithm and implicitly selects the correct state hypothesis. In this way, our algorithm is robust to measurement outliers in the initialization phase. Additionally, we detect the proper initialization of the filter and ensure the termination of the initialization phase by monitoring the internal TDOA residual.

We have experimentally determined the parameters of the particle filter based on empirical measurements. Furthermore, we have evaluated the robustness and the accuracy of the particle filtering approach using our ultrasound localization system, which is described in the following chapter in Section 7.3.3. In these experiments, we demonstrated that in comparison to tracking a moving beacon by iterative optimization, our probabilistic approach is superior in terms of localization error, especially in case of measurement outliers. Furthermore, we have shown that the filter successfully initialized in less than 35 seconds in all experiments.

At its current state, the particle filtering approach is implemented for state estimation in two-dimensional environments. We feel positive about the implementation in three dimensions, as for all components of the algorithm, the state vector size, the motion and sensor model, the SVD alignment, and the initialization algorithm, the conversion is straightforward. However, it might be necessary to increase the number of particles to compensate for the higher dimensionality of the state space. Both, the increased higher vector size and the higher number of particles will increase the computational cost of the algorithm. Then, we will be able to carry out further experiments localizing flying vehicles and smart phones using robust calibration-free TDoA.

7 Real-World Experiments

We have created a localization system to evaluate the practical feasibility of our TDoA algorithms in a physical test environment [5]. With the localization system we pursue two goals. On the one hand, we generate a test bench for elusive influences from the environment that emerge only in real-world tests, for instance time synchronization and reverberation issues. We evaluate the performance of our approaches under these conditions, especially the practical Cone Alignment and particle filter algorithms. Second, we use our approaches in practical application and create a localization solution based on sound and ultrasound. Especially, in indoor environments such a solution may compensate for the lack of a reference system such as GPS.

In our system, multiple receiver devices connect in a wireless or wired network, where they record sounds and exchange time differences of acoustic events over the network. By using the time differences as input for our algorithms, we can self-calibrate the positions in the network. An application of this can be locating the laptops in a computer pool, just by recording the noises of people in the room. Another goal is to find the positions or to follow the trajectory of an acoustic signal source, which can be combined with self-calibration of the receivers. Then we obtain a "plug and play" acoustic tracking system for indoor environments.

7.1 The Localization Framework

Our localization system consists of multiple distributed receivers, which are computers, and a software system, running on each computer. The devices are connected to an acoustic receiver such as a microphone, or a USB device. To date we use ordinary x86-based laptops running Windows or Linux, the ARM-based embedded *Gumstix* computers and the Apple iPhone. As the communication protocol is system independent an arbitrary mix of platforms and systems may be used in the same experiment. The only limitation is the availability of receiver types for every platform. The x86-laptops can use built-in microphones, they and the Gumstix based devices support USB-connected sound and ultrasound microphones, on iPhone are only the built-in microphones available [51, 54, 109, 113]. See Fig. 7.1 for a screenshot of the Windows GUI of the software. The prototypical software framework that we have designed and implemented consists of modules for network connection management, time synchronization, audio recording and processing, algorithms for TDoA localization, and an interface for visualization. Fig. 7.2 overviews the components of the framework.

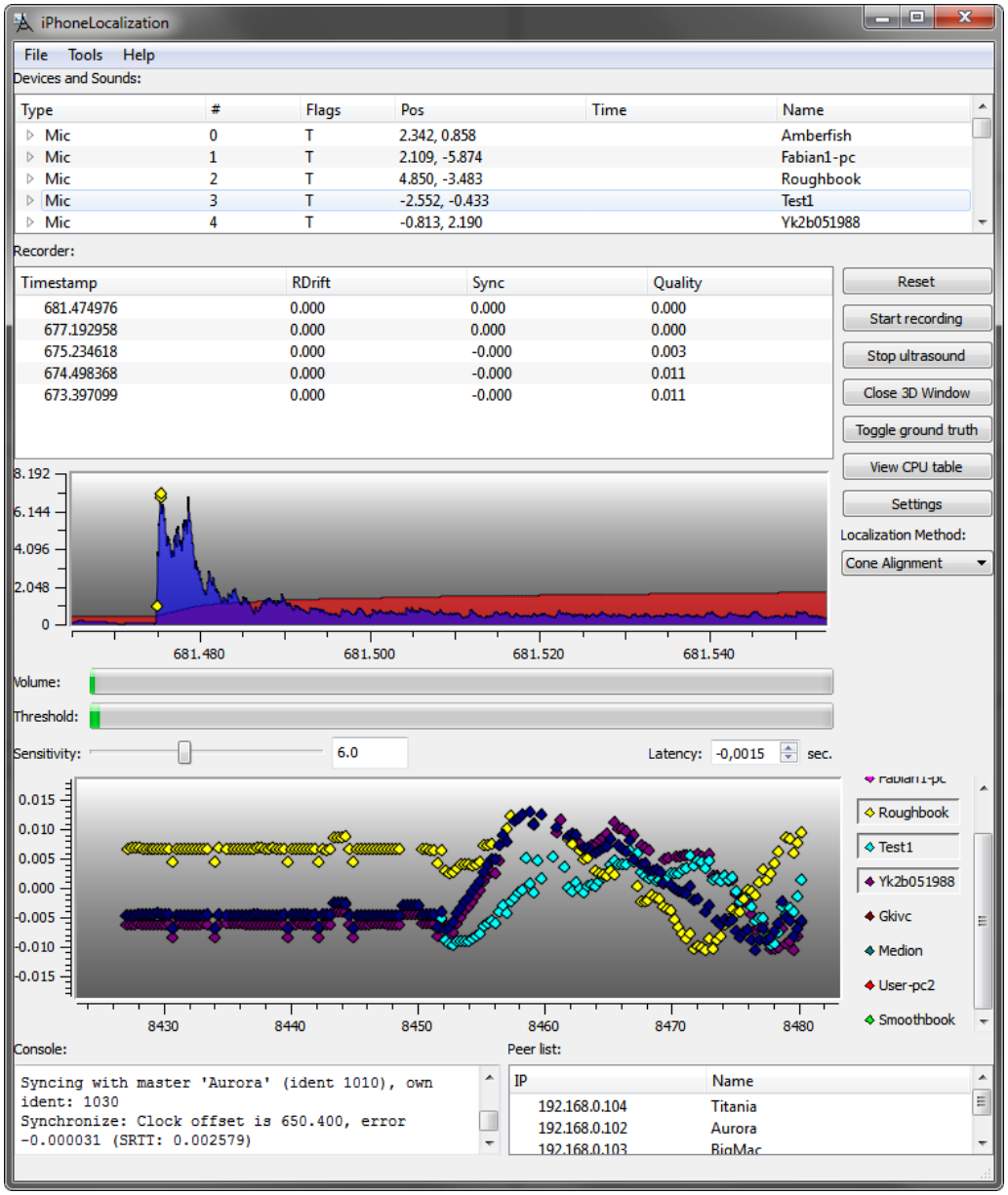


Figure 7.1: Screenshot of the GUI of the localization software. **(a)** Positions of receiver devices and sounds. **(b)** List of recorded timestamps of this device. **(c)** Visualization of the latest detected sound signal (blue), the noise threshold (red), and the amplitude-based time mark algorithms (yellow diamonds). **(d)** Time differences of signals relative to other devices in the network. **(e)** Console log and list of connected network devices.

Time synchronization. TDoA localization with unsynchronized receivers is possible in general. For example, distance estimation approaches can be used without synchronization, where the offset between receivers is estimated from the average of the time differences. However, the number of required sounds will increase to compensate for the additional variables. Another problem in unsynchronized localization is the skew of clocks, which needs to be included into the mathematical model or eliminated by precise calibration of the clocks. Therefore, the devices in our localization framework use synchronization between receivers.

As soon as multiple devices are connected in a local network, the devices detect each other by UDP broadcast messages, and a master device is negotiated, depending on the order of appearance of devices [77]. In the *Synchronization* component, the clients adjust their clocks to the master, who acts as a time reference. We implement a prototypical synchronization protocol similar to the Network Time Protocol [114], relying on the exchange of synchronization messages between nodes in the network. The clients send a series of pings at times t_{client} to the master, which answers by its current time t_{ref}, yielding tuples $(t_{\text{client}}, t_{\text{ref}}) \in \mathcal{K}$. This timestamp is corrected by a half of the *round trip time* (RTT), assuming the network transmission took the same runtime in both directions. The obtained timestamps are filtered for outliers, which result from network jitter in a wireless network. Experiments also indicated that clock skew correction is essential for the high precision event timers (HPET) [115] that we use. Although they run with accurately constant speed, we observe skew rates between HPET clocks of 300 ppm, which is very high.

For synchronization we obtain both estimated reference time $\tilde{t}_{\text{ref}}$ and clock skew $\tilde{t}_{\text{skew}}$ between client and master by linear regression of the set $\mathcal{K}$. Including RTT and skew compensation, the synchronized time of a client is

$$t_{\text{sync}} \;=\; \tilde{t}_{\text{ref}} + \tfrac{1}{2} t_{\text{rtt}} + (t_{\text{curr}} - t_{\text{last}}) \cdot \tilde{t}_{\text{skew}} \tag{7.1}$$

where t_{last} denotes the client time of the last synchronization, and t_{curr} denotes the current time of the client. With this approach we achieve a synchronization precision of better than 0.1 ms in a 802.11 b/g Wi-Fi network with round trip times of about 5 ms. See [116] for a survey of synchronization in wireless sensor networks and [117] for a suggestion of round-based synchronization.

Signal recording and detection. The *Timestamp Recorder* component records from the built-in microphone of the device or from a USB-connected receiver and searches the sound or ultrasound stream for distinctive audio events. In the clapping experiment and in the basic ultrasound localization system we use a simple edge detection technique. In this threshold-based approach, we search for the moment when the amplitude of the audio stream rises above an environment noise dependent threshold for the first time. Here, background noise is filtered implicitly. Fig. 7.3 displays an example for clapping hands. More advanced algorithms such as phase correlation and pattern-based algorithms are discussed in Chapter 8.

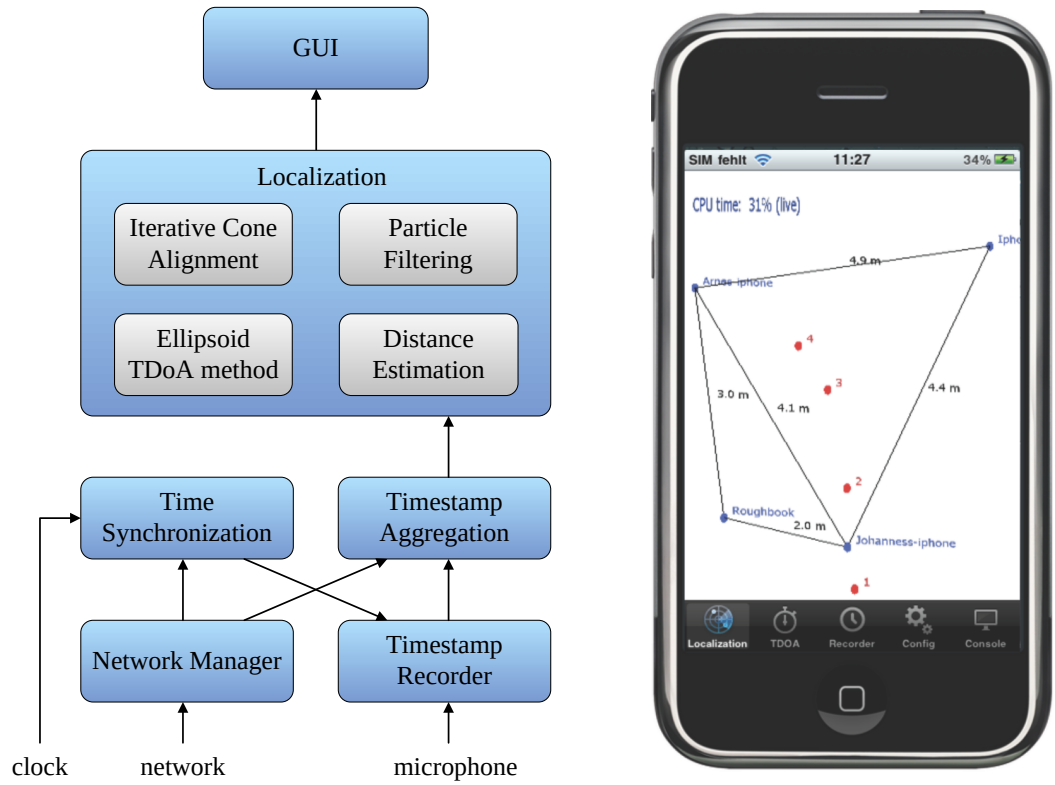


Figure 7.2: *Left*: Components of the software framework. *Right*: Our application on an Apple iPhone 3GS, displaying four receivers (blue dots) and four signal locations (red dots). [51]

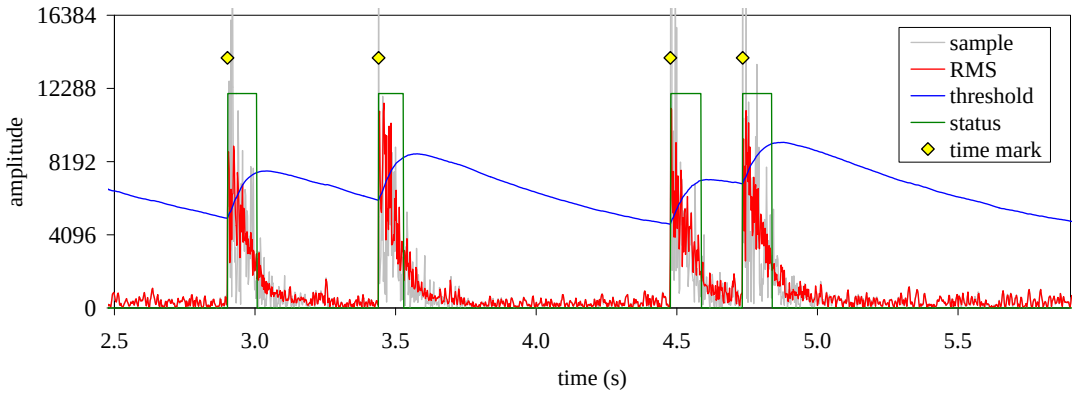


Figure 7.3: Environment noise dependent threshold detection of clapping hands. The moment when the signal rises above the threshold is chosen as the timestamp. [51]

In comparison of various amplitude-based detection techniques, threshold comparison turned out to be the most robust approach with only minor drawbacks in precision. Maximum search, either directly or by the derivative (edge detection) showed to be slightly more precise but appeared to be ambiguous with fatal results in cases when different devices chose different maxima of a sound signal. In case of a clicking audio signals with a steep initial edge we calculate time marks with a precision of 0.1 ms, the spatial equivalent is 3 cm. However, when signals are smooth, like the human voice, the time point of signal detection cannot be clearly determined. Snapping one's fingers or clapping one's hands is still adequate.

The *Timestamp Aggregation* component collects the local time marks from the Timestamp Recorder and the time marks from remote devices in the network, combining them to a logical sound event of the real world, for example a finger snap. It is essential to aggregate the time marks to a sound event in the correct way, cf. Section 2.3. We choose temporal proximity of timestamps as the criterion. We assume that the time between sound events is greater than the maximum TDoA in a sound event, such that the aggregation is unique. Otherwise, we discard ambiguous events to prevent false assembly.

Although the Cone Alignment algorithm and the particle filter algorithm are feasible in three-dimensional space, as described in the respective chapters, the following experiments will be conducted in a planar setting, a restriction for the sake of simplicity of the experimental setup. In three dimensions the number of receivers would be higher, due to the determination of the equation system and the limited aperture of the microphones. Also, creating reference positions and sound signals is harder. Based on the results of our simulations we expect similar results if we manage to record TDoA measurements at the same precision as in two dimensions.

7.2 Acoustic Self-Localization

Our first real-world test took place in an outdoor setting on a green area on our campus, where we evaluated the Cone Alignment algorithm [5]. We arranged a scenario of four laptops and four Apple iPhones in a roughly elliptic formation of the dimensions $30\,\mathrm{m} \times 30\,\mathrm{m}$. The devices were connected by a Wi-Fi access point. Alternatively, one of the laptops could have been used as Wi-Fi hotspot, making the setup independent of external infrastructures. The goal was to locate the positions of all laptops and iPhones just from natural sounds from the environment and the positions of the sounds.

With the network connection the software running on each device could communicate with the other instances and provide synchronization among all devices. With their built-in microphones the devices recorded any incoming sound event. The time points of the sounds were calculated by analysis of the audio stream. Sharp sound events like clapping, or finger snapping are detected by comparing the audio signal to an environment noise dependent threshold, as described previously.

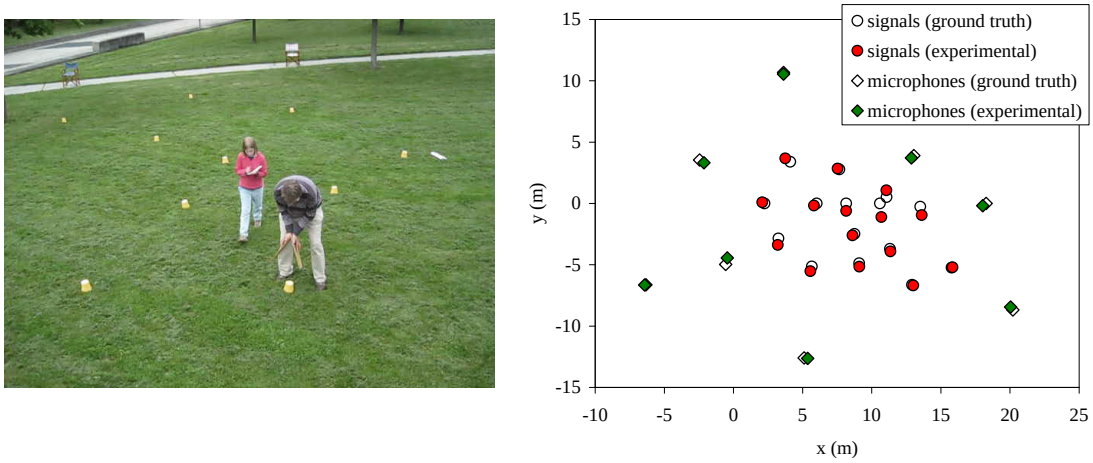


Figure 7.4: Four iPhones and four laptops are being located from 15 unknown sound signals. The experimenter claps two wooden planks while the assistant notes down the signal positions. Two of the iPhones are on the wooden chairs in the background. The average location error of the receivers is 0.28 m ($\sigma = 0.14$ m). [5]

We noted the positions of all laptops and smart phones by measuring the distances to two anchor points, which were chosen as reference points for a Cartesian coordinate system. Then we calculated the x/y-coordinates of the devices up to a precision of 10 cm using trilateration. Now, an assistant walked in the experiment field creating noises by clapping two wooden bars, which made a noise with sharp characteristics, see Fig. 7.4. He was allowed to choose the locations of the sounds arbitrarily, but to move in between, such that the signals were well distributed.

The positions of the sound signals were marked with plastic cones on the floor. Since the assistant was free to move we could chart the sound signals only to a precision of 30 cm. Using the synchronous time base the software on every receiver calculated a synchronized timestamp of every sound event. A filter removed signals which had been missed by more than one computer, which occurred in some cases as a result of environmental noise. After filtering we identified a total of 15 sound signals at 15 noted positions.

Given these timestamps as input, the Cone Alignment algorithm computed the relative locations of the receivers and sounds. The experimental data and the reference positions were aligned by a congruent transformation by using singular value decomposition (SVD), minimizing the distances between experimental and ground truth positions. Recall, as pointed out in Section 1 and 4.2, that the algorithm does not use anchor points and provides only relative localization.

The average location error of the microphones after alignment was 0.28 m with a standard deviation of 0.14 m. The average error of the sounds was measured to 0.39 m with a standard deviation of 0.28 m, see Fig. 7.4. The larger error of the sounds might have been influenced by imprecise noise generation above the plastic cones and the

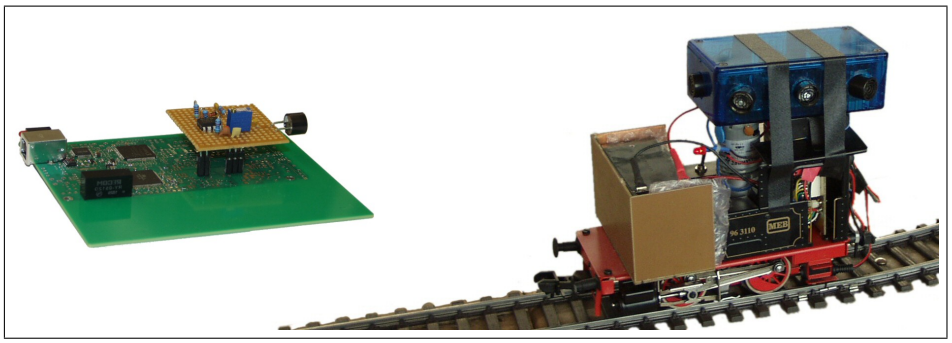


Figure 7.5: *Left*: Receiver board with ultrasound capsule and USB connector. *Right*: The first version of the ultrasound beacon (blue box) with eight ultrasound capsules facing in all directions, attached to a model train. [5]

measurement of their positions. We did not encounter the local minima issue in this strictly over-determined scenario.

In these audio experiments we saw that in a controlled environment we could obtain very precise timing of the audio events – and audible sounds are available for free in many situations. By replacing the TDoA calculation by the more robust cross correlation, this would also work for human voices, see Section 8.1. For now, we require that no additional clicking noise is created during the experiments, which would impose a risk of association ambiguity. In the next section we present an ultrasound tracking system that is less vulnerable to environmental disturbance.

7.3 The Ultrasound Localization System

A multitude of tracking systems and positioning approaches are available for indoor and outdoor localization. Many of the commercially available systems are optical systems and very expensive. A number of localization approaches is based on RSSI evaluation of Wi-Fi hotspots, yet to date their precision is not convincing. Approaches using TDoA multilateration usually require receivers with calibrated locations, for which the positions have to be tediously measured by hand. This can be disadvantageous for industrial applications, as these have to be easy to use.

We propose a novel tracking system for moving targets using our calibration-free TDoA algorithms [5, 54]. It can quickly be set up, without the need to calibrate the positions of receivers. Of course, when the positions of at least three of the devices are given, the obtained relative coordinates can be converted to absolute coordinates.

We understand that audible sounds are not appropriate here, they would be annoying. Therefore we use ultrasound at a frequency of 40 kHz. In recordings with a high-frequency microphone we saw that only few acoustic sources emit signals up to the ultrasound range, which reduces the vulnerability of the tracking system to

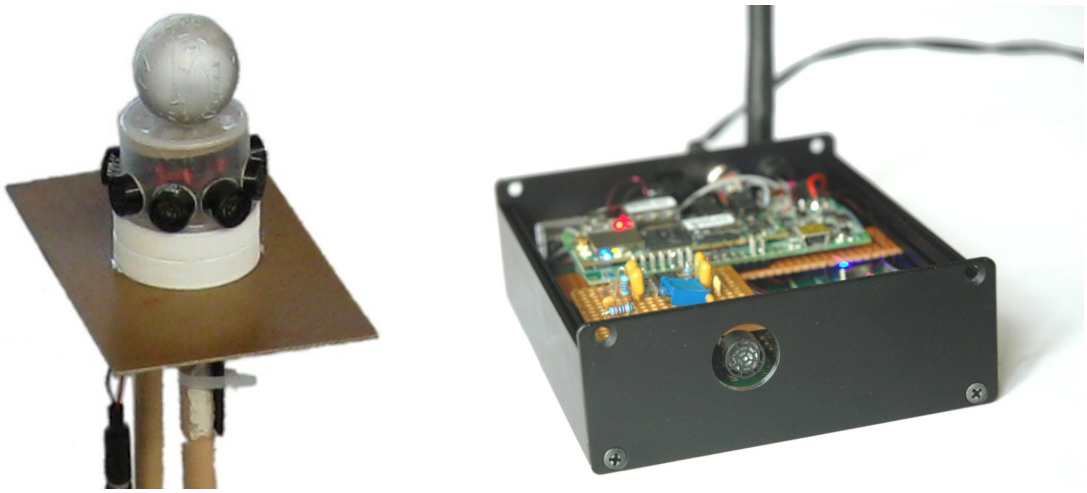


Figure 7.6: *Left*: The second version of the ultrasound beacon on the flagpole of a RC model car. A reflective marker for the optical motion capturing system (MoCap) is attached on top. *Right*: One of our integrated ultrasound receiver units with a Gumstix Overo computer and wireless communication. [109]

environmental influences. Typical environmental noises such as the human voice, air conditioning, noises of the model train and car, or even a bypassing aircraft, seem to disappear above 30 kHz. Our ultrasound tracking system that we have developed in cooperation with the Laboratory for Electrical Instrumentation, IMTEK, University of Freiburg, consists of a mobile ultrasound emitter beacon and receivers that record and process the signals from the beacon. It has been assembled from off-the-shelf components and underprices most commercially available tracking systems.

The ultrasound beacon emits short ultrasound pulses of 40 kHz at periodic intervals. With eight ultrasound capsules facing in all directions it creates an approximately isotropic signal (Fig. 7.6). It can be carried by a person or it may be attached to a moving unit, for example a model car or a model aircraft. As it is battery powered it can be used independent from line voltage, therefore a moving vehicle can be tracked in real-time. The interval of the pulses can be freely chosen. It should be so large that signals arriving at the receivers can be distinguished. For example, in an experimental setup with the dimensions of 20 m the interval should be larger than 50 ms. We send short peaks of 1 ms with no information encoded in the signal. In our experiments we use an interval of 300 ms, a compromise between update rate and network load.

We record the ultrasound signal by USB-connected ultrasound receiver devices that we have developed, see Fig. 7.5. Each receiver device consists of an ultrasound capsule attached to a custom controller board amplifying and digitizing the signal. The controller board is connected to a computer which detects the ultrasound bursts of the beacon and calculates their points in time. Furthermore, for flexible integration into arbitrary environments we have built integrated receiver units that do not require an

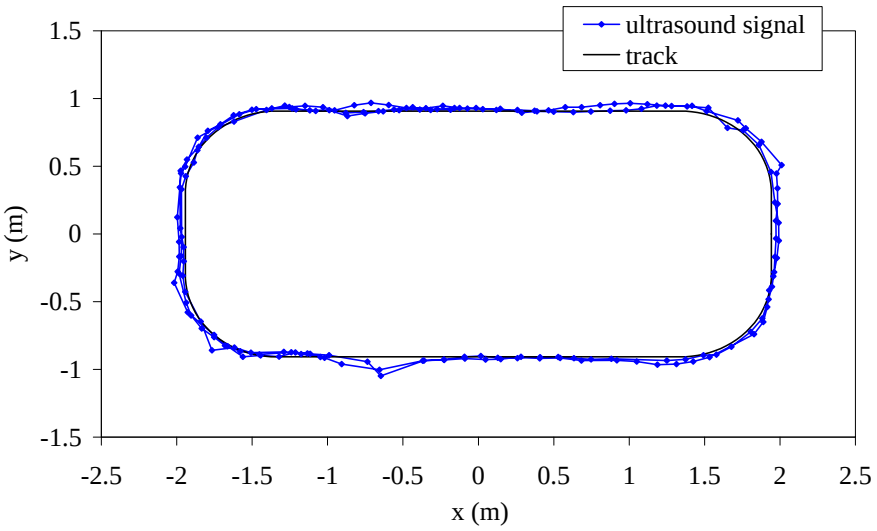


Figure 7.7: Model train experiment with five receivers (not in the picture) around an oval track of the dimensions 1.8 m × 3.9 m. The average error of the estimated receiver positions is 44.5 cm (σ = 7.7 cm). For the estimated signal beacon position we observe a RMS track error of only 2.5 cm. [5]

external computer. These are based on Gumstix Overo Computer-on-Modules (COMs) with an ARM Cortex-A8 processor, see Fig. 7.6. With their low power consumption and the small dimensions of 12 cm × 11 cm × 4 cm they can be battery powered and quickly installed into the environment. For both types of receivers we use a Wi-Fi connection for data communication and time synchronization.

7.3.1 Model Train

In an experiment we track a moving model train [5, 54], where the ultrasound beacon is attached to its roof (Fig. 7.5). On a simple trajectory, a rectangle of the dimensions 1.8 m × 3.9 m and a curve radius of 0.6 m, the train moves with a velocity of about 0.5 m/s (Fig. 7.7).

Five receivers were placed evenly around the track, at a distance of 4 – 7 m. As we conduct an experiment in the plane, we placed the receivers at the same height as the beacon. The ultrasound microphones were roughly oriented towards the oval track and connected to adjacent laptops. With our software running they find each other in a Wi-Fi network and synchronize their clocks. Using a measuring tape we measured the positions of the ultrasound capsules of the receivers up to a precision of 3 cm. For the dimensions of the train track we describe the geometrical shape of the track.

For the tracking experiment we assume that the signals are spatially coherent, such that the moving beacon has limited velocity. In this way, we filter implausible timestamps. The phenomenon of multipath propagation, i.e. echoes from walls, was handled

Figure 7.8: *Left*: RC model car with ultrasound beacon on top and markers for the optical reference system (MoCap). *Right*: Experiment setting with the RC model car and eleven ultrasound receivers.

by issuing a dead time of the timestamp detector after every received signal. In rare cases a detector did not receive a signal in the direct path, but only a delayed echo of the same signal, which is then detected as the first signal. These false signals are detected and discarded in the way described.

After approximately three rounds the Cone Alignment algorithm got the TDoA data as the only input. We calculate both the unknown ultrasound receiver positions and the trajectory of the train on the track. The estimated receiver positions were fit to the measured coordinates using SVD. Comparing the data, we find it well matching the ground truth, but we observe some overestimation of the receivers, which exhibit an average deviation from the true positions of 44.5 cm ($\sigma = 7.7$ cm). The overestimation is weakly pronounced for the trajectory of the model train. We observe only a small overestimation resulting in a root mean square (RMS) track error of 2.5 cm.

Sippel et al. [118, 119] use a similar setup of an oval trajectory with a model train. In the indoor radar experiments they obtain an overall standard deviation of 3.6 cm, with notable overestimation of the real track of about 20 cm and with large outliers in case of disturbances. Using a laser scanner precise results are obtained, however the authors describe that the scanner is susceptible to losing track of the train. Both techniques require calibration and they are prone to influences of the environment. In contrast, our ultrasound system is not affected by obstacles in the environment, as long as a line-of-sight to the beacon exists, and the financial effort should be way below the costs of the radar and the laser system.

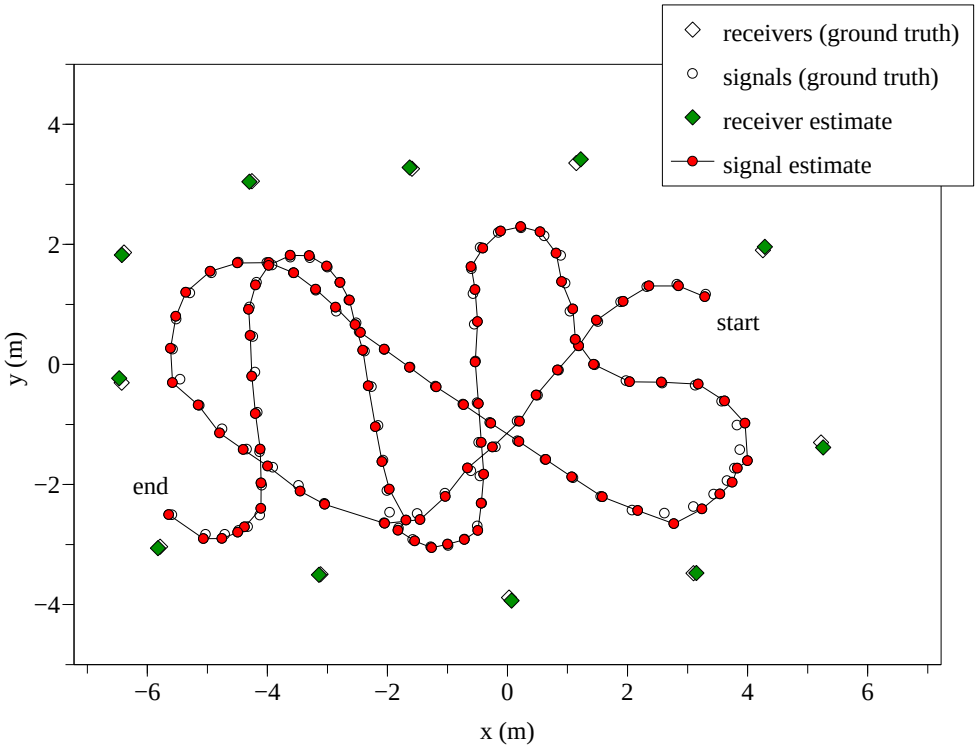


Figure 7.9: Evaluation of the trajectory of the RC car in the time interval 70 – 102 seconds with Cone Alignment. On its path, the model car generates ultrasound pulses every 0.3 seconds, which are received by up to eleven receivers. [5]

7.3.2 RC Model Car

For a second experiment [5] we have redesigned the ultrasound beacon, as the quality of the results in the last experiment may have been affected by the rectangular shape of the box (Fig. 7.5, right). Signals from the outer edge (i.e. the shorter edge) of the box will do a "head start", compared to the long edge, resulting in a timing error of the receivers. With the new cylindrical casing the signal is now emitted from the same radius from the center to all directions in the plane (Fig. 7.6, left).

Our present ultrasound system for this experiment consists of five integrated receiver units and additional six conventional ultrasound receiver devices connected to laptop computers, see Fig. 7.8. We arranged the receivers in an area of 12 m × 8 m in a roughly oval shaped formation on the same level with the ultrasound microphone directed towards the center of the tracking area. As pointed out previously, our arrangement in two dimensions simplifies the experimental setup immensely.

We attached the redesigned ultrasound beacon to a flagpole on top of a 1:12 scale RC model car. For continuous tracking we programmed the beacon to emit an ultrasound burst in a fixed interval of 300 ms. Although our tracking system does not

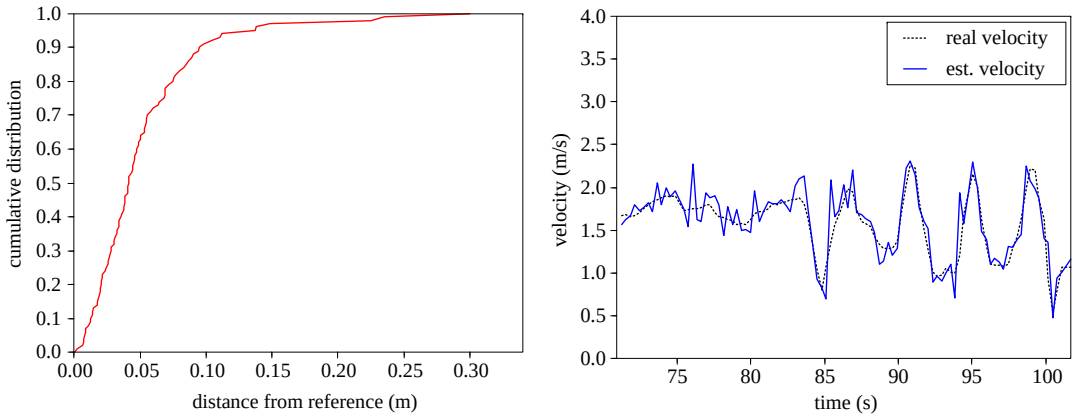


Figure 7.10: Evaluation of the time interval 70 – 102 seconds, see the top-down view in Fig. 7.9. *Left*: Cumulative distribution function (CDF) of the errors after evaluation with Cone Alignment. The mean error of the ultrasound beacon is 5.4 cm ($\sigma = 4.7$ cm), compared to the references. *Right*: Estimated and reference velocity of the signal beacon. [5]

rely on fixed intervals, the 300 ms interval turned out to allow for a precise and all-over tracking of the moving beacon while the receivers can still uniquely identify the individual bursts.

To evaluate the precision and robustness of our ultrasound tracking system we used a MotionAnalysis optical motion capture system (MoCap) with nine Raptor-E cameras. The MoCap cameras were installed at the ceiling of the experimental area observing the full area inside the receiver array. The positions of the receiver devices were obtained by attaching a reflective marker to each device exactly above the ultrasound microphone. We attached another marker to the top of the ultrasound beacon, so that the trajectory of the sender could also be determined at a tracking frequency of 100 Hz. In this way, we generate position references with a precision of millimeters. In an experiment, where the beacon is carried by a person, he wore a hat with reflective markers on its top to obtain the ground-truth positions of the person instead of the position of the beacon.

This time, our focus was to create a large-scale experiment with signals received by only a subset of the receivers and allow an arbitrary trajectory of the signal beacon. The model car moved on the field between the receivers for several minutes on a random trajectory, generating a few hundreds of signals. Because of the limited range and aperture of the ultrasound capsules, and because the model car was guided through the whole experiment field, even outside the perimeter of the eleven receivers, the signals were not necessarily received by all of the receivers. We discarded the signals that were received by less than three receivers, as the positions of these signals cannot be calculated reliably, because the equation system is under-determined.

After the experiment a subset of the signals with the model car driving on a path throughout the experiment field was chosen for evaluation with the Cone Alignment

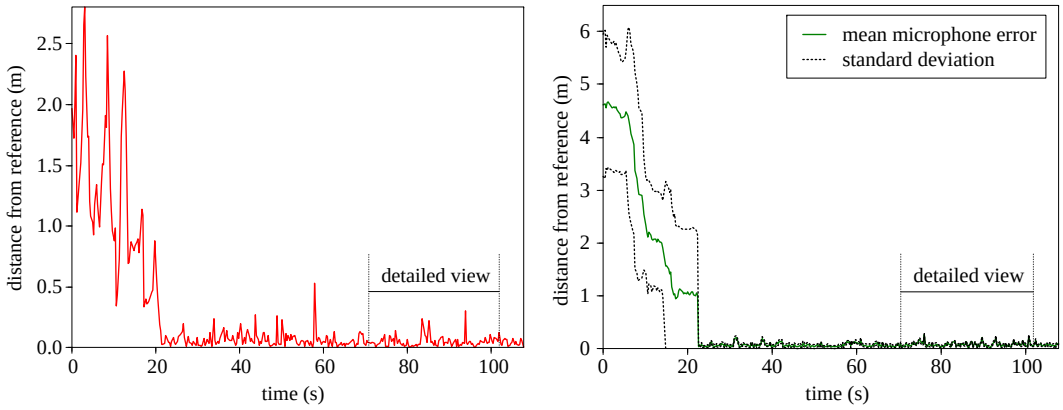


Figure 7.11: Iterative adding of the most recent signal, up to 80 signals. In the beginning neither the receivers nor the signal beacon positions are known. After 22 seconds a good estimation of the receivers is found, so the model car can be tracked. *Left*: Error of the ultrasound beacon. *Right*: Mean error and standard deviation of the receiver positions. The marker *detailed view* denotes the data segment in Fig. 7.9. [5]

algorithm. We selected a section of the model car path with constant velocity of 1.8 m/s in the beginning, followed by variations of the velocity between 0.7 m/s and 2.3 m/s, which is a data set that we also use in [109]. The subset consisted of 101 signals, where the number of received timestamps per signal ranged from three to eleven. The data was evaluated by the algorithm and the resulting position estimates for the ultrasound trajectory and for the receivers were mapped by SVD to the references, see Fig. 7.9. The mean error of the moving signal beacon of this experiment was 5.4 cm ($\sigma = 4.7$ cm), see the distribution of errors in Fig 7.10, the mean error of the stationary microphones was 8.6 cm ($\sigma = 5.0$ cm). We did not observe the overestimation that occurred the previous experiment, most probably because of the redesigned beacon.

In an online evaluation of the experiment with recorded data, we iteratively added signals, up to a maximum of 80 signals, corresponding to about 24 seconds of time. Using the Cone Alignment algorithm we evaluate the positions of the most recent signal and of all receivers, given only the TDoA information. For this, we have created a modification of Cone Alignment, where particles are successively added into the running mass-spring simulation, while the abort condition of the algorithm is disabled. In this setting we, naturally, face an initialization problem of the receivers. No information is available in the beginning, neither the positions of the beacon, nor of the receivers. When the car starts to move, TDoA information is rare and not well distributed, most probably leading to defective position assumptions. As soon as the ultrasound signals arrive from different origins, the algorithm can recover, and the positions of signals and receivers can be calculated[1].

[1]Demonstration video of Cone Alignment: `http://www.youtube.com/watch?v=nKyYJy20CTc`

In this experiment we evaluated a total of 350 signals from an interval of 110 seconds. 22 seconds after the beginning, a plausible estimation of the receivers was found (Fig. 7.11), enabling precise tracking of the ultrasound beacon with an error of centimeters. We also saw in this experiment that Cone Alignment can run in real-time on a standard PC with moderate CPU load when new ultrasound signals from 11 receivers arrive every 300 ms. In some rare cases we observed that the online algorithm got stuck in a local minimum, even when TDoA data was collected from throughout the experiment field. Re-initialization of the algorithm, as described in Section 4.3.4, may probably solve this problem.

7.3.3 Evaluation of the Particle Filter

We have also verified the capability of our particle filtering approach using the described indoor test setting [109]. In the two-dimensional arrangement of stationary receivers a moving signal beacon was to be tracked, similar to the testbed in the Cone Alignment experiment in the previous section. As in the experimental evaluation of the optimization algorithm, the beacon emits short ultrasound bursts, which can be detected by the receivers.

For the first experiment, which we denote as the *model car experiment*, we installed the ultrasound beacon to an R/C model car on a pole, such that it resides at the same height as the receivers (see Fig. 7.8). In several runs of three to ten minutes we navigated the model car through the experimental area[2], at varying velocities at an average of 1.5 m/s and up to 4 m/s. In *Exp. 1* the path of the model car included sharp turns, sudden stops, and rapid starts. In *Exp. 2* we extended the movements by leaving the perimeter of the ultrasound microphones and the optical reference and re-entering at another location to force the particle filter to recover at totally different positions. Naturally, the signal is not received by all receivers, as the car navigates into a corner of the area, or even outside.

In the *pedestrian experiment* (*Exp. 3*) a person carried the sender in his hand. He walked a random path of a few minutes through the area, with sharp turns and stops. At all times, he held the beacon at the same height as the receivers to satisfy the planar constraint. In this experiment the direct path of the ultrasound signal to the receivers was obstructed by the body of the person, blocking some of the signals. Most notably, the ultrasound beacon and the hat with the reflective markers were not vertically aligned, such that their horizontal positions deviate up to half a meter. Consequently, the reference positions are less reliable in this experiment.

We have split the experiments into several data sets to evaluate the behavior of the algorithm from different aspects. We obtain four data sets from the model car experiment, namely *Exp. 1a*, *Exp. 1b*, *Exp. 2a*, and *Exp. 2b*. From the pedestrian experiment we obtain the data sets *Exp. 3a* and *Exp. 3b*.

[2]Demonstration video of the particle filter: `http://www.youtube.com/watch?v=V85wejcYyXs`

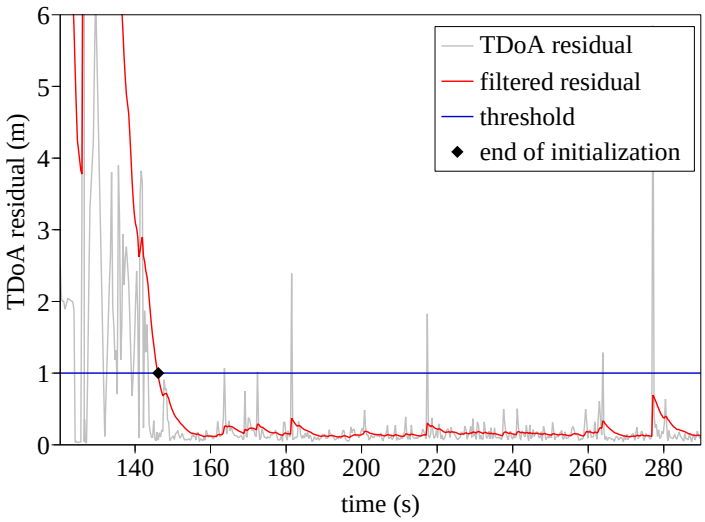


Figure 7.12: TDOA residual of *Exp. 1a*. After 26 seconds the filtered TDOA residual falls below the threshold of $\epsilon_{\text{init}} = 1$. Then, initialization with the optimization algorithm ends. Recall that the TDOA residual is not to be confused with position error of the beacon compared to the MoCap reference. [109]

Initialization phase. In the beginning, no positions of neither the beacon nor the receivers are known. The particle filter algorithm is initialized with a uniform distribution. As the number of measurements increases, the iterative TDoA algorithm described in Section 6.2.2 begins to compute configurations of likely receiver positions and the beacon using a random subset of the available signals. We choose to use the algorithm with a fixed number of 10 signals, which is a compromise between computational load and a good chance of converging.

We have evaluated the initialization scheme described in Section 6.2.2 with six runs for each of the six data sets. During the initialization period the beacon moved on a curvy trajectory in the experimental area, which we require to initialize. Our localization algorithm monitors and low-pass filters the internal TDOA residual of the particle filter estimate as shown in Fig. 7.12. It automatically finishes the initialization phase as soon as the filtered TDOA residual drops below the threshold $\epsilon_{\text{init}} = 1$. As shown in Fig. 7.13, the end of initialization of the filter is achieved in all of the tested scenarios after a maximum initialization time of 35 seconds. When the initialization was finished, the position error of the beacon was already below 70 cm in all cases, which indicates a proper initialization near to the true state.

Tracking phase. After the initialization phase is finished, we deactivate the iterative algorithm and switch to position tracking in the particle filter. Additionally, we enable kernel smoothing on the receiver estimates to prevent from information loss during the resampling step, cf. Section 6.2.3. Fig. 6.2 illustrates the beacon and the receiver estimate in the particle filter.

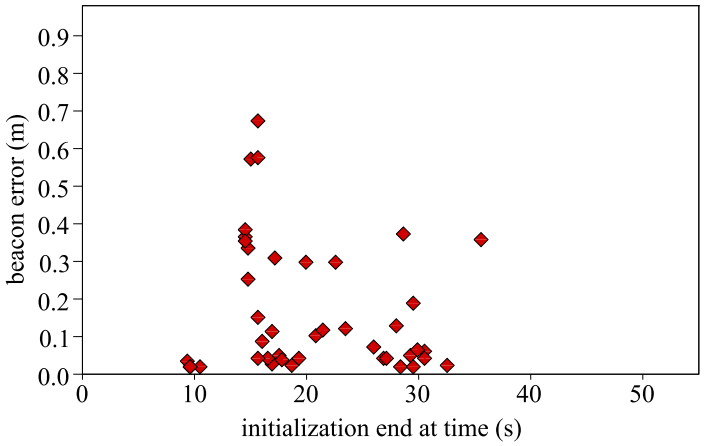


Figure 7.13: The end time of the initialization, which is triggered by the TDOA residual falling below $\epsilon_{\text{init}} = 1$, and the corresponding localization error for six particle filter runs on each of the six data sets. In all runs the initialization time does not exceed 35 seconds. [109]

During the tracking phase the estimated beacon positions are very consistent to the indicated references. We observe an effect of the constant velocity motion model, which assumes a straight path of the beacon and consequently pushes the particles slightly to the outside curve. In an extract of 30 seconds of the first data set *Exp. 1a* we find that 95 % of all positions deviate less than 10 cm from the reference positions, at a mean of 4.8 cm (Fig. 7.14).

In *Exp. 2* we navigated the model car outside the range of the receivers, re-entering at another position after three to five seconds. As shown in Fig. 7.15, the estimate of the beacon diverges, as soon as not enough TDoA measurement data is available. However, a few steps after re-entering the range of the receivers new measurement data is received and the particle filter quickly recovers to the correct position.

7.3.4 Robustness to Outliers and Comparison of Algorithms

A common problem for TDoA localization in indoor environments are extreme measurement errors caused by echoes from nearby walls, noise from the environment, and wrong assignment of signals. These typical errors appear as single spikes in the TDoA graph, with magnitudes of 10 ms to more than 100 ms. We verify the robustness of our particle filtering approach by a systematical simulation of such spikes.

We compared the particle filtering algorithm to the iterative *Cone Alignment* algorithm [5, 109]. Recall this iterative optimization approach, which minimizes the constraints between signals and receivers according to Eq. (2.1) in terms of energy minimization. For every new signal we call the algorithm with a set of 80 previous signals, which is described in the previous section. Although defective measurements are carried for a longer time, the high number of measurements taken into account increases the precision and the robustness of the algorithm.

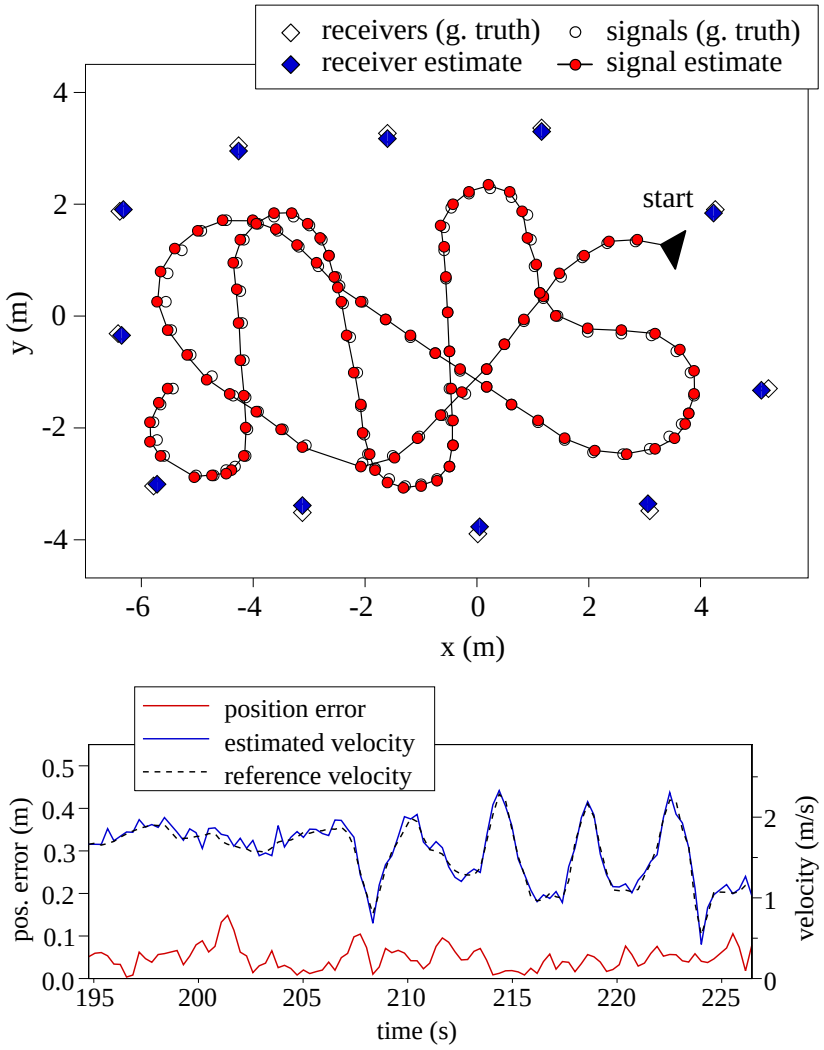


Figure 7.14: An extract of 30 seconds of Exp. 1a. *Top*: In sharp corners the particles are slightly drawn to the outside, which is an effect of the constant velocity motion model. *Bottom*: Position error and velocity of the beacon. The mean position error is 4.8 cm ($\sigma = 2.8$ cm). [109]

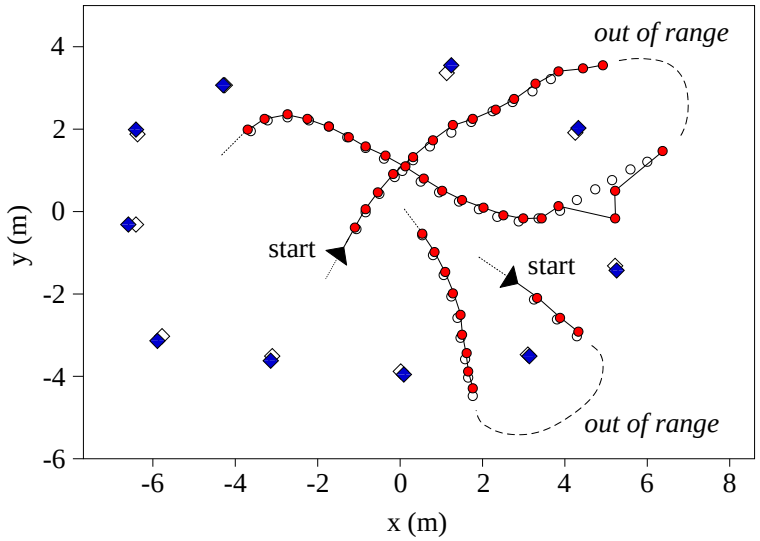


Figure 7.15: Data set *Exp. 2a*: The beacon estimate when the signal is lost. Only a few steps after new TDoA information is available the particle filter estimate reliably converges to the true state. [109]

For the evaluation of the robustness of the algorithms we systematically manipulated our measurement data by randomly generating outliers every 50 to 70 measurements. Their magnitudes were sampled from a Gaussian distribution with standard deviations of 10, 20, 50, 100, 200 and 500 milliseconds. For all of the six data sets we compare the deviation of the estimated positions from the reference positions for the different standard deviations.

Fig. 7.16 shows the error distribution of all algorithms with and without outliers. In all data sets without outliers the particle filter with kernel smoothing performs slightly better than the iterative approach. In contrast, in the presence of outliers, even the particle filter without kernel smoothing outperforms the Cone Alignment algorithm which is susceptible to these errors. Especially with kernel smoothing the particle filter performs robustly as soon as the estimates of the receiver positions have converged. The position errors in case of different magnitudes of outliers are illustrated in Fig. 7.17. As the estimates of the signal and receiver positions are related through the TDoA constraints between them, a wrongly estimated beacon position, caused by an outlier in the TDoA data, can also corrupt the receiver position estimates. In this way, the estimate of iterative algorithms like the Cone Alignment can literally "explode", resulting in a complete loss of the estimate.

Using the probabilistic state estimation approach we did not observe initialization failures of the algorithm in these data sets. By repeated initialization attempts with different subsets of measurements the particle filter can eventually find a good estimate of the receivers and thereby overcome the local minima issue that occurred in the online version of the Cone Alignment algorithm.

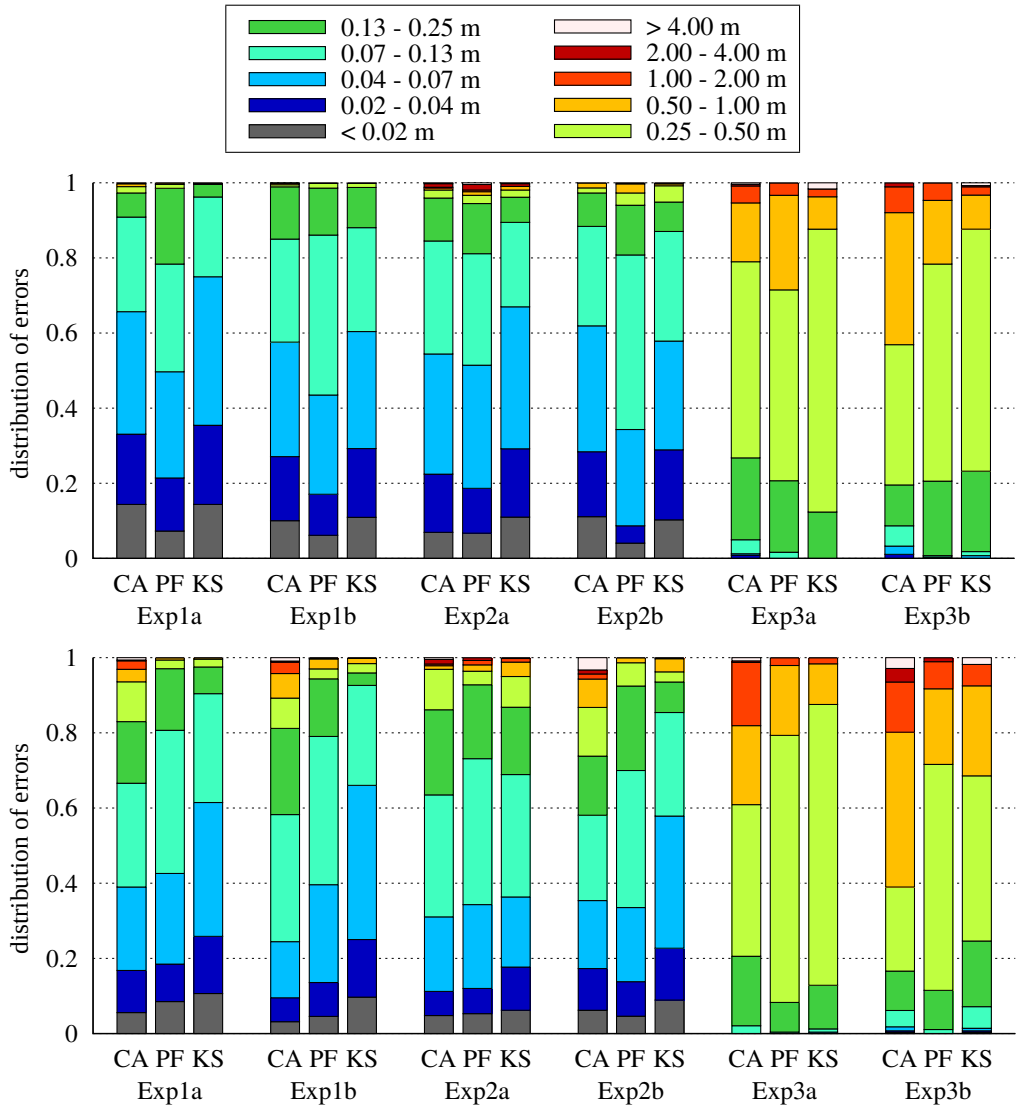
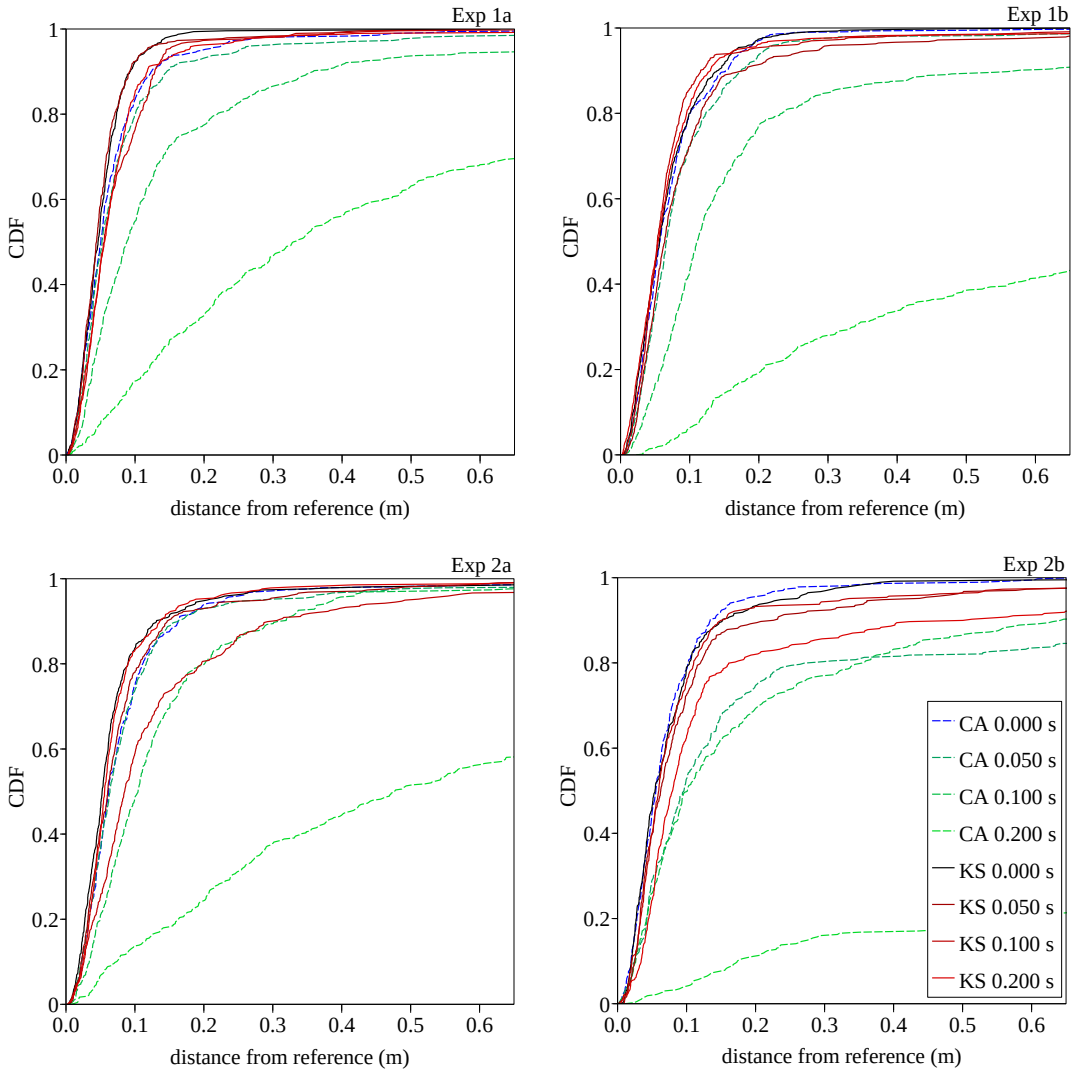
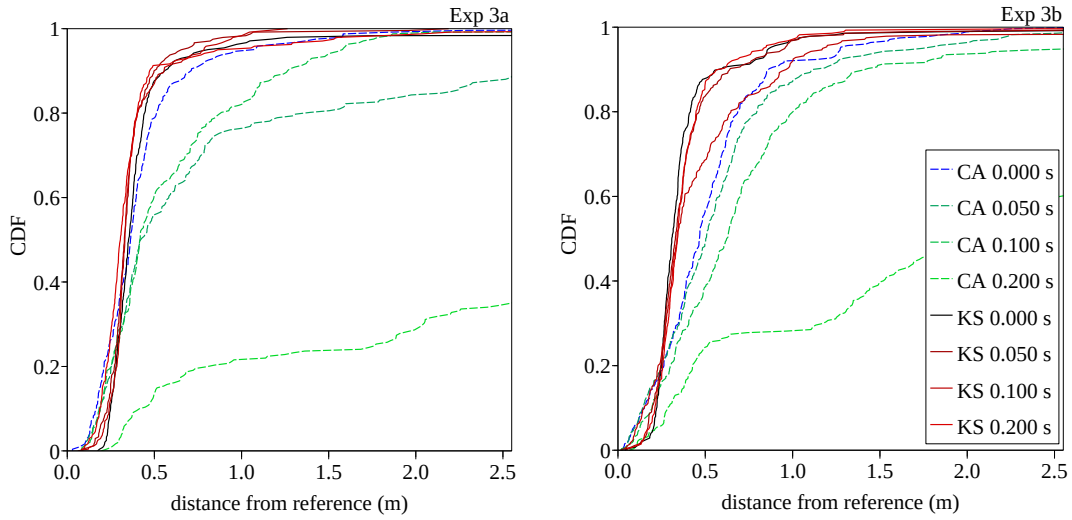


Figure 7.16: The error distribution of the two model car experiments *Exp. 1* and *Exp. 2* and of the pedestrian experiment *Exp. 3*, comparing Cone Alignment (CA), particle filtering without (PF) and with kernel smoothing (KS). Read the plot as "Higher segments are better". *Top*: No outliers. *Bottom*: Simulated outliers with 100 ms standard deviation. [109]



(a) While in the RC car experiments *Exp. 1* and *Exp. 2* the particle filter (KS) is robust even to extreme outliers of 200 ms, the Cone Alignment algorithm (CA) exhibits an increasing proportion of bad estimates.



(b) In the pedestrian experiments *Exp. 3* the absolute error is higher, as the reference positions of the hat with MoCap markers are less reliable, but the proportions of errors are similar as in the RC car experiments.

Figure 7.17: The cumulative distribution of the position errors of the beacon for Cone Alignment (CA) and for the particle filter with kernel smoothing (KS) for varying magnitudes of measurement outliers. [109]

7.4 Related Work

We discuss a short survey of popular acoustic localization systems. In the decentralized *Cricket* system [1, 120] distributed static and mobile nodes are equipped with an ultrasound beacons and a "listener". By transmission of ultrasound and RF signals the locations of new static nodes and of mobile nodes can be inferred, relative to the previously configured nodes. In the *3DLocus* system [121, 122, 123] stationary bi-directional ultrasound units communicate with a mobile unit. Stationary units are synchronized with the mobile unit by wireless or wired network, and the ultrasound signal is decoded by cross correlation, yielding high precision ranging up to 9 mm. In [15] is the ultrasound ranging experiment *AHLoS* presented, where in an iterative approach the positions of ultrasound nodes are determined from already given nodes. In the *ActiveBat* [124, 125] system are synchronized mobile ultrasound badges located by receivers in the ceiling.

BeepBeep [2] is a ranging system for smart phones and PDA. At least two devices estimate the distance between each other by exchange of acoustic sounds of frequencies up to 20 kHz by using the built-in speakers and microphones of the devices. By listening not only to the other unit's signal, but also to the own emitted sound, the hardware delay of the devices can be eliminated. In *Whistle* [126] a similar approach is used for hyperbolic localization. Mobile phones are located by sending a sound signal over an

intermediate signal repeater, which is also a mobile phone, eliminating the need for synchronization. While the idea is elegant, both systems suffer from increased risk of missing a measurement, due to the high number of intermediate waypoints of signals. In *WALRUS* [127] are pulses of 21 kHz of stationary beacons received by mobile PDAs and laptops. As they are synchronized to the beacons by Wi-Fi, the ranges between beacons and mobile unit can be calculated.

Some auto-calibration systems and experiments use sounds from the environment to locate network devices with built-in microphones, as in Section 7.2, for example the calibration-free TDoA approaches in [41, 46, 47], or the approaches for tracking a moving speaker by cross correlation [37]. In *PANDAA* [48] are sounds events detected by their amplitude, and the distance between a pair of receivers is calculated by a method similar to the maximum estimator from Section 3.1, yet with explicit consideration of Gaussian measurement errors. See [128] for a survey on further indoor and outdoor location systems.

7.5 Summary

For evaluation of our algorithms, and to study the characteristics and challenges of TDoA localization, we created a localization framework where nodes in a computer network record sound or ultrasound signals, and exchange the timestamps to calculate the position of the nodes. We implemented this framework on laptops, smart phones, and embedded computers. We have demonstrated that the framework is capable to calculate the positions of laptops and iPhones with our algorithms by clapping sounds from the environment, of which the time differences of arrival are calculated.

Using our framework, we have created the *calibration-free ultrasound localization system*, consisting of a small and simplistic ultrasound emitter and specific ultrasound receivers. With the system we can localize the position of the beacon relative to the receivers without the need to measure the positions of the receivers a priori. The receiver positions are calibrated automatically during operation of the localization system by our calibration-free TDoA algorithms. In our system, the simplistic mobile ultrasound beacon does not require additional communication, nor synchronization with the receivers.

We have conducted extensive experiments with the ultrasound tracking system, where we randomly navigated the signal beacon mounted on a R/C model car through an indoor environment. In these experiments we demonstrated the accuracy and reliability of the probabilistic state estimation algorithm. We performed a detailed analysis of the algorithm and compared it to iterative Cone Alignment in terms of robustness. In comparison to tracking a moving beacon by iterative optimization, our probabilistic approach is superior in terms of localization error, especially in case of measurement outliers. We have demonstrated that the filter successfully initialized in less than 35 seconds in all experiments. In the subsequent tracking phase the algorithm

localized the beacon with an average position error of less than 5 cm and was able to reliably recover the beacon position in case of signal loss. In experiments with random measurement outliers, which may occur from environmental influences such as noise and echoes, the particle filter was marginally affected, even in case of very high error magnitudes.

8 Advanced TDoA Calculation

Reliable calculation of time differences is essential for TDoA localization. For the ultrasound system, the simple amplitude-based edge detection of signals that we introduced in Section 7.1 turned out to be adequate, as only few sources of disturbance emit signals up to the ultrasound frequency range of 40 kHz. However, for the acoustic self-localization by environmental sound, and for smart phone localization in the range of 20 kHz as described in [113], the simple edge detection technique is prone to additional recording of false sound events and to ambiguities of the edge position in case of smooth sound, for instance the human voice. Furthermore, edge detection effectively limits the range of transmission, as the signal has to be clearly louder than the environmental noise threshold, for it can be distinguished. We go on a short excursion into the field of signal processing and present two approaches for robust TDoA calculation that we have developed.

Cross correlation of audio signals is a promising approach, especially when a multichannel audio signal is available on a computer, for instance in the application example of calibrating a microphone array, as suggested in [42]. We have implemented the computationally more efficient phase correlation algorithm, which calculates the cross-power spectrum in the frequency domain, for which we have developed an algorithm for speaker tracking based on particle filtering, which we present in the following.

8.1 Cross Correlation

Both cross correlation and phase correlation with Fourier transform calculate a similarity measure of two audio tracks $g_1, g_2 \in \mathbb{R}^N$ for a shift $\tau \in \mathbb{N}$ of the tracks, which is the time difference of arrival (TDoA). Typically, a window of size N of both tracks is used to calculate the cross correlation as

$$(g_1 \star g_2)(\tau) \;=\; \sum_{i=1}^{N} g_1(i)\, g_2(\tau + i)\;. \tag{8.1}$$

As can be seen, to calculate the cross-correlation for all shifts τ in a window of size N time $\mathcal{O}(N^2)$ is required.

8.1.1 Phase Correlation Algorithm

We use phase correlation instead, which requires time $\mathcal{O}(N \log(N))$ for the Fast Fourier Transform (FFT) of both audio tracks, plus a linear term for calculating the cross-power spectrum, where the window size for FFT is $N = 2^u, u \in \mathbb{N}^+$. For the implementation we follow the approach of Valin et al. [37].

The first step is to compute the Fourier transform $G_1 = \mathcal{F}(g_1)$ and $G_2 = \mathcal{F}(g_2)$, where $G_1, G_2 \in \mathbb{C}^{\frac{N}{2}+1}$. The coefficients of the weighted normalized cross-power spectrum H are calculated by

$$H(k) \;=\; w(k)\,\frac{G_1(k)\,G_2(k)^*}{|G_1(k)|\,|G_2(k)|}\,, \qquad\qquad (1 \le k \le \tfrac{N}{2}+1)$$

where $(\cdot)^*$ denotes the complex conjugate, and $|\cdot|$ the complex magnitude. Normalization by the spectral magnitude ensures balancing of all frequencies by eliminating the overweight of low frequencies, which causes widened correlation peaks [37]. In contrast, the weighting parameter $w(k)$ may be used to emphasize certain frequency ranges. It is used as a frequency filter in the chirp spread spectrum algorithm, which is described in the next section. For comparison of two audio tracks we set $w(k) = 1$. By inverse Fourier transform we obtain an approximation to the cross correlation, the vector $h = \mathcal{F}^{-1}(H)$ which maps a correlation $h(\tau)$ to a time difference τ. As a result of the window size is $-\frac{N}{2} \le \tau \le \frac{N}{2}$.

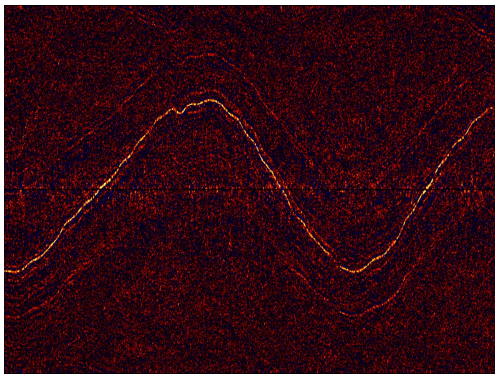
We use phase correlation to track a moving speaker in a recording of two audio microphones, for which we have implemented the algorithm in C++. We execute the algorithm on subsequent windows in the audio track. By merging the correlation windows, we obtain a matrix of a correlations $h(t, \tau)$ for a time frame t and time difference τ.

The obvious approach for calculating the time difference is to consider the correlation maximum. However, we saw that this is unreliable in moments of silence in the audio track. Alternatively, one might consider the matrix $h(t, \tau)$ as an image and combine different methods from the domain of pattern recognition [85]. For instance, a typical strategy is to remove noise with a Gaussian kernel, then enhance the edges with the Sobel operator, then use a threshold to define a line. Yet, we could still not achieve convincing results with these techniques, due to the small contrast.

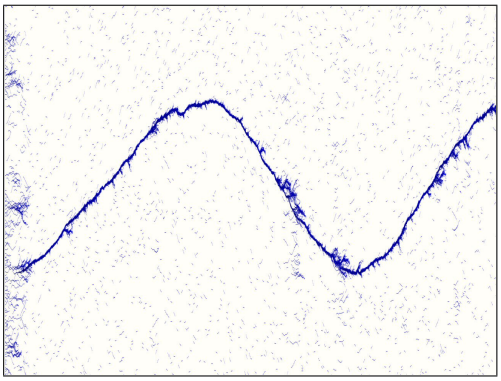
8.1.2 Particle Filter

We rely instead on the assumption that the speaker's trajectory, and the resulting time difference, is continuous. We have developed a particle filter in a one-dimensional state space that propagates in time, trying to follow the trace of highest correlation in every time frame of the correlation matrix.

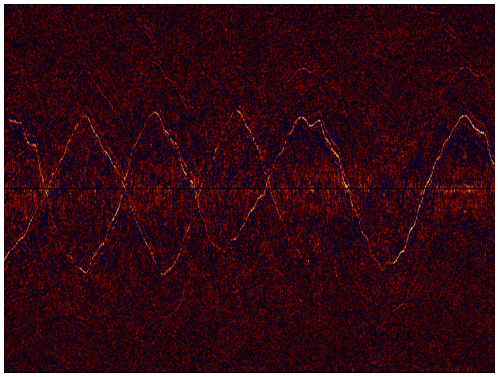
The particle filter consists of a set of particles $\mathcal{M} = \{(\mathbf{x}^{[k]}, w^{[k]})\}_{(1 \le k \le K)}$ which are updated in three algorithm components, sampling, weighting, and resampling, as



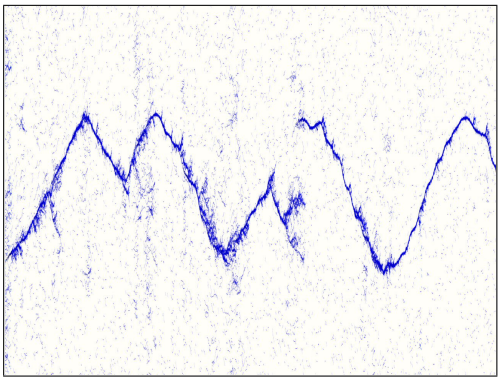
(a) Phase correlation diagram. Bright colors indicate positive correlation.



(b) Particle filter diagram. The particle filter follows the time difference of the speaker.

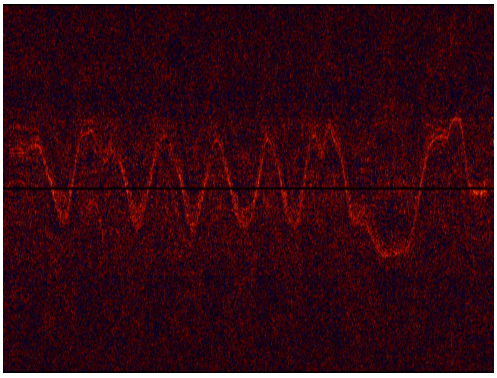


(c) Phase correlation diagram. Bright colors indicate positive correlation.

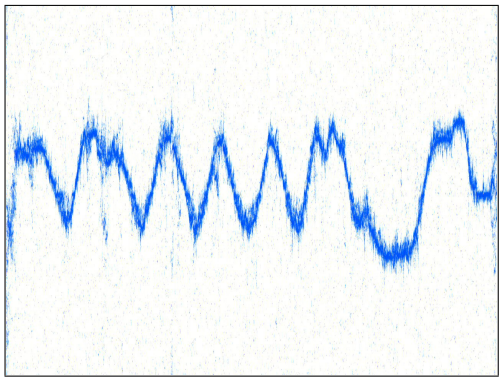


(d) Particle filter diagram. The particle filter cannot follow both speakers.

Figure 8.1: *Top*: One speaker moving between two microphones, reading a newspaper. The horizontal axis is the time axis in a recording of 37 seconds. *Bottom*: Tracking of two speakers moving between two microphones for 83 seconds. The window size is $N = 1024$, therefore the vertical axis is the TDoA in the interval $\pm 11.6\,\text{ms}$.



(a) Phase correlation diagram. Bright colors indicate positive correlation.



(b) Particle filter diagram. The particle filter follows the time difference of the quadrotor.

Figure 8.2: Tracking of the screaming uniform noise of an AR.Drone quadrotor of 88 seconds duration. The window size is $N = 512$, therefore the time difference is $\pm 5.8\,\text{ms}$.

described in Chapter 6. The state $\mathbf{x} = (\hat{\tau}, \dot{\tau})^T$ represents the time difference $\hat{\tau} \in \mathbb{R}$ and the change in time difference, the velocity $\dot{\tau} \in \mathbb{R}$. For the motion model we use a constant velocity model, where in every time frame t the position is updated by the velocity as

$$\hat{\tau}_{t+1}^{[k]} = \hat{\tau}_t^{[k]} + \dot{\tau}_t^{[k]} \quad \text{and} \quad \dot{\tau}_{t+1}^{[k]} = \dot{\tau}_t^{[k]} + \xi \,, \tag{8.2}$$

where ξ is a Gaussian random variable. For the observation model we decided to use a simplistic formulation, where the weight update is calculated from the q nearest measures by

$$p(\mathbf{z} \mid \mathbf{x}) = \sum_{\ell=-q}^{q} h(t, \lfloor \hat{\tau} \rfloor + \ell)\, \bar{\mathcal{N}}_q(\ell) \,, \tag{8.3}$$

where $\bar{\mathcal{N}}_q$ is a discrete Gaussian kernel of size $2q + 1$.

For initialization, all particles are uniformly distributed in the first frame $t = 0$. In every movement step, a particle propagates into the next time frame, depending on its velocity, where the weight is updated according to the observation. Resampling is performed according to the low variance resampling scheme [108]. We consider that the speaker may become silent and start somewhere else, the so-called "kidnapped robot problem" [108], for which we generate random uniformly distributed samples in every prediction step. For more details on the particle filter algorithm refer to Chapter 6.

8.1.3 Experimental Evaluation

In an experiment we placed two microphones in a distance of few meters and had a speaker walk between them back and forth, while reading a story from a newspaper. While speaking, the speaker was advised to reduce breaks between words and sentences,

in order to generate a continuous acoustic flow. We recorded the audio track at a sampling rate of 44,100 kHz and evaluated a section of 38 seconds with a window size of $N = 1024$ and 50% overlap of windows, corresponding to 3261 correlation frames. On the cross correlation result we see a clear line for the eye where the time difference of the speaker appears, framed by thinner lines of echoes from a wall, see Fig. 8.1(a). We see that the particle filter can well track that speaker in Fig. 8.1(b).

We repeated the experiment with two simultaneous speakers. We chose two male persons, yet with different vocal ranges, to facilitate distinction of both. For the first 50 seconds both persons were walking and speaking, then one of both speakers became silent, yielding a total duration of the audio track of 83 seconds. The evaluation yielded a clearly distinguishable correlation diagram, see Fig. 8.1(c), however the particle filter could not track both speakers, but followed only one of the correlation curves, see Fig. 8.1(d). This is not surprising, as the particle filter was not designed as a multi-hypothesis tracker.

In a third experiment we had a Parrot AR.Drone quadrotor hover between two microphones, for which the high-pitched screaming noise of its rotor gear is characteristic. We recorded an audio file of 87 seconds and evaluated with a window of $N = 512$, due to smaller distance of the microphones. According to the evaluation, see Fig. 8.2, even this sonorous mechanical sound is adequate for tracking by correlation.

Phase correlation of similar audio information of two microphones is robust and works with a continuous sound signals, as demonstrated in the experiments. A drawback is the high computational cost of the algorithm and the high amount of information to exchange between receiver nodes, which opposes the employment of phase correlation in a low-power wireless sensor network.

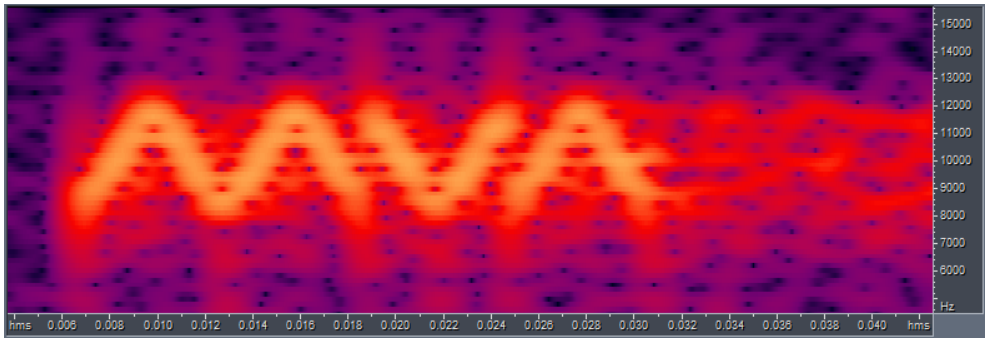
In contrast, pattern detection in an audio track uses the benefits of both worlds. No information is exchanged between nodes and the computational cost may be decreased if a short pattern is chosen. In the following we present *chirp spread spectrum* as an algorithm of audio pattern detection.

8.2 Chirp Spread Spectrum

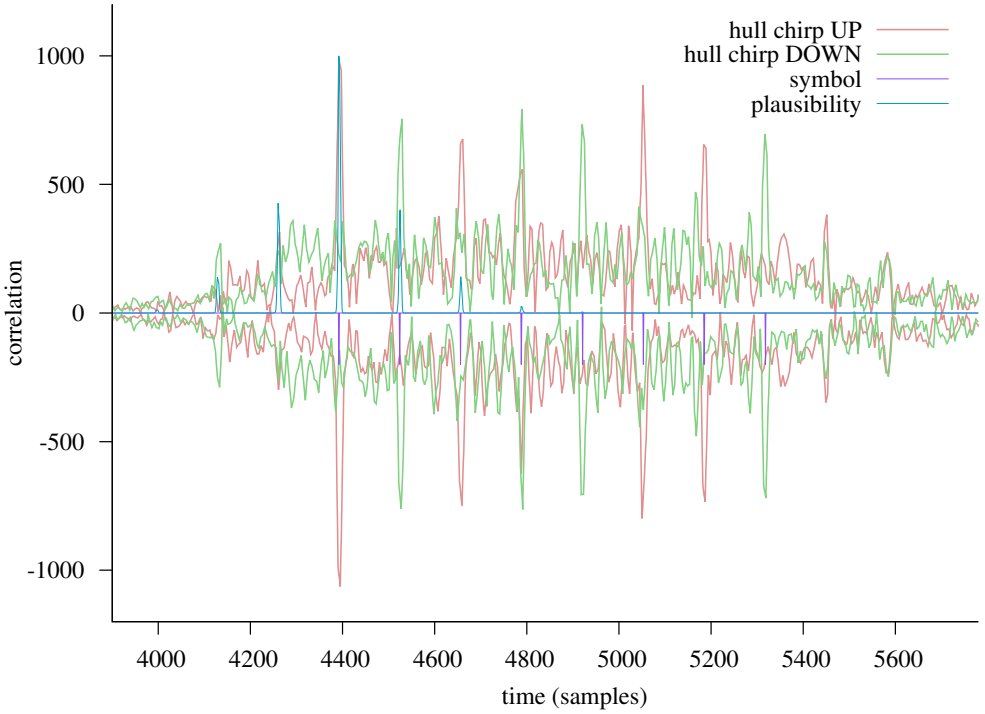
In digital communication, spread spectrum is a wideband technique to enhance the fault-tolerance and noise resilience of a signal transmission. In *chirp* spread spectrum (CSS) a linearly modulated sinusoid waveform is used [113, 129]. Depending on begin and end frequencies f_1 and f_2 and the duration of the chirp ℓ, for time $0 \leq t \leq \ell$ a chirp is defined in the time domain as

$$A(t) \;=\; \sin\left(2\pi\left(f_1 t + (f_2 - f_1)\frac{t^2}{2\ell}\right)\right) \;. \tag{8.4}$$

A chirp in an audio or ultrasound signal can be detected by calculating the correlation of the recorded signal g_1 and a pattern of the chirp g_2 which is created from $\left(A, \vec{0}\right)^T$



(a) The spectral view of the recorded audio track, where the horizontal axis is the time and the vertical axis is the frequency, illustrates the sequence of UP and DOWN chirps. Note the reverberation after the end of the sequence.



(b) The phase correlation of two chirp symbols UP (red) and DOWN (green) yields maxima in the correlation function. The most likely position of the sequence begin is found by calculating the product (blue) of eight test positions (magenta).

Figure 8.3: Evaluation of a recorded audio chirp sequence of 8 bits in the frequency range 8 kHz – 12 kHz and duration of a chirp of 3 ms. The transmitted bit string '10100110' (LSB first), corresponds to the ASCII letter 'e'.

by using a window size of $N = 2^u \geq \ell\,\Omega$, for $u \in \mathbb{N}^+$, where Ω is the sampling frequency. The rest of the window is padded with zeros. The phase correlation of the pattern and the recorded signal is calculated as described in the previous section. For CSS we choose the cross spectrum weight

$$w(k) = \begin{cases} 1 & \text{if } f_1 \leq \frac{k\Omega}{\frac{N}{2}+1} \leq f_2\ , \\ \zeta & \text{else}\ , \end{cases} \tag{8.5}$$

where $0 \leq \zeta < 1$, therefore reducing the influence of signals outside the frequency range of the chirp.

By means of the correlation, even if a notable amount of noise is contained in the recording, a sharp peak can be observed at shift position τ, if $f_2 \neq f_1$. The position of the peak determines the time of the recorded chirp. The magnitude, depending on the noise and the window size, is an indicator of the clarity of the detection.

The chirp spread spectrum technique can be used for robust transmission of a data word, for example to distinguish the sender of a signal by an identification number, by combining 2^q chirp symbols that represent a sequence of q bits. In the simplest form an "UP" chirp in the frequency range (f_1, f_2) and a "DOWN" chirp in (f_2, f_1) is chosen, where $f_1 < f_2$, representing "1" and "0", respectively.

In an experiment we created sequences of $n = 8$ bits, in total a message of 118 ASCII characters from a text. Bits of a sequence were encoded as UP and DOWN chirps in the frequency range 8 kHz to 12 kHz at a duration of 3 ms per chirp, yielding a total sequence length of 24 ms. The sequences were concatenated in an audio track with a pause of 400 ms between each sequence. The audio track of 50 seconds was played by a conventional loudspeaker and recorded by a high quality measurement microphone at a distance of few meters at a sampling rate of 44.1 kHz. For both chirp symbols we calculated h_1 as the correlation of the recorded signal and the UP pattern, and h_2 as correlation of the DOWN pattern respectively. We calculate the estimated begin of the sequence, which is required for determining the TDoA, by the product

$$C(t) = \prod_{i=0}^{n-1} \max\Big(h_1(t + i\ell), h_2(t + i\ell)\Big)\ , \tag{8.6}$$

taking into consideration that a sequence may consist of either symbols. Analogously, once the position of the sequence is determined, the bit values are obtained by comparison of the correlation values. See Fig. 8.3(a) for an example of the spectral view of a recorded sequence and Fig. 8.3(b) for the detected correlation of the sequence of chirps.

Chirp spread spectrum is an elegant technique to combine robust transmission of a signal under high environmental noise, and precise time determination of the signal by the correlation peak. We have executed the CSS algorithm by phase correlation to benefit from the low time complexity of the Fast Fourier Transform. However, for chirps of short duration calculating the correlation in the time domain, as defined in Eq. (8.1), is worth consideration.

8.3 Summary

In this chapter we have presented two approaches to increase the robustness of TDoA calculation. We have implemented a phase correlation approach to experiment with direct comparison of audio signals, and we have developed a particle filtering approach to ensure reliable tracking of a speaker's position, which we demonstrated in several experiments. Furthermore, we have created an algorithm based on chirp spread spectrum, where we demonstrated a promising result for error-resilient transmission and TDoA calculation of binary data.

9 The Energy-Autarkic Calibration-Free Ultrasound Tracking System (eCULTS)

The idea of TDoA without the need of pre-defined anchors proceeds in the industrial MicroTEC-Südwest project eCULTS, which follows from the origins of the calibration-free ultrasound localization system that was developed during this thesis. eCULTS is an acronym for **e**nergy-autarkic **C**alibration-free **UL**trasound **T**racking **S**ystem. The project, which is funded by the German Federal Ministry of Education and Research (BMBF), envisages the development of an indoor localization system for commercial intralogistics, where reference nodes are deployed that operate energy-autarkic, and that require no manual calibration during installation of the system. The goal of the project is to combine the high availability and cost efficiency of ultrasound based solutions to a localization system that can be installed and maintained with low effort by the customer.

eCULTS is a collaboration of academic and industrial partners. The Department of Computer Science (IIF) and the Department of Microsystems Engineering (IMTEK) of the University of Freiburg are involved as initiators and collaborators of the project, among several industrial collaborators such as the SICK AG, Pepperl+Fuchs, the HSG Institute for Microsystems Engineering (HSG-IMIT), and others. The project started in October 2012, the current funding phase lasts until June 2015.

9.1 Background

The domain of in-house logistics of commercial enterprises, sometimes also referred to as *intralogistics*, covers the entire range of organization, execution and optimization of in-house material flow and warehousing. While traditionally by the means of material transportation a static infrastructure is understood, involving conveyor belts and manually operated cranes and fork lifts, a trend towards autonomous agents has been developing in the past time.

An important factor is localization and tracking of these autonomous systems. In the outdoor domain of logistics GPS has been playing an important role for fleet management for years. However, in intralogistics is GPS barely available and with a precision of few meters too imprecise for navigation of autonomous units. Also, localization systems based on radio are usually too imprecise for positioning and navigation in indoor environments. Furthermore, they require extensive effort for installation and mainte-

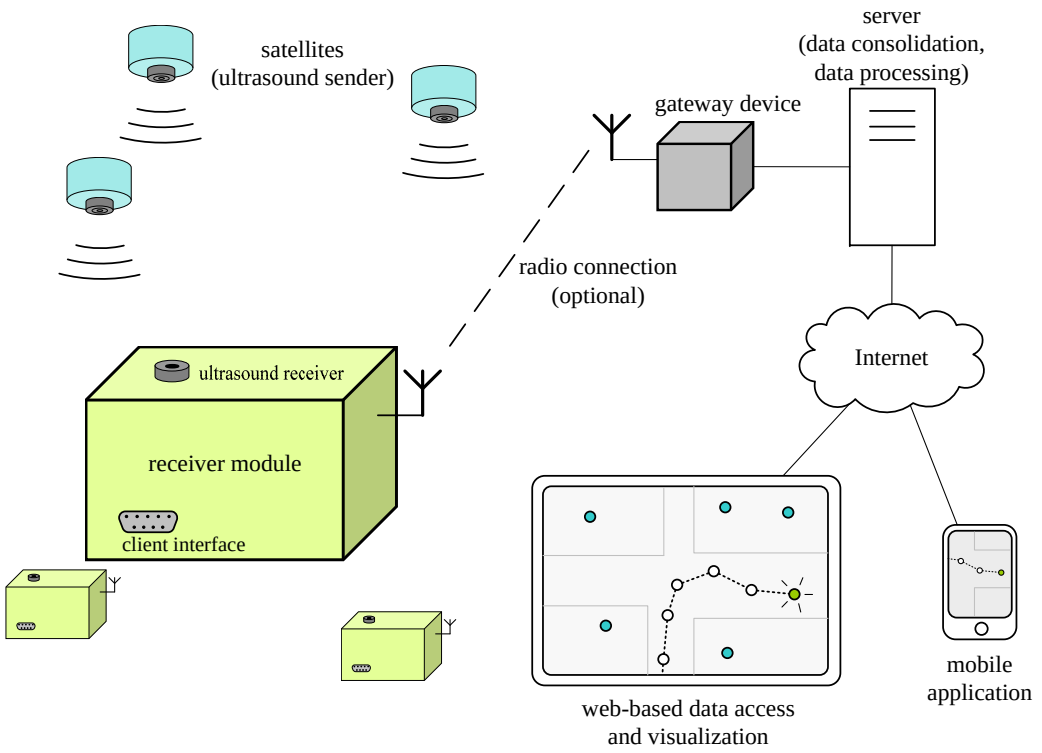


Figure 9.1: Prospected component overview of the *eCULTS* localization system. One or more mobile receivers navigate by means of static ultrasound beacons (satellites). The locality information is directly available to the receiver unit. Optionally, the navigation process is monitored by web-based applications.

nance of reference nodes, which implies that a large amount of technical knowledge is required by the customer – or the installation and maintenance is executed by the system provider, which is expensive to the customer.

Systems based on laser scanning are employed in many cases. These require effort for installation of scanable reference tags and for calibration of the laser systems. Yet, in many prospective plans in intralogistics, the employment of a large number of inexpensive small agents is advised, instead of a small number of expensive ones. It does not seem profitable to equip all these units with expensive high performance laser scanners. A low-priced alternative is required.

9.2 Localization of Mobile Receivers

In an application scenario, mobile units such as in the *small-scale redundant intralogistics system* (KARIS) [130] navigate autonomously in a warehouse, forwarding goods between a pick-up and a handoff place, which are scheduled dynamically. For this scenario, several signal emitters are regularly placed, preferably at the hall ceiling. The

autonomous units receive periodical ultrasound pulses from the emitters. From the times and time differences of the signals the receivers can determine their positions in a hall. Optionally, they forward the locality information to a server, where the data is consolidated, so the mobile units can be monitored and assigned tasks. See Fig. 9.1 for an overview of the prospective system concept.

Localization of receivers in eCULTS is done solely by evaluation of the time points of reception of the signals emitted by the ultrasound senders. No additional radio connection is required for this purpose. Optionally can an inertial measurement unit and odometry support the ultrasound localization. The ultrasound signals contain identification codes of the senders, so they can be distinguished by the receiver. A feasible technique may be chirp spread spectrum, as described in Section 8.2. Reliable discrimination of the senders is a requirement for all time-of-arrival based algorithms.

The setting in the eCULTS project may be considered as the "inverted" problem of calibration-free TDoA, compared to the scenario in Chapter 6 and the experimental Section 7.3. The advantage of the inversion is that a constant number of stationary emitters may be used to locate an arbitrary number of mobile agents, and that the locality information is instantly available to the agents where it is required.

Localization is done in a two-phased approach, containing a *calibration phase* and a *tracking phase*. After installation of the system begins the calibration phase, where the positions and phase differences of satellites are calculated by an automatic approach based on ultrasound measurements of a reference receiver. After accomplishment the tracking phase begins, which is the normal operation phase. By the initially gathered reference information, the receiver can now be positioned in relation to the reference. In practical implementation both phases are not clearly distinct. For instance, depending on the localization result can positions of satellites be tested for plausibility and corrected during operation of the system, if necessary.

9.2.1 Calibration Phase

As a simple suggestion, a test receiver is once brought close to all satellites. In this way, the distance and the time offset between two satellites are calculated from the minimum and maximum time differences that occur [51, 61]. An approximation of the distance is $\tilde{d} = \frac{1}{2\,c}(\tau_{\max} - \tau_{\min})$, where c is the sound velocity and τ is the maximum, respective minimum, observed time difference. The time offset between the senders is $\tilde{\delta} = \frac{1}{2}(\tau_{\max} + \tau_{\min})$. From the distances the Cartesian coordinates are calculated by solving an equation system $\|\mathbf{s}_i - \mathbf{s}_j\| = \hat{d}_{ij}$, where $\mathbf{s}_i$, $\mathbf{s}_j$ are satellites and $\hat{d}_{ij}$ is an observed distance between both. See Sections 3.1f. and 3.4.

Alternatively, measurements are conducted at multiple well-distributed random positions in the warehouse. Given measurements at position $\mathbf{m}_i$ of the time difference τ_{ijk} between senders $\mathbf{s}_j$ and $\mathbf{s}_k$ leads to a system of equations that is solved for positions $\mathbf{m}_i$, $\mathbf{s}_j$, $\mathbf{s}_k$ and the time offset δ_{jk} between senders j and k. Without loss of generality, we choose sender k as a reference node. Given a sufficient number of measurements,

	satellite time offset	
	known	*unknown*
Receiver is *stationary* to measure time differences	Closed-form solution of the hyperbolic Equation (9.1), instant localization with a single measurement	Multiple measurements at different places using Eq. (9.1). After that, the time offset is *known*.
Receiver moves *continuously*	Probabilistic filtering using Kalman or particle filter. Estimate is quickly available after filter initialization. Eqns. (9.2)(9.3).	Kalman or particle filter. Estimation of position, velocity and time offsets. Requires multiple measurements for initialization. Eqns. (9.2)(9.3).

Table 9.1: Different tracking modes of the receiver, depending of availability of synchronization and movement characteristics of the receiver.

a unique solution can exist. To solve the equation system iterative optimization approach are employed, such as gradient descent and the multivariate Gauss-Newton method [5], see Chapter 4. A mixture of both techniques is thinkable – as soon as some of the distances are known, the rest can be found more easily.

An important goal of the project is that the calibration procedure does not require professional staff, but can be performed by staff of the consumer. The person that conducts the calibration should be notified about the progress of the procedure, and he should receive specific instructions by adequate means, for instance by feedback from the display of a calibration device. Once the calibration phase is completed the configuration parameters are committed to the mobile receiver units, either by radio communication or by a one-time download.

9.2.2 Tracking Phase

Known time offsets. As soon as the positions $\mathbf{s}_j$ and time offsets δ_{jk} of the satellites are known, the receiver can directly calculate the time differences. The tracking problem is then reduced to finding the trajectory of the receiver. If the receiver is temporarily *stationary*, i.e. it does not move until a signal was received by every sender in range, then just a hyperbolic equation system of the form

$$\tfrac{1}{c}\|\mathbf{m}_i - \mathbf{s}_j\| - \tfrac{1}{c}\|\mathbf{m}_i - \mathbf{s}_k\| \;=\; T_{i_j j} - T_{i_k k} - (i_j a_j - i_k a_k) + \delta_{jk} \tag{9.1}$$

is solved for $\mathbf{m}_i$, which may be done quickly by linear estimators or by few iterations of the Gauss-Newton algorithm. $1 \le i_j, i_k \le \ell$ are continuous indices of the signals from the senders j and k.

However, if the receiver moves continuously, which is most likely the case for an autonomous unit, then an equation system is obtained with a new receiver position for every measurement, which leads to equations that are no longer hyperbolic:

$$\tfrac{1}{c}\|\mathbf{m}_{i_j j} - \mathbf{s}_j\| - \tfrac{1}{c}\|\mathbf{m}_{i_k k} - \mathbf{s}_k\| \;=\; T_{i_j j} - T_{i_k k} - (i_j a_j - i_k a_k) + \delta_{jk}\ . \tag{9.2}$$

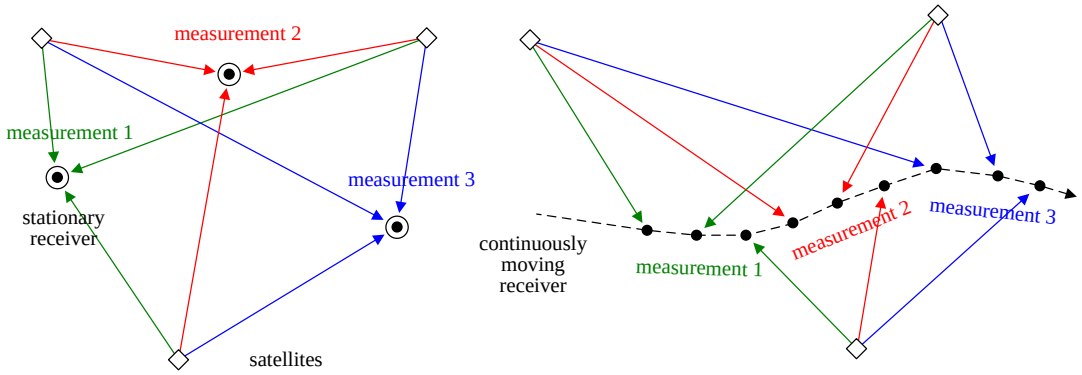


Figure 9.2: Three measurements from three senders. *Left*: The receiver is temporarily *stationary*, until it has obtained measurements from every sender. This results in nine equations for *three* unknown receiver positions. *Right*: The receiver moves continuously, which yields *nine* unknown receiver positions.

For ℓ measurements from m senders, the number of equations is ℓm. Even if the receiver moves on the floor, therefore in space $\mathbb{R}^2$, the number of variables is $2\ell m$, therefore the equation system is under-determined, see Fig. 9.2 for an example. Such a problem cannot be solved by the usual approaches without further assumptions.

In eCULTS one can assume coherent movement of the receiver, so recursive state estimators can be employed, for instance the Kalman filter. According to Eq. (9.2) we obtain a non-linear observation model, so using the Extended or the Unscented Kalman filter is advisable. Alternatively, a particle filter is used which is robust against non-Gaussian observations and which can represent a non-Gaussian state estimation, yet at the price of higher computational cost. We used particle filtering to track a moving sender, see Chapter 6. For the probabilistic estimators must be noted that positioning is not immediately available, in contrast to optimization or closed-form calculation, but requires some measurements for initialization of the filter, even if the time offset of the senders is given.

One approach for enabling optimization in the continuous movement problem, may be in the simple case to ignore the movement of the receiver. Probably, the velocity of the receiver is small, so the induced positioning error from its movement is acceptable. In a more advanced approach the velocity is estimated from previous localization and accordingly the movement is assumed linear, which is reasonable for the short time of measurement. The positions of the receiver are then reduced to a linear function that can be expressed by two variables in $\mathbb{R}^2$, which is not more than in the stationary case.

Unknown time offsets. If the offset between the senders is not known, and only the positions are given, then the shift in sending intervals can be observed, if the receiver moves towards a sender j, or away from it. This is called Frequency Difference of Arrival (FDoA), also known as the *Doppler effect*. Given a sufficient number of these

distance differences in a continuous movement of the receiver, a "finger print" of the possible locations is obtained which can be described by a probabilistic filter. Because of the reduction of multiple hypotheses using a particle filter is advised. For estimation of a state $(\mathbf{m}, \mathbf{v})^T$, where $\mathbf{v}$ is a velocity estimate, the hyperbolic constraint is

$$\tfrac{1}{c}\|\mathbf{m}_{i_1 j} - \mathbf{s}_j\| - \tfrac{1}{c}\|\mathbf{m}_{i_2 j} - \mathbf{s}_j\| \;=\; T_{i_1 j} - T_{i_2 j} - (i_1 - i_2)\, a_j \;, \tag{9.3}$$

where a_j is the given sender interval, and $1 \leq i_1, i_2 \leq \ell$ are two measurements of a receiver. More advanced is simultaneous estimation of the time offsets between two satellites j and k in a state $(\mathbf{m}, \mathbf{v}, \delta_{1k}, \ldots, \delta_{mk})^T$ using Eqns. (9.2). We have published such an approach that considers tracking of a moving receiver given unsynchronized senders in [131].

In case the movement of the receiver is not continuous, but temporarily stationary, then simpler hyperbolic equations are obtained of the form (9.1) which are solved by optimization for receiver positions $\mathbf{m}_i$ and offsets δ_{jk}. Once the offsets are known, they can be re-used in subsequent state estimation. For a brief sketch of the four operation modes see Table 9.1.

9.3 Challenges

One of the most challenging issues to be solved originates from the fact that the described model ignores skew of imprecise clock generators. Synchronization is valid only for a short period of time after calibration of the system if the satellites manage their own local clocks and are not externally triggered.

Standard clock oscillators are precise up to 10 parts per million, which is equivalent to an error of 0.6 ms/min, or 36 ms/h, or 0.86 s/day. The required synchronization for TDoA localization in the range of centimeters of 0.1 ms is maintained for only 10 seconds. Therefore, the time offsets cannot be trusted even if the sender clocks were synchronized initially. As a solution, one of the following, or the combination, are thinkable:

1. The satellites may be synchronized by additional hardware that receives a trigger signal, either by wire or wireless. This may increase cost and energy consumption, and the availability of the trigger signal must be guaranteed.
2. Monitoring of the time skew by stationary reference receivers. Here, the updated skew information has to be forwarded to the mobile receiver units that need a radio connection to obtain the updated offset information. Depending on the scenario, a large number of references may be required to cover all satellites.
3. Implementation of algorithms that can calculate the offset, as in Eq. (9.2), or do not require it at all, as in Eq. (9.3). The variability of the offsets raises the complexity of the equation system and increases the required time and number of measurements to calculate the receiver position.

9.4 Summary

Scope of the MicroTEC-Südwest project eCULTS is the development of an ultrasound localization system for industrial intralogistics. The prospected application scenario is localization and navigation of autonomous mobile units for material transportation. In the context of the project new problems of TDoA localization are adressed, which are calibration-free localization of a moving receiver and error-resilient transmission of distinguishable ultrasound signals.

eCULTS is a challenging project, both from an academic point of view of TDoA localization, and from the collaboration aspect. The academic interests of university encounter the pragmatic strategy of industrial enterprises, where the challenge is to amalgamate the expectations of cooperation partners from different areas. However, we see here the opportunity to contribute with academic research to the development of an industrial consumer product, of which the offspring was academic research.

10 Conclusions

In this thesis we have contributed to the domain of indoor localization by proposing several novel approaches for TDoA-based calibration-free localization. We have addressed the problem from several points-of-view, which are rapid approximative algorithms, the problem of a complete solution for the general setting, and resilience of algorithms in case of measurement errors. For these partial problems we have presented novel algorithms mainly in four fields – the far-field assumption, local optimization, branch-and-bound algorithms, and probabilistic state estimation.

10.1 Summary of this Thesis

Under the assumption that signals originate from remote locations, the so-called *far-field assumption*, the propagation direction is reduced to parallel vectors, which simplifies the equation system and allows for closed-form algorithms. Far-field approximation yields a fast and robust solution for self-localization by using radio signals and natural sounds from the environment, without the need to control these signals, furthermore it can be used as initial guess for optimization.

Our main contribution to far-field approximation is the Ellipsoid TDoA method. The algorithm uses the fact that pairs of TDoA measurements from three receivers in the plane form an ellipse when senders emit signals from the distance. This ellipse is robustly determined by regression, and the distances and angles between the receivers are calculated from the parameters of the ellipse. The concept is also demonstrated for the three-dimensional case. The Ellipsoid TDoA method is to date to our knowledge the first algorithm that solves the minimum problem in the TDoA far-field assumption in two and three dimensions for synchronized and unsynchronized receivers, hereby surpassing previous approaches that consider this problem. We have demonstrated the robustness of the algorithm in simulation and in real-world experiments, where we have shown that the Ellipsoid method can also be employed when the far-field assumption is violated to some extent.

Approximative algorithms are limited in generality, especially for the minimum cases when measurements are rare. In the field of non-linear optimization we have set focus to the failure cases. Due to the high dimensionality of the calibration-free TDoA problem, iterative optimization of a system of hyperbolic equations tends not to find the global optimum, which is the only acceptable solution. This problem is mostly inevitable in local optimization, however we have presented an algorithm that increases the chance to find the global minimum. We have proposed the Cone Alignment algorithm, which

is an iterative optimization algorithm based on a mass-spring simulation where signal and receiver positions are represented by physical particles. A TDoA error function is modeled by physical springs between the particles. By relaxation of the springs the error function is minimized. Hereby, the particles gather momentum to overcome local minima. In extensive simulations we tested the characteristics of the algorithm and compared it to standard optimization algorithms, which are gradient descent, the Gauss-Newton method and the Levenberg-Marquardt algorithm. We demonstrated that by repeated execution the algorithm could find the correct solution in 99.4 % of random simulations of the two-dimensional minimum problem if no TDoA error is assumed, which is more than the other algorithms.

Although the discussed local search algorithms promise a high chance of success, the intrinsic problem of local iterative optimization persists: The question *if* and *when* the solution is found, remains unanswered. To address this problem of completeness, we have developed a polynomial time branch-and-bound algorithm that is a proof to enumerate all solutions to calibration-free TDoA up to an error bound ϵ. The algorithm is based on subdivision of a five-dimensional search space into subspaces, by which the minimum problem of four receivers in the plane is represented, and test of each subspace for being a possible explanation of the TDoA measurements, given uncertainty ϵ. In practical implementation we demonstrated that the algorithm is asymptotically faster than enumerating all cells of size ϵ, which is the brute-force approach. The branch-and-bound algorithm is to our knowledge the first theoretical solution to the calibration-free TDoA problem.

The primarily application-oriented contribution in this thesis is the calibration-free ultrasound localization system. During the time of this thesis we have developed a software framework for localization that provides the algorithmic backbone of an ultrasound localization system for indoor application. In the localization system, a mobile ultrasound sender emits short pulses that are received by several ultrasound receivers, by which the sender is located. As its main characteristic, the localization system is *calibration-free*, therefore no initial manual measurement of the reference receivers is required. This eliminates a major source of agony in TDoA-based localization, where usually receivers were measured by hand, or placed precisely in geometrical constraints, for example in a grid. In our contributed localization system, the receivers may be randomly distributed in a room. The calibration of the system is performed by our TDoA auto-calibration algorithms.

For the localization system we have created a probabilistic state estimation algorithm based on a particle filter that estimates both the mobile sender and the positions of receivers. The algorithm is robust against measurement outliers, which is a common pitfall in TDoA localization. This is achieved through a probabilistic sensor model for TDoA data which explicitly considers the measurement uncertainty. For the reliable initialization of the particle filter, we apply an iterative optimization approach to multiple subsets of TDoA data, where the best solution is implicitly selected by appropriate weighing of the sensor model.

We have verified the robustness of the ultrasound localization system in extensive experiments in a large indoor hall, where we demonstrate that our approach ensures proper initialization of the particle filter and provides accurate position estimates for the signal beacon and the receivers even in case of measurement outliers. Compared to position references of an optical motion capture system we have achieved mean position errors below 5 centimeters.

In the following we discuss a categorization of our presented algorithms in terms of their affinity to the three subproblems of TDoA – rapidness, completeness, and robustness.

10.2 Categorization of Algorithms

The field of calibration-free TDoA localization is promising in its aspects and versatile in application, yet also multi-layered and complex. The non-linearity and high dimensionality of the minimum problem reduce the chance of an "allrounder algorithm" that solves the general problem fast and reliable. However, during the time of this thesis we have approached the problem of calibration-free TDoA localization based on three main aspects. First, we have developed and analyzed approximative solutions that lead to a *rapid*, yet precise solution in many cases. Second, we have worked on the *completeness* of a solution, i.e. the difficult characteristics of TDoA that prevent to find the correct result, even if it does exist. Third, we improved on the response of calibration-free TDoA towards measurement outliers, and created an algorithm that can track a target with high *robustness* and high precision. We attempt to arrange our algorithms in a three-dimensional diagram of these categories for a qualitative discussion of their characteristics. See Fig. 10.1 for our suggested arrangement. Although not of our origin, we have included Pollefeys' factorization algorithm, because of its distinctness from the other algorithms.

Rapidness. We start by consideration of rapidness of algorithms. At the top end are the statistical approximations that estimate distances based on the distribution of a signal. For the consideration of m measurements they depend on the solution of a linear equation system, and therefore require a runtime of $\mathcal{O}(m^3)$ for simple matrix algorithms and for a constant number of references to be estimated. Some of the algorithms are even faster, when only minima, maxima, and average of measurements are determined in $\mathcal{O}(m)$. Slightly slower is the factorization approach [42] where a small number of iterative steps is recommended to improve the numerical quality of the result. Next are the iterative optimization approaches, where Gauss-Newton is fast due to the quadratic convergence, yet it can be used only if an initialization close to the solution is provided. The iterative gradient-based and spring relaxation approaches tend to be slow at the end of the runtime when the gradient is shallow. Still more efficient than brute-force search, yet with exponential growth of the search tree, is the branch-and-bound algorithm a comparatively time-consuming algorithm.

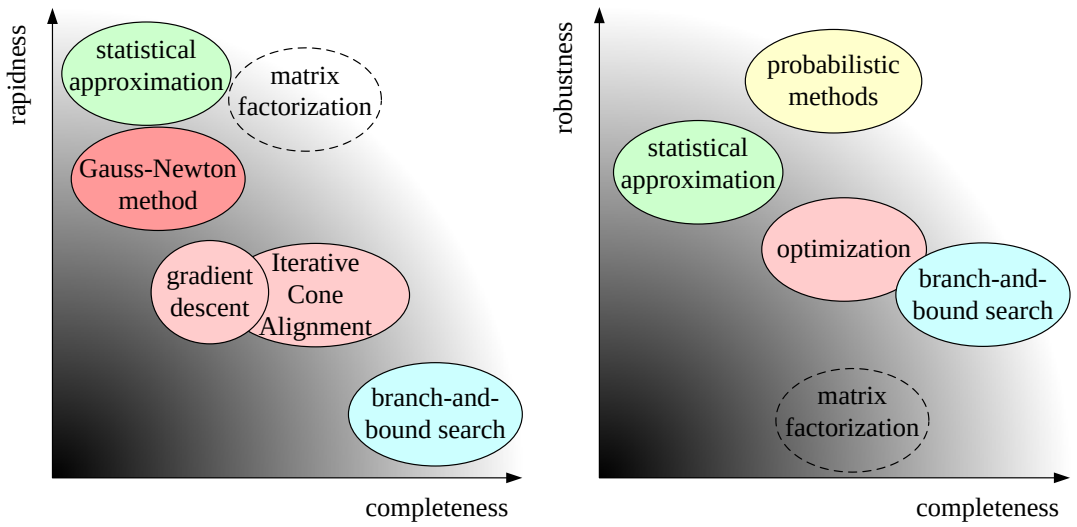


Figure 10.1: Completeness of algorithms versus rapidness and robustness. The three criteria can be arranged in a three-dimensional quality space. For calibration-free TDoA localization exist algorithms that yield a quick approximation, or are complete, or robust to measurement errors.

Completeness. While rapidness might be only a symptomatic criterion, the completeness of the solution is one of the central challenges that we pursued. We defined a *solution* as the ability of an algorithm to converge towards the true result, if the setting is determined and non-degenerate, and if the measurement error approaches zero. In such a setting, the branch-and-bound algorithm is complete, as it provably finds the solution.

All other proposed algorithms are incomplete. We rely on numerical argumentation to categorize the level of completeness, and consider the ability to return the uncertainty of a result. For instance, the iterative algorithms may return the wrong result due to local minima, yet this may be detected by the residual. Matrix factorization may fail due to side conditions such as semi-definite matrix properties, and the probabilistic algorithms may fail due to inability to initialize. Mass-spring optimization appears to be more reliable than gradient-based optimization, which was numerically demonstrated, and both are more reliable than Gauss-Newton alone. Last, statistical approximation is intrinsically error-prone for small numbers of receivers and signals and depends strongly on assumptions.

Robustness. The last category is the behavior of algorithms in case of measurement errors. For most algorithms we assume that the distribution of errors is Gaussian. However, this is only realistic for errors caused by measurement imprecision. Environmental effects, such as reflections from walls cause errors of high magnitude that are hardly predictable and hard to describe by a distribution and in a model.

We have created a probabilistic state estimation algorithm for TDoA based on a particle filter that uses optimization attempts for initialization, which are infused and weighted into the filter belief, where a single successful attempt can initialize the filter. The subsequent tracking phase is also implicitly robust, which we demonstrated in experiments with outliers of large magnitude. Also, if the algorithm loses the TDoA signal due to shadowing of the signal, it can recover reliably, which we also demonstrated. With these characteristics, our particle filter algorithm is the top end of robustness in our scale.

Algorithms follow in terms of robustness that use over-determination to compensate for errors. For instance, some statistical approximation algorithms use least squares regression, so do the gradient bases approaches. However, least squares minimization can work sufficiently only under the Gaussian assumption, where it is equivalent to likelihood maximization. Non-proportional outliers are not considered in these models, and the algorithms that rely on least-squares are vulnerable to such errors of large magnitude. Error mitigation algorithms such as RANSAC could possibly be used to eliminate the non-proportional errors. As their application is probably easier in rapid closed-form algorithms which are repeated with a different subset of measurements, we consider the statistical approximation algorithms as more robust than the time-consuming iterative algorithms. We suppose the branch-and-bound algorithm as less robust, as the limit of the ϵ-test is susceptible to exclusion caused by a single outlier – and repeated tests with subsets of measurements would further enlarge the runtime. The end of the flagpole is marked by the matrix factorization approach that induces position errors in the order of 10^3, compared to the measurement errors.

During the time of this thesis we have created for each of the three criteria an algorithm that marks the head among the algorithms that we are aware of in literature. It depends on the requirements of application which one is adequate to engage in a specific TDoA localization scenario. We discuss now an outlook towards future challenges in calibration-free TDoA.

10.3 Outlook

We have proposed several promising approaches to anchor-free TDoA in this thesis, covering a large bandwidth of application scenarios. However, a number of improvements are thinkable.

We suggest enhancement of the Ellipsoid TDoA method, which requires consideration of more than the minimum number of receivers, which is not straightforward due to the geometric representation. Also a combination of approximative and iterative algorithms might be considered, with the goal to create a generalized solver with adaptive choice of algorithms, minimizing the computational time.

In the probabilistic approaches we suggest the employment of an adaptive algorithm that decreases the state size and the number of particles by estimating only the moving

beacon once the receiver locations have been determined, with the purpose to reduce the computational cost of the algorithm. Here, malicious relocation of a receiver can be detected by evaluation of the residuum of the TDoA equation system and compensated by temporary recalibration using the full state size. Furthermore, the employment of the Extended and the Unscented Kalman Filter may be considered, which are more efficient in terms of computational cost. Yet, it is an open question how to compensate for the non-linearities of the TDoA equation system and the non-Gaussian ambiguities of position estimation while using the Kalman Filter. For instance, an alternative for our proposed sample injection during of initialization of the probabilistic algorithm is required.

An important aspect of future work on anchor-free time-of-arrival based localization is unification and standardization of the variety of scattered approaches. One the one hand is a multitude of different T(D)oA application scenarios with and without synchronization of either receivers or senders, varying mobility of receivers or senders, different acquisition sources of signals, where in many cases the same algorithms may be applied with modified side conditions. On the other hand are the different sensors of mobile handheld devices and autonomous systems, such as (ultra-)sound emitters and microphones, accelerometers, micro-electromechanical gyroscopes, odometry, or even laser scanners, where sensor data fusion approaches are required to unify anchor-free TDoA localization with other absolute and relative means of navigation.

Here is the development of a generalized abstract TDoA localization framework required that can be accessed by a clearly distinguished programming interface by the user. By such a framework one might assemble a promising strategy for localization and navigation, for example said combination of fast approximation and precise optimization. Also the integration into established robotics and navigation frameworks such as OpenSLAM[1] and the Robot Operating System[2] should be considered. Parts of these suggestions for unification and standardization are subject to consideration in the ongoing industrial eCULTS project.

[1] OpenSLAM, `http://www.openslam.org`
[2] Robot Operating System, `http://www.ros.org`

Bibliography

[1] Nissanka B. Priyantha, Anit Chakraborty, and Hari Balakrishnan. The Cricket Location-Support System. In *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 32–43, 2000.

[2] Chunyi Peng, Guobin Shen, and Yongguang Zhang. BeepBeep: A High-Accuracy Acoustic-Based System for Ranging and Localization using COTS Devices. *ACM Transactions on Embedded Computing Systems (TECS)*, 11(1):4, 2012.

[3] W. Navidi, W.S. Murphy, and W. Hereman. Statistical Methods in Surveying by Trilateration. *Computational statistics & data analysis*, 27(2):209–227, 1998.

[4] Henrik Stewénius. *Gröbner Basis Methods for Minimal Problems in Computer Vision*. PhD thesis, Lund University, April 2005.

[5] Johannes Wendeberg, Fabian Höflinger, Christian Schindelhauer, and Leonhard Reindl. Calibration-Free TDOA Self-Localization. *Journal of Location Based Services*, 2013.

[6] Jean-Marie Zogg. GPS – Essentials of Satellite Navigation. u-blox AG, 2009.

[7] Le Yang and K. C. Ho. An Approximately Efficient TDOA Localization Algorithm in Closed-Form for Locating Multiple Disjoint Sources With Erroneous Sensor Positions. *IEEE Transactions on Signal Processing*, 57:4598–4615, Dec. 2009.

[8] K. C. Ho, Xiaoning Lu, and L. Kovavisaruch. Source Localization Using TDOA and FDOA Measurements in the Presence of Receiver Location Errors: Analysis and Solution. *IEEE Transactions on Signal Processing*, 55(2):684–696, 2007.

[9] M.D. Gillette and H.F. Silverman. A Linear Closed-Form Algorithm for Source Localization From Time-Differences of Arrival. *Signal Processing Letters, IEEE*, 15:1–4, 2008.

[10] Julius Smith and Jonathan Abel. Closed-Form Least-Squares Source Location Estimation from Range-Difference Measurements. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 35(12):1661–1669, 1987.

[11] Christopher Drane, Malcolm Macnaughtan, and Craig Scott. Positioning GSM Telephones. *IEEE Communications Magazine*, 36:46–54, 1998.

[12] Ralph D. Hippenstiel, Timothy Haney, and Tri T. Ha. Improvement of the Time Difference of Arrival (TDOA) estimation of GSM Signals Using Wavelets. *NPS-EC-00-08, Naval Postgraduate School*, 2000.

[13] Veljo Otsason, Alex Varshavsky, Anthony LaMarca, and Eyal de Lara. Accurate GSM Indoor Localization. In *Ubicomp*, pages 141–158, 2005.

[14] Brian Ferris, Dirk Hähnel, and Dieter Fox. Gaussian Processes for Signal Strength-Based Location Estimation. In *Proceedings of Robotics: Science and Systems Conference (RSS)*, 2006.

[15] A. Savvides, C.C. Han, and M.B. Strivastava. Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors. In *Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 166–179. ACM, 2001.

[16] Mihail L. Sichitiu and Vaidyanathan Ramadurai. Localization of Wireless Sensor Networks with a Mobile Beacon. In *Proceedings of the First IEEE Conference on Mobile Ad-hoc and Sensor Systems*, pages 174–183, 2004.

[17] Ingwer Borg and Patrick J.F. Groenen. *Modern Multidimensional Scaling: Theory and Applications*. Springer Verlag, 2005.

[18] Jose A. Costa, Neal Patwari, and Alfred O. Hero III. Distributed Weighted-Multidimensional Scaling for Node Localization in Sensor Networks. *ACM Transactions on Sensor Networks (TOSN)*, 2(1):39–64, 2006.

[19] Tynia Yang, Jinze Liu, Leonard McMillan, and Wei Wang. A Fast Approximation to Multidimensional Scaling. In *IEEE workshop on Computation Intensive Methods for Computer Vision*, 2006.

[20] G.C. Calafiore, L. Carlone, and M. Wei. A Distributed Gauss-Newton Approach for Range-based Localization of Multi Agent Formations. In *2010 IEEE International Symposium on Computer-Aided Control System Design (CACSD)*, pages 1152–1157. IEEE, 2010.

[21] Srdjan Capkun, Maher Hamdi, and Jean Pierre Hubaux. GPS-free positioning in mobile ad-hoc networks. In *Proceedings of the 34th Annual Hawaii International Conference on System Sciences*, page 10 pp., Jan. 2001.

[22] Lance Doherty, Kristofer S. J. Pister, and Laurent El Ghaoui. Convex position estimation in wireless sensor networks. In *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1655–1663, 2001.

[23] Pratik Biswas and Yinyu Ye. Semidefinite Programming for Ad Hoc Wireless Sensor Network Localization. In *IPSN '04: Proceedings of the 3rd international symposium on Information processing in sensor networks*, pages 46–54, New York, NY, USA, 2004. ACM.

[24] A. Efrat, D. Forrester, A. Iyer, S.G. Kobourov, C. Erten, and O. Kilic. Force-Directed Approaches to Sensor Localization. *ACM Transactions on Sensor Networks (TOSN)*, 7(3):1–25, 2010.

[25] K. Coogan, V. Khare, S.G. Kobourov, and B. Katz. MSDR-D Network Localization Algorithm. In *Proceedings of ALGOSENSORS '10*, 2010.

[26] Frank Dabek, Russ Cox, Frans Kaashoek, and Robert Morris. Vivaldi: A Decentralized Network Coordinate System. In *Proceedings of the ACM SIGCOMM '04 Conference*, Aug. 2004.

[27] H. Schau and A. Robinson. Passive Source Localization Employing Intersecting Spherical Surfaces from Time-of-Arrival Differences. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 35(8):1223–1225, 1987.

[28] Benjamin Friedlander. A Passive Localization Algorithm and Its Accuracy Analysis. *IEEE Journal of Oceanic Engineering*, 12(1):234–245, 1987.

[29] Y.T. Chan and K.C. Ho. A Simple and Efficient Estimator for Hyperbolic Location. *IEEE Transactions on Signal Processing*, 42(8):1905–1915, 1994.

[30] Ralph Bucher and D. Misra. A Synthesizable VHDL Model of the Exact Solution for Three-Dimensional Hyperbolic Positioning System. *Vlsi Design*, 15(2):507–520, 2002.

[31] Wade H. Foy. Position-Location Solutions by Taylor-Series Estimation. *IEEE Transactions on Aerospace and Electronic Systems*, AES-12(2):187–194, March 1976.

[32] Don J. Torrieri. Statistical Theory of Passive Location Systems. *IEEE Transactions on Aerospace and Electronic Systems*, AES-20(2):183–198, March 1984.

[33] Dragana Carevic. Automatic Estimation of Multiple Target Positions and Velocities Using Passive TDOA Measurements of Transients. *IEEE Transactions on Signal Processing*, 55:424–436, Feb. 2007.

[34] G. Shen, R. Zetik, and R. S. Thoma. Performance Comparison of TOA and TDOA Based Location Estimation Algorithms in LOS Environment. In *Positioning, Navigation and Communication, 2008. WPNC 2008. 5th Workshop on*, pages 71–78. Ieee, 2008.

[35] Kenneth D. Frampton. Acoustic Self-Localization in a Distributed Sensor Network. *IEEE Sensors Journal*, 6(1):166–172, 2006.

[36] Yong Rui and Dinei Florencio. New direct approaches to robust sound source localization. In *Proceedings of IEEE ICME 2003*, pages 6–9. IEEE, 2003.

[37] Jean-Marc Valin, François Michaud, Jean Rouat, and Dominic Létourneau. Robust Sound Source Localization Using a Microphone Array on a Mobile Robot. In *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*, pages 1228–1233, 2003.

[38] P. Pertila, M. Mieskolainen, and M.S. Hamalainen. Passive self-localization of microphones using ambient sounds. In *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, pages 1314–1318. IEEE, 2012.

[39] R.L. Moses, D. Krishnamurthy, and R.M. Patterson. A Self-Localization Method for Wireless Sensor Networks. *EURASIP Journal on Advances in Signal Processing*, pages 348–358, 2003.

[40] Joshua N. Ash and Randolph L. Moses. Acoustic time delay estimation and sensor network self-localization: Experimental results. *The Journal of the Acoustical Society of America*, 118:841–850, 2005.

[41] R. Biswas and S. Thrun. A Passive Approach to Sensor Network Localization. In

Proceedings of the International Conference on Intelligent Robots and Systems, 2004. (IROS 2004). 2004 IEEE/RSJ, volume 2, pages 1544–1549, 2004.

[42] M. Pollefeys and D. Nister. Direct computation of sound and microphone locations from time-difference-of-arrival data. In *2008 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2445–2448. IEEE, 2008.

[43] Yubin Kuang and Kalle Åström. Stratified Sensor Network Self-Calibration From TDOA Measurements. In *Proceedings of 21st European Signal Processing Conference (EUSIPCO)*, 2013.

[44] S. Thrun. Affine Structure From Sound. In *Proceedings of Conference on Neural Information Processing Systems (NIPS)*, Cambridge, MA, 2005. MIT Press.

[45] Yubin Kuang, Erik Ask, Simon Burgess, and Kalle Åström. Understanding TOA and TDOA Network Calibration using Far Field Approximation as Initial Estimate. In *International Conference on Pattern Recognition Applications and Methods*, 2012.

[46] Simon Burgess, Yubin Kuang, and Kalle Åström. Node Localization in Unsynchronized Time of Arrival Sensor Networks. In *Proceedings of the 21st International Conference on Pattern Recognition*, 2012.

[47] R. Biswas and S. Thrun. A Distributed Approach to Passive Localization for Sensor Networks. In *Proceedings of the National Conference on Artificial Intelligence*, volume 20, page 1248. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2005.

[48] Zheng Sun, Aveek Purohit, Kaifei Chen, Shijia Pan, Trevor Pering, and Pei Zhang. PANDAA: Physical Arrangement Detection of Networked Devices through Ambient-Sound Awareness. In *Proceedings of the 13th International Conference on Ubiquitous Computing (UbiComp)*, pages 425–434. ACM, 2011.

[49] J. Saloranta and G. Abreu. Solving the Fast Moving Vehicle Localization Problem via TDOA Algorithms. In *2011 8th Workshop on Positioning Navigation and Communication (WPNC)*, pages 127 –130, 2011.

[50] Erik Ask, Simon Burgess, and Kalle Åström. Minimal Structure and Motion Problems for TOA and TDOA Measurements with Collinearity Constraints. In *Proceedings of 2nd International Conference on Pattern Recognition Applications and Methods (ICPRAM)*, 2013.

[51] T. Janson, C. Schindelhauer, and J. Wendeberg. Self-Localization Application for iPhone using only Ambient Sound Signals. In *Proceedings of the 2010 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 259–268, Nov. 2010.

[52] Niklas Goby. Kommunikations-Komplexität der Assoziationsproblematik in der Schalllokalisation. Master's thesis, University of Freiburg, 2012.

[53] Paul J. Besl and Neil D. McKay. A Method for Registration of 3-D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 14(2), 1992.

[54] Johannes Wendeberg, Fabian Höflinger, Christian Schindelhauer, and Leonhard Reindl. Anchor-free TDOA Self-Localization. In *2011 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–10. IEEE, 2011.

[55] Ronald L. Graham. An Efficient Algorithm for Determining the Convex Hull of a Finite Planar Set. *Information Processing Letters*, 1(4):132–133, 1972.

[56] C. Bradford Barber, David P. Dobkin, and Hannu Huhdanpaa. The Quickhull Algorithm for Convex Hulls. *ACM Transactions on Mathematical Software (TOMS)*, 22(4):469–483, 1996.

[57] K.S. Arun, T.S. Huang, and S.D. Blostein. Least-Squares Fitting of Two 3-D Point Sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 698–700, 1987.

[58] E. A. Dean. Atmospheric Effects on the Speed of Sound. Technical report, Physics Department, University of Texas at El Paso, 1979.

[59] Christian Schindelhauer, Zvi Lotker, and Johannes Wendeberg. Brief Announcement: Network Synchronization and Localization Based on Stolen Signals. In *Proceedings of 30th Annual ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing (PODC)*, 2011.

[60] C. Schindelhauer, Z. Lotker, and J. Wendeberg. Network Synchronization and Localization Based on Stolen Signals. In *Proceedings of 18th International Colloquium on Structural Information and Communication Complexity (SIROCCO)*, 2011.

[61] Johannes Wendeberg, Thomas Janson, and Christian Schindelhauer. Self-Localization based on Ambient Signals. *Theoretical Computer Science*, 453:98–109, 2012.

[62] Thomas Janson, Christian Schindelhauer, and Johannes Wendeberg. Self-localization Based on Ambient Signals. In *Proceedings of 6th International Workshop on Algorithms for Sensor Systems, Wireless Ad Hoc Networks and Autonomous Mobile Entities (ALGOSENSORS)*, 2010.

[63] Christian Schindelhauer and Johannes Wendeberg. Localization Solely based on Ambient Signals. Technical Report 261, University of Freiburg, 2010.

[64] Simon Burgess, Yubin Kuang, Johannes Wendeberg, Kalle Åström, and Christian Schindelhauer. Minimal Solvers for Unsynchronized TDOA Sensor Network Calibration using Far-Field Approximation. In *Proceedings of 9th International Symposium on Algorithms and Experiments for Sensor Systems, Wireless Networks and Distributed Robotics (ALGOSENSORS)*, 2013.

[65] Nikolai Nawri. Berechnung von Kovarianzellipsen. Received from: http://imkbemu.physik.uni-karlsruhe.de/~eisatlas/covariance_ellipses.pdf, 1996.

[66] W. Gander, G.H. Golub, and R. Strebel. Least-Square Fitting of Circles and Ellipses. *BIT Numerical Mathematics*, 34(4):558–578, Dec. 1994.

[67] Andrew W. Fitzgibbon, Maurizio Pilu, and Robert B. Fisher. Direct Least

Squares Fitting of Ellipses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):476–480, 1999.

[68] Xianghua Ying, Li Yang, Jing Kong, Yongbo Hou, Sheng Guan, and Hongbin Zha. Direct Least Square Fitting of Ellipsoids. In *21st International Conference on Pattern Recognition (ICPR)*, pages 3228–3231. IEEE, 2012.

[69] T. Eren, D. K. Goldenberg, W. Whiteley, Y. R. Yang, A. S. Morse, B. D. O. Anderson, and P. N. Belhumeur. Rigidity, Computation, and Randomization in Network Localization. In *INFOCOM 2004. Twenty-third annual Joint Conference of the IEEE Computer and Communications Societies*, volume 4, pages 2673–2684. IEEE, 2004.

[70] E.D. Bolker and B. Roth. When is a Bipartite Graph a Rigid Framework. *Pacific Journal of Mathematics*, 90(1):27–44, 1980.

[71] Rainer Kümmerle, Giorgio Grisetti, Hauke Strasdat, Kurt Konolige, and Wolfram Burgard. g^2o: A General Framework for Graph Optimization. In *2011 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3607–3613. IEEE, 2011.

[72] Marco Crocco, Alessio Del Bue, Matteo Bustreo, and Vittorio Murino. A Closed Form Solution to the Microphone Position Self-Calibration Problem. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2597–2600. IEEE, 2012.

[73] Marco Crocco, Alessio Del Bue, and Vittorio Murino. A Bilinear Approach to the Position Self-Calibration of Multiple Sensors. *IEEE Transactions on Signal Processing*, 60(2):660–673, 2012.

[74] Yubin Kuang, Simon Burgess, Anna Torstensson, and Kalle Åström. A Complete Characterization and Solution to the Microphone Position Self-Calibration Problem. In *38th International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2013.

[75] Jorge Moré. The Levenberg-Marquardt Algorithm: Implementation and Theory. In *Numerical analysis*, pages 105–116. Springer, 1978.

[76] I. Gelfand and L. Tsetlin. The Non-Local Search Principle in Automatic Optimization Systems. *Proceedings of USSR Academy of Sciences*, 137:295, 1961.

[77] Johannes Wendeberg. Laufzeitbasierte Schallortung mit unbekannten Sender- und Empfängerpositionen. Master's thesis, University of Freiburg, 2009.

[78] M. Teschner, B. Heidelberger, M. Muller, and M. Gross. A Versatile and Robust Model for Geometrically Complex Deformable Solids. In *Computer Graphics International, 2004. Proceedings*, pages 312–319. IEEE, 2004.

[79] M. Gissler, M. Becker, and M. Teschner. Local Constraint Methods for Deformable Objects. In *Proc. of the 3rd Workshop in VR Interactions and Physical Simulation (VRIPHYS)*, pages 1–8, 2006.

[80] F. Vesely. *Computational Physics: An Introduction*. Springer, 2001.

[81] Carl T. Kelley. *Iterative Methods for Optimization*, volume 18. Society for Industrial and Applied Mathematics, 1987.

[82] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes: The Art of Scientific Computing.* Cambridge University Press, 2007.

[83] Fred James. Function Minimization. Technical report, Proceedings of the 1972 CERN Computing and Data Processing School. (Reprint), 1972.

[84] Klaus Schittkowski. Mathematische Grundlagen von Optimierungsverfahren. Mathematisches Institut, Universität Bayreuth, 1999.

[85] Christopher M. Bishop. *Pattern Recognition and Machine Learning*, volume 1. Springer New York, 2006.

[86] J.R. Shewchuk. An Introduction to the Conjugate Gradient Method Without the Agonizing Pain. Technical report, Carnegie Mellon University, Pittsburgh, PA, 1994.

[87] William W. Hager and Hongchao Zhang. A Survey of Nonlinear Conjugate Gradient Methods. *Pacific Journal of Optimization*, 2(1):35–58, 2006.

[88] Reeves Fletcher and Colin M. Reeves. Function Minimization by Conjugate Gradients. *The Computer Journal*, 7(2):149–154, 1964.

[89] William W Hager and Hongchao Zhang. A New Conjugate Gradient Method with Guaranteed Descent and an Efficient Line Search. *SIAM Journal on Optimization*, 16(1):170–192, 2005.

[90] Eldon Hansen. Global Optimization Using Interval Analysis – The Multi-Dimensional Case. *Numerische Mathematik*, 34(3):247–270, 1980.

[91] Åke Björck. *Numerical Methods for Least Squares Problems.* Society for Industrial and Applied Mathematics, 1996.

[92] Arnold Neumaier. Complete Search in Continuous Global Optimization and Constraint Satisfaction. *Acta Numerica*, 13(1):271–369, 2004.

[93] Arnold Neumaier, Oleg Shcherbina, Waltraud Huyer, and Tamás Vinkó. A Comparison of Complete Global Optimization Solvers. *Mathematical Programming*, 103(2):335–356, 2005.

[94] Oleg Shcherbina, Arnold Neumaier, Djamila Sam-Haroud, Xuan-Ha Vu, and Tuan-Viet Nguyen. Benchmarking Global Optimization and Constraint Satisfaction Codes. In *Global Optimization and Constraint Satisfaction*, pages 211–222. Springer, 2003.

[95] C.A. Floudas and C.E. Gounaris. A Review of Recent Advances in Global Optimization. *Journal of Global Optimization*, 45(1):3–38, 2009.

[96] Johannes Wendeberg and Christian Schindelhauer. Polynomial Time Approximation Algorithms for Localization Based on Unknown Signals. In *Proceedings of 8th International Symposium on Algorithms for Sensor Systems, Wireless Ad Hoc Networks and Autonomous Mobile Entities (ALGOSENSORS)*, pages 132–

143. Springer Berlin Heidelberg, 2012.

[97] Johannes Wendeberg. Simulation zur Lokalisation von Schallquellen mit variabler Mikrofonanzahl. Bachelor thesis, University of Freiburg, 2007.

[98] D. Kennedy. Some branch and bound techniques for nonlinear optimization. *Mathematical Programming*, 42(1):147–157, 1988.

[99] R. J. Dakin. A tree-search algorithm for mixed integer programming problems. *The Computer Journal*, 8(3):250–255, 1965.

[100] E. L. Lawler and D. E. Wood. Branch-And-Bound Methods: A Survey. *Operations Research*, 14(4):699–719, 1966.

[101] F. Gustafsson and F. Gunnarsson. Positioning using time-difference of arrival measurements. In *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing Proceedings (ICASSP)*, volume 6, pages VI–553. IEEE, 2003.

[102] P. Zou, Z. Huang, and J. Lu. Passive Stationary Target Positioning Using Adaptive Particle Filter with TDOA and FDOA Measurements. In *Communications, 2004 and the 5th International Symposium on Multi-Dimensional Mobile Communications Proceedings. The 2004 Joint Conference of the 10th Asia-Pacific Conference on*, volume 1, pages 453–456. IEEE, 2004.

[103] M. Kawanishi, R. Maruta, N. Ikoma, H. Kawano, and H. Maeda. Sound Target Tracking in 3D using Particle Filter with 4 Microphones. In *SICE, 2007 Annual Conference*, pages 1427–1430. IEEE, 2007.

[104] N. Ikoma, O. Tokunaga, H. Kawano, and H. Maeda. Tracking of 3D Sound Source Location by Particle Filter with TDOA and Signal Power Ratio. In *ICCAS-SICE, 2009*, pages 1374–1377. IEEE, 2009.

[105] J.A. Cho, H. Na, S. Kim, and C. Ahn. Moving-Target Tracking Based on Particle Filter with TDOA/FDOA Measurements. *ETRI Journal*, 34(2), 2012.

[106] Frank Dellaert, Dieter Fox, Wolfram Burgard, and Sebastian Thrun. Monte Carlo Localization for Mobile Robots. In *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, volume 2, pages 1322–1328. IEEE, 1999.

[107] S. Thrun, D. Fox, W. Burgard, and F. Dellaert. Robust Monte Carlo Localization for Mobile Robots. *Artificial Intelligence*, 128(1-2):99–141, 2000.

[108] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics*. MIT Press, 2005.

[109] Johannes Wendeberg, Jörg Müller, Christian Schindelhauer, and Wolfram Burgard. Robust Tracking of a Mobile Beacon using Time Differences of Arrival with Simultaneous Calibration of Receiver Positions. In *Proceedings of 2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2012.

[110] J.H. Kotecha and P.M. Djuric. Gaussian particle filtering. *IEEE Transactions on Signal Processing*, 51(10):2592 – 2601, 2003.

[111] J.S. Liu. Metropolized independent sampling with comparisons to rejection sampling and importance sampling. *Statistics and Computing*, 6, 1996.

[112] Jane Liu and Mike West. Combined Parameter and State Estimation in Simulation-Based Filtering. In Arnaud Doucet, Nando de Freitas, and Neil Gordon, editors, *Sequential Monte Carlo Methods in Practice*, chapter 10. Springer New York, 2001.

[113] Fabian Höflinger, Johannes Wendeberg, Rui Zhang, Manuel Bührer, Joachim Hoppe, Amir Bannoura, Leonhard M. Reindl, and Christian Schindelhauer. Acoustic Self-calibrating System for Indoor Smartphone Tracking (ASSIST). In *Proceedings of 2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2012.

[114] David L. Mills. Internet Time Synchronization: The Network Time Protocol. *IEEE Transactions on Communications*, 39(10):1482–1493, 1991.

[115] Julien Ridoux and Darryl Veitch. Principles of Robust Timing over the Internet. *Communications of the ACM*, 53(5):54–61, 2010.

[116] B. Sundararaman, U. Buy, and A.D. Kshemkalyani. Clock Synchronization for Wireless Sensor Networks: A Survey. *Ad Hoc Networks*, 3(3):281–323, 2005.

[117] A.F. Anta, M.A. Mosteiro, and C. Thraves. Deterministic Recurrent Communication and Synchronization in Restricted Sensor Networks. *INRIA RR*, 486277, 2010.

[118] M. Sippel, W. Kuntz, and L. Reindl. New Approach in Precise Laser Tracking. In *IEEE Instrumentation and Measurement Technology Conference Proceedings*, pages 446–451, 2008.

[119] M. Sippel. *Localization of Mobile Objects with Embedded Microsystems*. PhD thesis, University of Freiburg, 2010.

[120] Adam Smith, Hari Balakrishnan, Michel Goraczko, and Nissanka Priyantha. Tracking Moving Devices with the Cricket Location System. In *Proceedings of the 2nd International Conference on Mobile Systems, Applications, and Services*, pages 190–202. ACM, 2004.

[121] José Carlos Prieto, Antonio R. Jiménez, and Jorge I. Guevara. Subcentimeter-Accuracy Localization through Broadband Acoustic Transducers. In *IEEE International Symposium on Intelligent Signal Processing (WISP 2007)*, pages 1–6. IEEE, 2007.

[122] José Carlos Prieto, Antonio R. Jiménez, Jorge Guevara, Joao L. Ealo, Fernando Seco, Javier O. Roa, and Francisco Ramos. Performance Evaluation of 3D-LOCUS Advanced Acoustic LPS. *IEEE Transactions on Instrumentation and Measurement*, 58(8):2385–2395, 2009.

[123] J. Guevara, Antonio R. Jiménez, J José C. Prieto, and Francisco Seco. Auto-Localization Algorithm for Local Positioning Systems. *Ad Hoc Networks*, 10(6):1090–1100, 2012.

[124] Mike Addlesee, Rupert Curwen, Steve Hodges, Joe Newman, Pete Steggles, Andy Ward, and Andy Hopper. Implementing a sentient computing system. *Computer*, 34(8):50–56, 2001.

[125] Oliver J Woodman and Robert K Harle. Concurrent Scheduling in the Active Bat Location System. In *8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, pages 431–437. IEEE, 2010.

[126] B. Xu, R. Yu, G. Sun, and Z. Yang. Whistle: Synchronization-Free TDOA for Localization. In *2011 31st International Conference on Distributed Computing Systems (ICDCS)*, pages 760–769. IEEE, 2011.

[127] G. Borriello, A. Liu, T. Offer, C. Palistrant, and R. Sharp. WALRUS: Wireless Acoustic Location with Room-Level Resolution using Ultrasound. In *Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services*, pages 191–203. ACM, 2005.

[128] Jeffrey Hightower and Gaetano Borriello. Location Systems for Ubiquitous Computing. *Computer*, 34(8):57–66, 2001.

[129] Peng Zhang and Hao Liu. An Ultra-Wide Band System with Chirp Spread Spectrum Transmission Technique. In *6th International Conference on ITS Telecommunications Proceedings*, pages 294–297. IEEE, 2006.

[130] Thomas Stoll. *Dezentral gesteuerter Aufbau von Stetigförderern mittels autonomer Materialflusselemente*. PhD thesis, Karlsruhe Institute of Technology (KIT), 2012.

[131] Joan Bordoy, Patrick Hornecker, Fabian Höflinger, Johannes Wendeberg, Rui Zhang, Christian Schindelhauer, and Leonhard Reindl. Robust Tracking of a Mobile Receiver using Unsynchronized Time Differences of Arrival. In *Proceedings of 2013 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2013.