*Forschungsberichte aus dem
Institut für Nachrichtentechnik des
Karlsruher Instituts für Technologie*

Noha El Gemayel

Smart Passive Localization Using Time Difference of Arrival

Band 36

Druck: Frick Kreativbüro & Onlinedruckerei e.K.
Brühlstraße 6
86381 Krumbach

ISSN: 1433-3821

Forschungsberichte aus dem Institut für Nachrichtentechnik des Karlsruher Instituts für Technologie

Herausgeber: Prof. Dr. rer. nat. Friedrich Jondral

Band 1	Marcel Kohl **Simulationsmodelle für die Bewertung von Satellitenübertragungsstrecken im 20/30 GHz Bereich**
Band 2	Christoph Delfs **Zeit-Frequenz-Signalanalyse: Lineare und quadratische Verfahren sowie vergleichende Untersuchungen zur Klassifikation von Klaviertönen**
Band 3	Gunnar Wetzker **Maximum-Likelihood Akquisition von Direct Sequence Spread-Spectrum Signalen**
Band 4	Anne Wiesler **Parametergesteuertes Software Radio für Mobilfunksysteme**
Band 5	Karl Lütjen **Systeme und Verfahren für strukturelle Musteranalysen mit Produktionsnetzen**
Band 6	Ralf Machauer **Multicode-Detektion im UMTS**
Band 7	Gunther M. A. Sessler **Schnell konvergierender Polynomial Expansion Multiuser Detektor mit niedriger Komplexität**
Band 8	Henrik Schober **Breitbandige OFDM Funkübertragung bei hohen Teilnehmergeschwindigkeiten**
Band 9	Arnd-Ragnar Rhiemeier **Modulares Software Defined Radio**
Band 10	Mustafa Mengüç Öner **Air Interface Identification for Software Radio Systems**

Forschungsberichte aus dem Institut für Nachrichtentechnik
des Karlsruher Instituts für Technologie
Herausgeber: Prof. Dr. rer. nat. Friedrich Jondral

Band 11	Fatih Çapar **Dynamische Spektrumverwaltung und elektronische Echtzeitvermarktung von Funkspektren in Hotspotnetzen**
Band 12	Ihan Martoyo **Frequency Domain Equalization in CDMA Detection**
Band 13	Timo Weiß **OFDM-basiertes Spectrum Pooling**
Band 14	Wojciech Kuropatwiński-Kaiser **MIMO-Demonstrator basierend auf GSM-Komponenten**
Band 15	Piotr Rykaczewski **Quadraturempfänger für Software Defined Radios: Kompensation von Gleichlauffehlern**
Band 16	Michael Eisenacher **Optimierung von Ultra-Wideband-Signalen (UWB)**
Band 17	Clemens Klöck **Auction-based Medium Access Control**
Band 18	Martin Henkel **Architektur eines DRM-Empfängers und Basisbandalgorithmen zur Frequenzakquisition und Kanalschätzung**
Band 19	Stefan Edinger **Mehrträgerverfahren mit dynamisch-adaptiver Modulation zur unterbrechungsfreien Datenübertragung in Störfällen**
Band 20	Volker Blaschke **Multiband Cognitive Radio-Systeme**

Forschungsberichte aus dem Institut für Nachrichtentechnik des Karlsruher Instituts für Technologie

Herausgeber: Prof. Dr. rer. nat. Friedrich Jondral

Band 21	Ulrich Berthold **Dynamic Spectrum Access using OFDM-based Overlay Systems**
Band 22	Sinja Brandes **Suppression of Mutual Interference in OFDM-based Overlay Systems**
Band 23	Christian Körner **Cognitive Radio – Kanalsegmentierung und Schätzung von Periodizitäten**
Band 24	Tobias Renk **Cooperative Communications: Network Design and Incremental Relaying**
Band 25	Dennis Burgkhardt **Dynamische Reallokation von spektralen Ressourcen in einem hierarchischen Auktionssystem**
Band 26	Stefan Nagel **Portable Waveform Development for Software Defined Radios**
Band 27	Hans-Ulrich Dehner **Interferenzuntersuchungen für inkohärente Multiband Ultra-Breitband (UWB) Übertragung**
Band 28	Maximilian Hauske **Signalverarbeitung für optoelektronische Sensoren**
Band 29	Jens Elsner **Interference Mitigation in Frequency Hopping Ad Hoc Networks**
Band 30	Georg Vallant **Modellbasierte Entzerrung von Analog/Digital-Wandler-Systemen**

Forschungsberichte aus dem Institut für Nachrichtentechnik des Karlsruher Instituts für Technologie

Herausgeber: Prof. Dr. rer. nat. Friedrich Jondral

Band 31	Martin Braun **OFDM Radar Algorithms in Mobile Communication Networks**
Band 32	Michael Sebastian Mühlhaus **Automatische Modulationsartenerkennung in MIMO-Systemen**
Band 33	Michael Schwall **Turbo-Entzerrung: Implementierungsaspekte für Software Defined Radios**
Band 34	Ralph Tanbourgi **Diversity Combining under Interference Correlation in Wireless Networks**
Band 35	Florian Engels **Multidimensional Frequency Estimation with Applications in Automotive Radar**
Band 36	Noha El Gemayel **Smart Passive Localization Using Time Difference of Arrival**

Vorwort des Herausgebers

Aufgrund der knappen Ressourcen für die Mobilfunkübertragung wird intensiv nach Möglichkeiten zur Steigerung der Effizienz bei der Nutzung des elektromagnetischen Spektrums gesucht. Neben der Einführung von Mehrantennentechnologien, der Installation von Pico- bzw. Femto-Zellen oder dem Ausweichen auf Mikrowellen sind Cognitive Radios Gegenstand aktueller Forschungsarbeiten. Für den Einsatz von Cognitive Radios wäre, insbesondere in Ballungsgebieten, die Kenntnis der elektromagnetischen Umgebung äußerst vorteilhaft. Dazu können Interferenzkarten genutzt werden, die Auskunft darüber geben, welche Störungen des Funkverkehrs an welchen Orten zu erwarten sind. Bevor Interferenzkarten erfolgreich erstellt und aktualisiert werden können, müssen jedoch einige herausfordernde Fragen beantwortet werden, unter anderem die nach der Ortung von Quellen elektromagnetischer Strahlung, die den Mobilfunkverkehr stören können.

Nun ist die Sender- oder, im vorliegenden Fall, die Störerortung ein Thema, an dem seit mehr als hundert Jahren gearbeitet wird. Dabei ist zu beachten, dass Sensoren und Systeme für die Funkortung in der Regel teuer sind und ihre Verlässlichkeit, gerade im für den Mobilfunk vorherrschenden Mehrwegeausbreitungsscenario, oft zweifelhaft bleibt. Besonders kostengünstig erscheint die Ortung auf der Basis von Laufzeitdifferenzmessungen (Time Difference of Arrival, TDoA), da dabei der einzelne Sensor, anders als ein Peiler, der den

Einfallswinkel elektromagnetischer Wellen bestimmt, kein Mehrantennensystem und die damit verbundenen parallelen Empfangszüge benötigt.

Die in der von Noha El Gemayel vorgelegten Dissertation aufgegriffene Aufgabenstellung dreht sich daher im Wesentlichen um die Themen

- Zusammenstellung und Bewertung bekannter Algorithmen zur Laufzeitdifferenzmessung sowie zur TDoA-Ortung in additivem weißen Gaußschen Rauschen und in Mehrwegeausbreitungsszenarien
- Design, Optimierung und Test der Algorithmen für ein geeignetes adaptives (smartes) TDoA-Ortungssystem
- Konzeption preiswerter TDoA-Sensoren und Begleitung von deren Realisierung auf Software Defined Radio Komponenten
- Planung eines Sensorfelds und Unterstützung seines Aufbaus in realer Umgebung
- Durchführung von Experimenten zur Verifikation und Bewertung der Leistungsfähigkeit des adaptiven TDoA-Ortungssystems

Die Einsatzmöglichkeiten für das adaptive TDoA-Ortungssystem sind vielfältig. Ein erster Prototyp bestand seine Bewährungsprobe bereits beim Einsatz im vom Bundesministerium für Bildung und Forschung (BMBF) geförderten Projekt Intelligente Sensoren für die Digitale Dividende [1], das gemeinsam von der LStelcom AG (Lichtenau in Baden) und dem KIT-Institut für Nachrichtentechnik bearbeitet wurde.

Karlsruhe, im Mai 2016
Friedrich Jondral

[1] Förderkennzeichen 01BU1031

Smart Passive Localization Using Time Difference of Arrival

Zur Erlangung des akademischen Grades eines

DOKTOR-INGENIEURS

der Fakultät für
Elektrotechnik und Informationstechnik
des Karlsruher Instituts für Technologie

genehmigte

DISSERTATION

von

Dipl.-Ing. Noha El Gemayel

geb. in

Kairo, Ägypten

Tag der mündlichen Prüfung:	17.05.2016
Hauptreferent:	Prof. Dr. rer. nat. Friedrich K. Jondral
Korreferent:	Prof. Dr.-Ing. habil. Gert F. Trommer

Acknowledgements

I came to Germany when I was 18 to study Electrical Engineering in Karlsruhe. Back then, the thought of doing a PhD had never crossed my mind. Some encounters and lessons in life have led me to believe that this was the right path for me. I would like to take the opportunity to express my gratitude to those involved in guiding me to this path and those who supported and encouraged me throughout this journey.

First, I thank my parents, Bella and Magdy, for their unconditional love and their sacrifices that opened up all the right doors for me. I would not be here if it wasn't for you. Thank you for agreeing to let me leave at such an early age and for always having my back.

Since this is chronologically accurate, I want to thank the German Academic Exchange Service for granting me the scholarship to study in Germany and for making my dream come true.

Looking back at my first discussion with Prof. Jondral about my career plans, he was the one who convinced me that the PhD is the right choice for me and who first told me that he believed in me. I am very thankful for his trust, his guidance as well as many lessons in life that I will never forget. His unique leadership encouraged his PhD students to develop individually, knowing they have his support.

Prof. Jondral managed to create a very friendly and respectful environment among all institute co-workers. I would like to thank all my colleagues and co-workers, the current and the former ones, for their help, for insightful scientific discussions as well as for many good times. My special thanks go to two colleagues, who played an important role during my time at the institute: Michael Mühlhaus, my neighbor in office 007, for his moral support and many essential (and some very funny) discussions and Dr.-Ing. Holger Jäkel for helping me develop my scientific working method, for proof-reading my thesis as well as for interesting and insightful talks about life and, of course, discipline.

Last but not least, I thank my family and friends for encouraging me and believing in me during this time. My utmost gratitude goes to Andi, for his love, his patience, his generosity and for being the best partner and friend I could hope for.

Dedicated to
my parents, Bella and Magdy

Zusammenfassung

Die Entwicklung kostengünstiger Komponenten der drahtlosen Funkkommunikation ermöglicht eine detaillierte Kenntnis über die Position verschiedener Übertragungsteilnehmer. Während alltägliche Anwendungen wie Sicherheits- und Notfallanwendungen sowie standortbezogene Dienste allgegenwärtig sind, existiert ein weiterer, eher unbekannter Nutzer solcher Information: Frequenzregulierer.

Das schnelle Wachsen drahtloser Netzwerke und die damit verbundene Forderung nach immer größeren Signalbandbreiten und höheren Datenraten erfordern eine Verlagerung in der Philosophie der Spektrumsregulierung von einer festen Lizenzierung zu einer flexiblen und dynamischen Zuordnung der Frequenzbereiche. Diese Flexibilisierung und Dynamik erfordern sogenannte Frequenzbelegungskarten, die Auskunft darüber geben, wann und wo eine Frequenz verwendet wird. Mit Hilfe dieser Karten können unlizenzierte Nutzer gefunden werden und unbenutzte Frequenzen neu zugeteilt werden, sofern keine Primärnutzer gestört werden.

Eine Realisierungsmöglichkeit eines solchen Systems besteht in der Verwendung eines Sensornetzes, das die Daten für die Belegungskarten bereitstellt. Da die Vermeidung von Primärnutzerstörung essentiell ist, müssen die Sensoren, vor Allem in städtischen Gebieten, eine enge Anordnung mit hoher Abdeckung besitzen. Die hohe Anzahl an Sensoren erfordert einen kostengünstigen

Sensor, der in der Lage ist, Sender zu erkennen und zu lokalisieren. Solch ein System kann mit Hilfe kostengünstiger softwarebasierter Empfänger realisiert werden, mit denen Signale detektiert und geortet werden sollen. Die Ortung unbekannter Sender ist das Thema der vorliegenden Arbeit.

In dieser Dissertation wird ein passives Ortungssystem auf Basis von Laufzeitdifferenzen entworfen mit dem Ziel, die Positionsinformation für die Erstellung der Frequenzbelegungskarten bereit zu stellen. Dafür werden zwei Ausbreitungsszenarien separat analysiert: der additive weiße Rauschkanal und der Mehrwege- kanal. Für jedes Szenario werden passende Methoden und Algorithmen definiert, neue und verbesserte Algorithmen präsentiert sowie theoretische Schätzgrenzen hergeleitet. Die Algorithmen werden mit Hilfe von Simulationen verglichen und analysiert. Die Analyse der Ergebnisse führt zum Entwurf eines kompletten passiven Ortungssystems, das die Zusammenhänge zwischen den verschiedenen Rechenschritten herstellt.

Zusätzlich wird zur Verifikation der Simulationsergebnisse ein Prototyp-System aufgebaut, mit dem Messungen an einem realen System durchgeführt werden können. Die Ergebnisse demonstrieren die Funktionalität eines kostengünstigen softwarebasierten Sensorsystems als passives Ortungssystem in einem realen Feldversuch.

Abstract

With the advent of cheap and simple wireless communication components, including terminals and infrastructure, the possibility of locating communication participants opened up a new field of applications based on location information. Apart from daily live applications, including emergency response, security as well as the well known location-based services, a very important user group of the location information, albeit less known, are frequency regulators.

With the vast expansion of wireless networks and the need for larger bandwidths and higher data rates, the philosophy of frequency regulators is shifting from exclusive licensing to a more dynamic and flexible frequency allocation, enabling a more efficient usage of the limited spectral resource. Such a task requires so called spectrum allocation maps, offering information about when and where a frequency is being used. Using this data, unlicensed users can be found and unused frequencies, so called white spaces, can be reallocated, provided that primary users do not experience any interference.

To realize such a frequency monitoring system, sensors gathering data for the construction of the allocation maps must cover large areas, including urban ones. Since reliability is essential to avoid interference, the density of the sensors must be high. Thus, a system based on low-cost hardware that can fulfill the task of identifying and locating unknown wireless transmitters is targeted.

This system can be realized using low-cost software-defined radios that can receive and detect signals as well as estimate the position of their sources. The latter is the research topic of this thesis.

In this work, a smart passive localization system using time difference of arrival (TDoA) measurements is designed and analyzed with the goal of providing the position information for the construction of frequency allocation maps. Two propagation channels are treated separately: the additive white Gaussian noise scenario and the multipath propagation scenario. In each scenario, methods suitable for the corresponding framework are depicted and described, novel algorithms offering an improved estimation accuracy are presented and theoretical bounds are derived. The chosen methods are implemented and their performances are compared to each other as well as to the theoretical bounds. Based on the analysis, a system design is proposed, linking the different elements of the localization system.

Further on, to verify the results, a low-cost system setup is presented and its measurement results are analyzed. Overall, the results demonstrate the applicability of the designed TDoA system as a low-cost passive localization system in real scenarios.

Contents

1 Introduction

Localization describes the process of determining the position of a point of interest within a reference coordinate system. This task has existed for many years and has used different techniques, primarily based on signals transmitted from or to the point of interest, which can be optical, electromagnetic, or acoustic. Acquiring the position of an object finds relevance in many applications of our daily lives: apart from the established and long existing navigation systems for aeronautical, nautical, and land navigation, the emergence of cellular networks and the associated location-based and context-aware services for mobile users have revived research in the field of localization. Location-based services include basic functions such as assistance, security, emergency managing as well as daily life tasks such as retail sales. With the enormous amount of applications using the information about the position of a user, new localization technologies are emerging very fast.

A very important, albeit less known, user of position information are frequency regulators, aiming at obtaining the position of various radio transmitters to enable the construction of frequency allocation maps. These maps should answer the question about which frequencies are used when and where. With this information, unlicensed transmitters can be found, statistical data can be gathered, and a more efficient usage of the spectrum resource can be planned. A lot of otherwise wasted resources can be saved not only by optimizing the usage through preplanned allocation, but by allowing flexible and dynamic frequency allocation. This topic has gained large attention especially since the emergence of white spaces as a result of digitizing TV-broadcast. The future of spectral allocation now includes the keywords *dynamic spectrum access* and *cognitive radios*.

In 2010, a joint research project was launched between the Communications Engineering Lab of the Karlsruhe Institute of Technology (KIT), together with *LS telcom AG* called *Intelligent Support System for the Digital Dividend*. The project, funded by the Federal Ministry of Education and Research, aimed at developing and designing a monitoring system that serves as the supporting data basis for dynamic frequency allocation, primarily in the abandoned UHF/VHF-band [1], the so called digital dividend. Parts of the research presented here are the result of this project.

The core tasks of the support system are primary user signal detection and signal source localization, the latter aiming at finding the position of licensed and unlicensed users. This work deals with the signal source localization. The targeted support system is based on the information gathered from sensors distributed over large regions. Since reliability is required to avoid interference, the density of the installed sensors must be high to ensure coverage, especially in dense urban cities. Therefore, low-cost hardware must be targeted to enable the setup of such a system. One promising localization technology that can be realized using low-cost hardware is Time Difference of Arrival (TDoA).

TDoA dates as far back as World War I, where it was used to find gun positions by tracking their acoustic signals [2]. Later on, the method was used by various navigation systems such as Decca, LORAN-C and now, GPS. Depending on the signal characteristics and the associated propagation models, the same technique requires various algorithms and estimation methods.

This work aims at developing and designing a smart localization system, based on low-cost hardware components that can accurately estimate the position of various signal sources in large spectrum regions using TDoA. The work aims at depicting and refining existing algorithms that can be applied in the described scenario, developing new algorithms for unsolved problems, establishing and analyzing connections between the different steps of the localization process as well as testing the designed system using low-cost hardware.

[1] VHF lie between 30 MHz and 300 MHz, UHF between 300 MHz and 3 GHz.

[2] Frank Parker Stockbridge, *How Far Off Is That German Gun? How 63 German guns were located by sound waves alone in a single day*, Popular Science monthly, December 1918, page 39

1.1 Outline of Work

The research focuses on three important pillars of a localization system: fundamental limits, algorithm design as well as experimental analysis.

Chapter 2 introduces the most important localization techniques. By addressing the localization scenario in detail, the choice of Time Difference of Arrival as the main localization technique is justified. The chapter proceeds by laying out the basic mathematical background of the TDoA problem as well as describing the challenges associated with TDoA to lay the ground for the following two chapters. The problem of estimating a position using TDoA measurements is a two-step problem. In the first step, the time delay differences are estimated using received signals gathered from spatially distributed sensors. In the second step, the estimated TDoAs are used, together with the known positions of the sensors, to yield a position estimate of the transmitter. The following chapters are therefore separated in a section discussing the time delay estimation, and one discussing the position estimation.

Chapter 3 handles the estimation problem in AWGN channels, covering the basic methods for estimating the delay differences and the accommodating interpolation techniques as well as attainable theoretical bounds. By analyzing the estimation error, the transition to the subsequent section is made. There, position estimation techniques using the estimated TDoAs are presented, alongside with algorithms to stabilize and enhance the estimation accuracy using sequential measurements. Apart from the signal processing and estimation algorithms, different geometrical aspects, like the positioning of the sensors, play an important role. These are discussed thoroughly, together with the associated estimation bounds. The presented algorithms are verified and compared using various simulation scenarios.

The real challenges of the estimation problem arise when the signals received by the system are propagated through multipath channels, a scenario that is found primarily in urban environments. Chapter 4 discusses the resulting problems of estimating time delays in multipath scenarios and proposes suitable methods, based on the maximum likelihood solution of the problem. Simulation results are discussed and compared to the theoretical bound. Due to the complexity of this problem, further error mitigation techniques need to be applied after the delay estimation, since the estimation result is not always unbiased. Therefore, methods based on various approaches are presented as a solution to enhance the position estimation accuracy. Simulation results verify and compare the presented methods.

To round off the study, sensors fulfilling the requirements of a TDoA measurement system were designed and set up on the university campus. Chapter 5 presents the sensor components, the geometrical set up as well as other measurement configuration details. Measurement results using the aforementioned algorithms are demonstrated and important aspects that must be additionally considered in real scenarios are addressed. Chapter 6 sums up the work and its contributions.

2 Localization

The term localization refers to the process of seeking the information about the physical position of an object of interest, which is usually called a *point of interest* or a *node*, by one or more measuring entities which are referred to as *anchors*. Different applications of this information in our daily lives include navigation, emergency calls and the recently emerged location-based services. Special applications include surveillance, both for civil and military purposes as well as for frequency regulation. Depending on the available measurement, several techniques can be applied to obtain an estimate of the desired position information. The accuracy of localization is determined by the following parameters: (i) the available hardware, resulting in an accurate or less accurate measured quantity (i.e., signal with time stamps), (ii) the scenario consisting of the signal used for the estimation, its dynamics as well as the propagation channel, (iii) additional requirements on computational complexity or on latency, limiting the choice of possible algorithms and (iv) the geometrical setup of the anchors and node.

In this chapter, a short review on some available localization techniques is presented. After introducing the main scenario, the choice of Time Difference of Arrival as the main technique for the localization system is justified. The main challenges facing Time Difference of Arrival (TDoA) are listed and described as a basic concept of the subsequent chapters.

2.1 Localization Techniques

A localization can be carried out by transmitting a signal to the point of interest from one or more anchors or by receiving a signal transmitted by the point of interest by one or more anchors. Both problems are equivalent in the mathematical sense. Since this work is concerned with monitoring frequency regions and localizing various transmitters in these regions, the latter scenario applies, meaning that here, the anchors receive a signal transmitted by the node. In accordance with the described localization scenario, throughout this work, the point of interest will be referred to as the *transmitter* and the anchors will be referred to as the *sensors*.

Signals received by the sensors are used to extract parameters such as delay, signal power, or frequency. These parameters can be transformed to physical parameters. The physical nature of the measured parameters in localization scenarios determines the geometrical problem. Whether it is a direction, a distance, a distance ratio or a distance difference defines loci of possible position points. Intersections of these loci result in the position estimate.

In [1] and [2], different techniques are described and compared. Here, the most common ones are picked and presented. Since the focus is on two-dimensional position determination, all basics as well as the following algorithms will be described for that scenario.

- Angle of Arrival (AoA): The transmit signal is received by an antenna array. By calculating the relative phase shift of the received signal at the different array elements, the signal source direction can be determined, leading to a *triangulation* problem. Each direction describes a line, the so called line of bearing, as locus. The intersection of two such lines yields the node position [3].

- Time of Arrival (ToA): Assuming a known transmit time as well as a known transmitted signal (i.e., synchronization between the sensors and the transmitter), the information about the distance between a sensor and the transmitter can be extracted by correlating the received signal with the known one. Referred to as *lateration* or *trilateration*, each distance describes a circle. The intersection of three circles yields a unique position [4].

- TDoA: In passive scenarios, there is no coordination between sensors and transmitter. Assuming that the sensors have synchronized clocks, a TDoA can be obtained by cross-correlating two signals received at two separate sensors. One TDoA represents a hyperbola, resulting in a *multilateration* problem. The intersection of three hyperbolas results in the node position [5]. Oftentimes, two hyperbolas (i.e., three sensors) are sufficient to estimate the transmitter position since the second intersection point of the hyperbolas often lies far outside the coverage area of the sensors.

- Received Signal Strength (RSS): Using available propagation models (i.e., path loss exponents) as well as a known transmit signal power, the received power represents the distance between a sensor and the transmitter, again yielding the same distance circles as in ToA. Apart from the range-based lateration, the RSS measurements can be used for fingerprinting, as was described in Chapter 11 of [2].
- Received Signal Strength Difference (RSSD): For an unknown transmit power, the power ratio can be used by the sensors to obtain the so called RSSDs, which also describe circles but with different relations. Three circles intersect in one point, yielding the desired position [6].
- Frequency Difference of Arrival (FDoA): If relative movement between the transmitter and the sensors exists, a Doppler shift can be measured from the received signals. Subtracting the frequencies of two received signals at two sensors yields a so called FDoA, describing curves that can be similar to the TDoA hyperbolas and needing at least two such curves to intersect in one point [7].

2.2 Localization Scenario

Resulting from the motivation behind the thesis, the general localization scenario which will be the basic scenario for the positioning system is shortly described here. Since this work aims at finding licensed and unlicensed transmitters for frequency management and regulatory reasons, the transmitted signal is assumed to be unknown, having an arbitrary bandwidth that is considered to be between 100 kHz and 2 MHz. These bandwidths have been chosen because they cover the lower bound for established standards for outdoor communication systems (e.g., GSM, UMTS, DVB, LTE,....). Higher bandwidths will lead to better results, making this scenario a worse case. On the other hand, the carrier frequencies are assumed to be within the UHF/VHF-frequencies, since this was the original purpose of the research.

Since we are concerned with finding unlicensed transmitters or with gathering information for a better spectral allocation, stationary or slowly moving transmitters and hence, transmitters staying in one region for a longer time period, are of interest.

The part where the signal is detected and/or separated from other signals exceeds the volume of this work and is assumed to be given. Thus, the times and frequencies used by the transmitter are known to the localization system without knowing the actual transmitted signal. The task is to use this extracted signal to find the position of its source.

From the aforementioned localization techniques, only AoA, RSSD, TDoA and FDoA can passively extract the position information. AoA needs expensive antenna arrays and is highly sensitive to propagation phenomena such as reflection and scattering. RSSD has a very instable geometry and would not be a reliable option and FDoA assumes a moving transmitter (or sensor) which is not guaranteed and is not accurate for slowly moving transmitters. Therefore, the best option for the described scenario is Time Difference of Arrival.

The transmitted signal can be received and recorded by multiple sensors representing the hardware components of the positioning system. The following requirements have to hold for these components:

- The sensors must be able to receive wireless signals with variable carrier frequencies and bandwidths.

- The sensor clocks must be accurately synchronized among each other.

- A central unit should gather the received data from all the sensors. If the positioning should be done in real time, the sensors must be connected to this center component to enable a timely data transmission.

- The position of the sensors has to be accurately known.

One possible realization of these components can be based on software defined radios that can operate on different carrier frequencies and with different bandwidths. To provide these components with synchronized clocks and with accurate position measurements, a GPS-module can be used. The details of the system used for validation of the algorithms will be described in Chapter 5. For now, it is only important to have an idea of the available information and system components to be able to understand the resulting challenges.

2.3 Position Estimation Using Time Difference of Arrival (TDoA)

Localization using TDoA is usually done by executing the following two steps:

1. **Estimating TDoAs from sensor pairs using their received signals.**
2. **Estimating the position of the transmitter using the estimated TDoAs.**

This method is often referred to as two-step-localization. Alternatively, one can directly estimate the position of a transmitter using received signals from multiple sensors and their known positions. This alternative method will be presented in section 4.2.4. Throughout the thesis, the use of the two-step-method will be preferred due to its convenience. It will be scrutinized next.

A TDoA localization scenario

Figure 2.1 shows an example of a Time Difference of Arrival positioning problem in the two-dimensional case. A transmitter TX positioned at $\boldsymbol{x} = [x_T, y_T]^T$ transmits its signal $s(t)$ at starting time t_0. Three sensor nodes $\mathrm{SN}_1, \mathrm{SN}_2, \mathrm{SN}_3$ with known coordinates $\mathbf{p_1} = [x_1, y_1]^T, \mathbf{p_2} = [x_2, y_2]^T, \mathbf{p_3} = [x_3, y_3]^T$ observe the signal of interest and record it.

The TDoAs between these sensors are

$$\Delta\tau_{i,j} = \tau_i - \tau_j = \frac{d_i - d_j}{c} \tag{2.1}$$

with c being the propagation speed and d_i being the unknown distance between sensor i and the transmitter. Each of these TDoAs describes a hyperbola with two sensors as its foci, containing all possible points on the x-y plane that fulfill the TDoA equation (2.1) (i.e., for each point on the hyperbola, the difference of the distances between the point and the foci is always equal $c \cdot \Delta\tau_{i,j}$). TDoAs with ambiguous signs define two hyperbola branches. In the described localization scenario, the TDoA sign is distinct. Consequently, the term hyperbola will be used throughout the thesis to describe the proper hyperbola branch. The point of intersection of at least three hyperbolas is the transmitter position. In some scenarios, two hyperbolas can be sufficient, assuming that the second intersection point can be excluded using *a priori* information or using logic (e.g., if the second intersection point is outside the coverage area of the sensors).

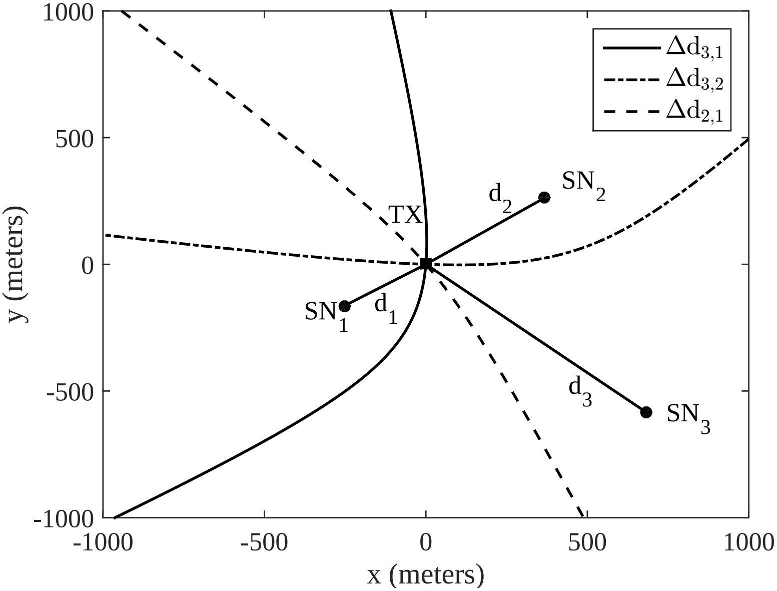


Figure 2.1: A TDoA scenario consisting of three sensors and one transmitter

The remaining question is how to obtain TDoAs from signals received at sensors $\mathrm{SN}_1, \mathrm{SN}_2, \mathrm{SN}_3$. Figure 2.2 shows an example of three received signals in an ideal channel scenario. The signals are observed here over a time window of $10\mu s$ with synchronized clocks at the sensors. In the figure, the TDoAs that generated the hyperbolas of Figure 2.1 are shown. These TDoAs can be observed in the peak positions of the cross-correlations defined as [1] [1]

$$\hat{C}_{i,j}(\tau) = \int_0^{T_o} r_i(t) r_j^*(t-\tau)dt, \tag{2.2}$$

as can be seen in Figure 2.3.Overall, $\frac{N(N-1)}{2}$ TDoAs can be obtained from N available signals (here: $N = 3$). But since, for example

$$\Delta\tau_{3,1} = \tau_3 - \tau_1 = \tau_3 - \tau_2 - \tau_1 + \tau_2 = \Delta\tau_{3,2} + \Delta\tau_{2,1}, \tag{2.3}$$

[1] This definition assumes the signal energies to be normalized to one.

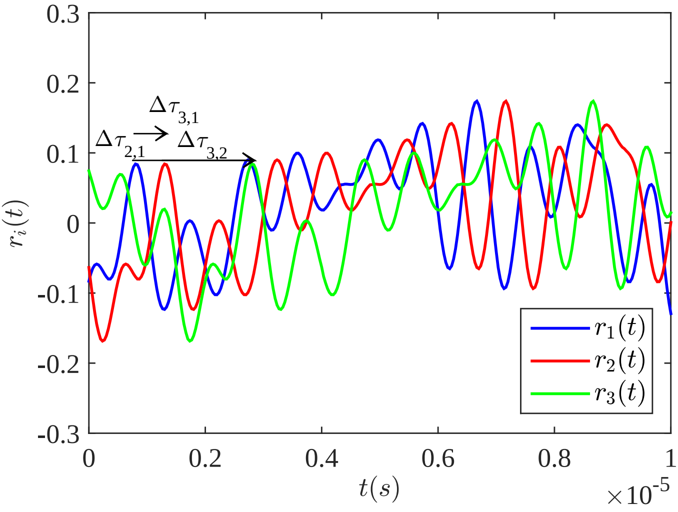


Figure 2.2: Received signal of interest at the TDoA sensors

a set of $N-1$ TDoAs includes all independent time differences, the remaining TDoAs are redundant and can be obtained from this set. In case of erroneous TDoA estimates, this statement does not hold since additional TDoA measurements can reduce the uncertainty regions. This processing gain resulting from considering all system dependencies comes with a much higher computational cost due to the additional cross-correlations that need to be calculated as well as the additional hyperbolic equations. Therefore, the trade-off is usually taken by defining one sensor as reference sensor and estimating $N-1$ TDoAs using this sensor as their reference.

Fig. 2.3 shows the cross-correlations between the received sensor signals. The correlation peaks at the true TDoAs.

2.4 TDoA Challenges

The previous section has shown a simple TDoA system setup and the important operations linked with the position estimation:

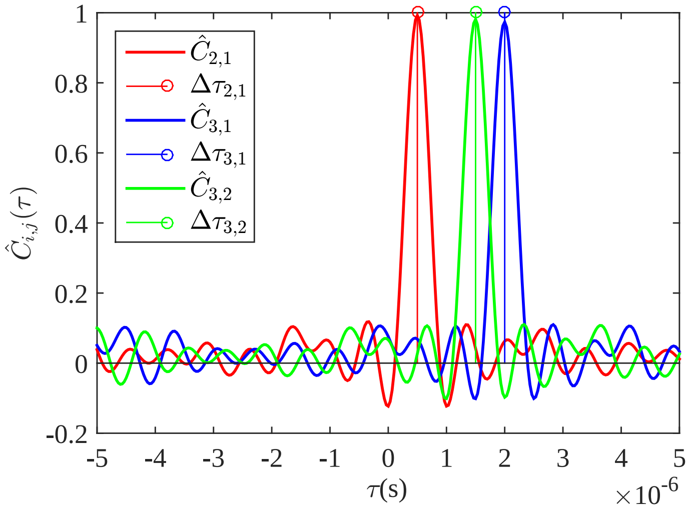


Figure 2.3: Cross-correlations of the signals in Figure 2.1

- Estimating the TDoAs from the signals by cross-correlating signal pairs and detecting their correlation peaks.
- Solving the hyperbola equation system to find the point of intersection of the hyperbolas.

In real scenarios and non-ideal channels, these operations face a number of challenges that will be presented next and that will be the topic of extensive analysis and discussion throughout the thesis. The main challenges of a TDoA system are:

- AWGN channels: In a best case scenario, a signal impinging on the sensors in the described setup would only be corrupted by additive white Gaussian noise. Since we are assuming a known carrier frequency, the signals are processed in the baseband and can be modeled as

  $$r_i(t) = s(t - \tau_i) + w_i(t), \quad w_i \in \mathcal{CN}\left(0, \sigma_i^2\right), \tag{2.4}$$

  whereas τ_i is the propagation delay and w_i being the circular complex zero mean additive white Gaussian noise term. Due to this additive error, the correlation peak shifts around the true delay value, introducing error to the estimated TDoAs.

- Discrete signals: (2.2) shows the cross-correlation of two continuous signals. In the setup described in Section 2.2, the sensors sample a signal of interest. The cross-correlation has to be performed in the discrete time domain, thus introducing quantization errors with respect to $\Delta\tau$. These errors can be reduced by applying interpolation techniques, as will be shown in section 3.1.

- Geometrical setup: Depending on the locations of the sensors as well as on the transmitter location, a TDoA estimation error can affect the position estimate very differently. While some setups are robust to large errors, others lead to complex hyperbola constellations, resulting in large errors or algorithm failures. This will be presented in section 3.2.

- Multipath propagation: Time delay estimation is a much more complex problem in multipath channels, where the received signals are modeled as

$$r_i(t) = \sum_{p=1}^{P_i} \alpha_i^{(p)} s\left(t - \tau_i^{(p)}\right) + w_i(t) \tag{2.5}$$

  with P_i being the number of received paths at sensor i, $\alpha_i^{(p)}$ being the gain of path p of sensor i and $\tau_i^{(p)}$ the corresponding path delay. Cross-correlating signals containing multiple, delayed versions of the same signal results in multiple correlation peaks. Depending on the signal bandwidth as well as on the channel, distinct paths can be resolved, resulting in estimates with accuracies similar to the case of Additive White Gaussian Noise (AWGN) channels. In case of unresolvable paths, the simple cross-correlation and peak detection would result in biased TDoAs. Different methods that deal with this problem will be introduced and analyzed section 4.1.

- Non-Line-of-Sight (NLoS): Although NLoS propagation is part of the multipath problem, it is usually treated separately in localization systems. There, it is considered as the problem occurring due to biased TDoA estimates, meaning that it is assumed that the time delay estimation has not been successful in terms of unbiased estimation. This can happen either because the direct signal path is too weak to be detected, or because multiple signal paths largely overlap at the receiver and can not be resolved.

> Based on the assumption of biased estimates, different methods were developed to identify and/or mitigate the NLoS errors. Section 4.2 shows a number of methods that were implemented for the described system setup.

The following two chapters present algorithms that were developed to conquer these challenges and analyzes the estimation performance bounds due to the introduced errors.

3

TDoA in AWGN Channels

In this chapter, important aspects of the TDoA system are discussed assuming Gaussian models, which represent the simplest scenario for estimation problems. The challenge lies in estimating parameters (i.e., time delay or position) of a signal embedded in Gaussian noise. The time delay estimation problem and the position estimation problem in Gaussian noise are discussed separately in the next two sections. Attainable estimation bounds are derived and theoretical results are compared to simulation results using the previously presented algorithms.

3.1 Time Delay Estimation

Time delay estimation is the first step for localization using TDoA [1]. The estimates of this step play an important role in the final position estimates of the positioning system. Therefore, analyzing different errors and their dependencies, as well as implementing algorithms to conquer them is one key point to obtaining a smart passive TDoA system.

In this section, the problem of time delay estimation in discrete time AWGN channels is addressed, focusing on interpolation techniques and their effects on the estimate.

3.1.1 System Model

The scenario described in Chapter 2 consisting of N spatially separated sensors, one being the reference sensor, is applied here. Following this scenario, the signal received by the reference sensor is used to estimate the TDoA between the reference sensor and the other $N-1$ sensors. Hence, each TDoA is estimated using two signals $r_i(t)$ and $r_j(t)$.

A very common signal model, assuming that no reflections on buildings or scattering on objects occurred, is the AWGN channel model. It takes into account the propagation delay and the thermal noise added at the receiver. Following this model, the received signals can be described as

$$\begin{aligned} r_i(t) &= s(t - t_0 - \tau_i) + w_i(t) \\ r_j(t) &= s(t - t_0 - \tau_j) + w_j(t) \end{aligned} \tag{3.1}$$

with t_0 being the transmit time, τ_i and τ_j being the distance dependent propagation delays, and w_i and w_j being the zero mean complex circular noise terms with a covariance matrix $\frac{\sigma_{w_i}^2}{2}\boldsymbol{I}_2$. In active systems, where the transmitted signal $s(t)$ and the transmit time t_0 are known, propagation delays can be measured directly. On the other hand, in passive systems, two received signals are used to estimate one TDoA. For this case, (3.1) can be rewritten as

$$\begin{aligned} r_i(t) &= s(t) + w_i(t) \\ r_j(t) &= s(t - \Delta\tau_{i,j}) + w_j(t) \end{aligned} \tag{3.2}$$

whereas $\Delta\tau_{ij}$ is the TDoA to be estimated.

3.1.2 Cross-Correlation

The most common method for time delay estimation is the matched filter followed by a peak detector. In passive systems, the matched filter is replaced by the cross-correlation given as

$$C_{i,j}(\tau) = \int_{-\infty}^{\infty} r_i(t) r_j^*(t-\tau) dt. \tag{3.3}$$

For a time limited signal, the estimated cross-correlation and subsequently the estimated TDoA is given as

$$\widehat{C}_{i,j}(\tau) = \int_0^{T_o} r_i(t) r_j^*(t-\tau) dt \tag{3.4}$$

$$\widehat{\Delta\tau}_{i,j} = \arg\max_{\tau} |\widehat{C}_{i,j}(\tau)| \tag{3.5}$$

with T_o being the limited observation window. This estimator is the maximum likelihood estimator for the time delay under the following assumptions [8]:

- The signal and noise terms are wide sense stationary (or can be assumed to be) for the observation length T_o. Only then is the correlation a function of the delay τ only and does not depend on the absolute time. In practice, observation windows are sometimes additionally limited to fulfill this requirement.
- $B \cdot T_o \gg 1$: the Bandwidth Time (BT) product should be much larger than one. The observation time needs to be much larger than the correlation time of the signal. Only then is the time-limited correlation an approximation of the true correlation.
- $T_o \gg \Delta\tau_{i,j}$: A large enough portion of the signal needs to be overlapped for the estimator to work.
- The noise spectra are assumed to be flat, a requirement that is valid for $w_i(t), w_j(t)$ being white Gaussian. Without this condition, the Maximum Likelihood (ML) estimator would require additional filtering (i.e., lower noise spectral ranges obtain higher weights).
- The signal and noise terms are independent.
- The noise terms are mutually independent.

Stochastic Analysis of the Cross-Correlator

Assuming the signal model in (3.2), the maximum likelihood estimator is found by minimizing the joint probability density functions of the signals r_i and r_j, conditioned on the values of $s(t), \tau$. For the derivation of the solution, the Fourier transform of the signals is taken in $(0, T_o)$ as

$$\begin{aligned} R_i(f) &= S(f) + W_1(f) \\ R_j(f) &= S(f)\mathrm{e}^{-\mathrm{j}2\pi\Delta\tau_{i,j}f} + W_2(f). \end{aligned} \tag{3.6}$$

Stein proved in [8] that under the previously described assumptions, the maximum likelihood estimator for the delay is

$$\widehat{\Delta\tau}_{i,j} = \arg\max_{\tau} \left| \int_0^{T_o} \mathrm{e}^{-\mathrm{j}2\pi f\tau} R_i^*(f) R_j(f) df \right|, \tag{3.7}$$

which is equivalent to the estimator given in (3.5). This result is optimal for unknown signal and noise spectra and assuming white noise. [9] and [10] presented the generalized cross-correlator estimator which shows better performances for known statistics of the signal and noise. In the Generalized Cross-Correlation (GCC) method, signals are prefiltered according to the available information. This step is important, for example, for colored noise. In that case, the maximum likelihood estimator would be required to emphasize signal energy by appropriate filtering of the received signals. If there is no information available *a priori*, a lowpass filter that passes all signal components and reduces noise power would be the right prefilter.

3.1.3 Interpolation Techniques

The previously described cross-correlation and peak detection were given in a continuous system. The received signals in the TDoA system are sampled with the sampling interval T_{samp}. Thus, the correlation estimator becomes

$$\begin{aligned} C_{i,j}^{D}[m] &= \sum_{n=0}^{K-1} r_i(nT_{\text{samp}}) r_j^*(nT_{\text{samp}} - mT_{\text{samp}}) \\ \widehat{\Delta\tau}_{i,j} &= \arg\max_{m} \left| C_{i,j}^{D}[m] \right|, \end{aligned} \tag{3.8}$$

where $n = 0, 1, ..., K-1$ and $K = \lfloor T_o / T_{\text{samp}} \rfloor$. Take for example a sampling frequency of 2 MHz, the peak detection yields estimates with a resolution of 0.5 μs, which is equivalent to a distance accuracy of 150 m assuming speed of light as the propagation speed. This quantization effect can be compensated by interpolating between the discrete time samples. Two different approaches are analyzed and discussed regarding their accuracy as well as their computational complexity.

Parabolic Interpolation

Assuming that the two received signals originated from one transmitted signal, the resulting calculated cross-correlation is nothing but the signal autocorrelation shifted from zero to τ. The symmetric nature of the autocorrelation around zero can therefore be assumed to occur at the peak around $\Delta\tau_{i,j}$. As a result, the correlation can be approximated by a parabola around the peak following

$$C_{i,j}(\tau) \approx a\tau^2 + b\tau + c, \tag{3.9}$$

where a, b and c are the unknown coefficients. The desired peak lies in the apex of the parabola at $\hat{\tau} = -\frac{b}{2a}$ [11]. To solve for the continuous maximum, three samples are needed. Since this approximation holds only for the main peak (i.e., the true correlation can have other maxima), the interpolation can be expected to work whenever the three samples used for interpolation lie in that main peak. The estimation steps are:

1. Calculating the discrete time correlation estimator given in (3.8).
2. Locating the maximum discrete lag $l = \underset{m}{\operatorname{argmax}} \left| C^D[m] \right|$.
3. Calculating the continuous estimate by using the discrete maximum and the three function values $C^D[l-1], C^D[l]$ and $C^D[l+1]$ as

$$\hat{\tau} = l \cdot T_{\text{samp}} - \frac{1}{2} \frac{C^D[l+1] - C^D[l-1]}{C^D[l+1] - 2C^D[l] + C^D[l-1]}. \tag{3.10}$$

Due to the fact that the correlation shape is not necessarily a parabola, but often for example a cos-function or a $\frac{\sin(x)}{x}$-function, the parabolic interpolation is erroneous and leads to biased estimates [12]. Calculated expressions for the mean and variance show how the quadratic interpolation leads to estimates clustered around nT_{samp}, $n \in \mathbb{Z}$ [13]. The interpolation error consists of two parts: the first part is the systematic error or the so called parabolic misfit, which is the difference between the true maximum of the correlation (without noise) and the interpolated one. The second part is the stochastic error, which results from the noise contributions in the three points used for interpolation. At a certain SNR level, the stochastic error becomes negligible and the systematic error remains as noise floor. This result will be shown in the discussion section. Still, for unknown signal and correlation shapes, the parabolic interpolation is computationally simple compared to other techniques such as upsampling and filtering which will be shown next.

Upsampling

A different interpolation approach assuming only the knowledge of the signal bandwidth is by making use of the Nyquist-Shannon-theorem for continuous-time band-limited sequences. This rule can be applied (i) either to the signals before cross-correlating them (ii) or to the discrete correlation. For any band-limited sequence $s[n]$, sampled at T_{samp} with $T_{\text{samp}} \leq \frac{1}{2B}$, the continuous sequence can be reconstructed as

$$s(t) = \sum_{n=-\infty}^{\infty} s[n] \, \mathrm{Si}\left(\frac{t - nT_{\text{samp}}}{T_{\text{samp}}}\right), \tag{3.11}$$

where $\mathrm{Si}(x) = \frac{\sin(x)}{x}$. In the software-based signal processing system, this is approximated by upsampling, i.e., by adding zeros between two samples and passing all signal components through a lowpass filter. The accuracy defined by this interpolation technique depends on the chosen upsampling factor as well as the chosen filter. After upsampling the received signals or the discrete correlation, a quadratic interpolation can be applied according to the previously described steps.

3.1.4 Estimation Bounds

Many estimation bounds for time delay have been discussed excessively in the literature [14]. The most common one is the Cramér Rao Lower Bound (CRLB), introduced as a lower bound for the variance of an unbiased estimate [15]. The CRLB is a local estimator for the time delay problem, meaning that it assumes that the right correlation peak has been found. It predicts the asymptotic behavior of the ML estimate for large data samples.

In reality, there appears a threshold phenomenon below a certain Signal-to-Noise Ratio (SNR) resulting from ambiguity in the correlation peak, making the CRLB a loose bound in these regions. Therefore, different bounds have been introduced, the most known one being the Ziv-Zakai Bound (ZZB) [16]. In [17], [18] and [19], the behavior of both bounds has been compared, focusing on the threshold phenomenon. For higher SNR regions, the ZZB converges to the CRLB. The main drawback of the ZZB is that it needs complex numerical calculations and that it usually incorporates *a priori* information about the parameter. Aiming at a less complex system with no information about the parameters, this thesis considers the CRLB as a good fit for the system.

In Appendix A, the general approach for calculating the CRLB is presented. The case of a discrete complex signal in complex white Gaussian noise is equivalent to our estimation problem. Hence, we need to rewrite our signal model to obtain the form $x[n] = s[n;\theta] + w[n]$, $w \sim \mathcal{CN}(0, \frac{\sigma_w^2}{2}\boldsymbol{I}_2)$ as in (A.5). This can be done by describing $r_j[n]$ as a function of $r_i[n]$,

$$r_j[n] = r_i[nT_{\text{samp}} - \Delta\tau_{i,j}] + \tilde{w}[n], \tag{3.12}$$

where $\tilde{w}[n] = w_i[n-\tau] + w_j[n]$, $\tilde{w} \sim \mathcal{CN}(0, \sigma_{w_i}^2 + \sigma_{w_j}^2)$. For uncorrelated white noise, shifting $w_i[n]$ will not change the probability density function of $\tilde{w}[n]$. Using the model in (3.12), the required differentiation with respect to θ, i.e., $\frac{\partial r_i(n;\Delta\tau_{i,j})}{\partial\Delta\tau_{i,j}}$, can not be computed directly. Thus, the Discrete Fourier Transform (DFT) of the sampled signals $r_i[n], r_j[n]$ is calculated and, assuming $\Delta\tau_{i,j} = lT_{\text{samp}}$, $l \in \mathbb{Z}$, the model becomes

$$R_j[k] = R_i[k]\mathrm{e}^{\frac{-\mathrm{j}2\pi\Delta\tau_{i,j}k}{N}} + \tilde{W}[k], \quad \tilde{W} \sim \mathcal{CN}\left(0, K\cdot\left(\sigma_{w_i}^2 + \sigma_{w_j}^2\right)\right), \tag{3.13}$$

where $k = 0, 1, ..., K-1$. The deviation of the simulation results from the CRLB is due to the fact that this assumption can not be held for fractional delays, reducing accuracy. Now, solving after (A.7) results in

$$\sigma_{\Delta\tau_{i,j}}^2 \geq \frac{K(\sigma_{w_i}^2 + \sigma_{w_j}^2)}{\Re\left\{\sum_{n=0}^{K-1}\left(\frac{\partial r_i(n;\Delta\tau_{i,j})}{\partial\Delta\tau_{i,j}}\right)^*\left(\frac{\partial r_i(n;\Delta\tau_{i,j})}{\partial\Delta\tau_{i,j}}\right)\right\}}, \tag{3.14}$$

resulting in

$$\sigma_{\Delta\tau_{i,j}}^2 \geq \frac{K(\sigma_{w_i}^2 + \sigma_{w_j}^2)}{\sum_{k=0}^{K-1} 4\pi^2\left(\frac{k}{K}\right)^2 |R_i[k]|^2}. \tag{3.15}$$

The bound expression shows how the expected estimation variance grows for decreasing SNR and sequence length K. Even though K in the nominator may give the impression that more samples result in lower accuracy, it must be noted that $|S[k]|^2$ also has a factor K. Therefore, K should be neglected in the nominator for the interpretation and the dependency in the denominator $\sum_{k=0}^{K-1}\left(\frac{k}{K}\right)^2$ is the one that counts. Considering the sampling theorem and hence, a sampling interval of $T_{\text{samp}} = 1/B$, the sequence length is $K = B\cdot T_o$. Assuming that the signal is white in B, i.e., $|S[k]|^2 = \sigma_S^2 = K\cdot\sigma_s^2$ and that

$\sigma_{w_i}^2 = \sigma_{w_j}^2$, the series in the denominator can be solved for and the bound can be approximated by

$$\sigma_{\Delta\tau_{i,j}}^2 \geq \frac{3}{2\pi^2 BT_o(\text{SNR}+1)}, \tag{3.16}$$

where $\text{SNR} = \frac{\sigma_s^2}{\sigma_{w_i}^2}$. Analyzing this result, the variance of the estimator decreases for higher signal bandwidths, longer observation intervals and increasing signal-to-noise ratio. This statement was observed previously using continuous models as in [20].

For low SNR values, the ML estimator can not distinguish the global maximum anymore, resulting in the previously described threshold effect, making the estimation results far worse than the attainable bounds. This is due to the fact that the CRLB assumes the right correlation maximum has been found and that the estimation inaccuracy results only from small shifts in the global maximum caused by noise [17].

Another problem is the assumption that $\Delta\tau_{i,j} = lT_{\text{samp}}$, $l \in \mathbb{Z}$, by that neglecting errors resulting from the interpolation. These effects will be shown in the next section.

3.1.5 Simulation Results and Analysis

For the simulation scenario, 1000 random Gaussian symbols were generated with a symbol duration of $T_s = 1s$, time shifted and white Gaussian noise was added. The time delays were generated randomly and contained fractional delays (i.e., not integers of the sampling rate). Using this setup, the following questions were of interest:

- What is the effect of quadratic interpolation? What is the effect of upsampling?
- How is the performance of the estimator compared to the Cramér Rao bound?
- How is the estimation error distributed?

Figure 3.1 shows the effect of quadratic interpolation in the simulated scenario. The discrete correlation contains two samples within the main lobe. The point on the left side of the maximum lies outside the main lobe and does not behave like the parabola anymore. Thus, as expected, two samples within the

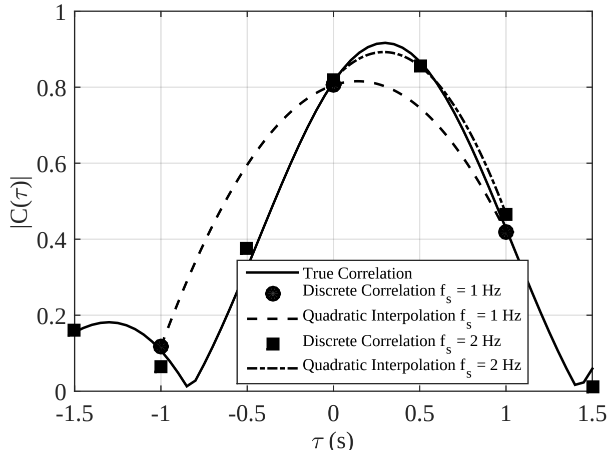


Figure 3.1: Quadratic interpolation using two sampling rates

main lobe are not sufficient to fulfill the requirement for the approximation, resulting in large errors. Using twice the sampling rate, the discrete correlation contains twice as much samples in the main lobe, resulting in a much better approximation through the parabolic interpolation. The higher the sampling rate, the closer the parabolic interpolation can get to the true curve and the higher the accuracy of the estimated delay. The quadratic interpolation needs at least three samples lying in the main lobe to attain a good approximation of the true curve.

In Figure 3.2, four different interpolations are compared to the CRLB. M presents the upsampling factor of the signals, while L presents the upsampling factor of the correlation. The same lowpass filter was used for one upsampling factor to filter the signals or the correlation. The important observations of this figure are:

- Using longer observation windows, the estimator can work even for SNR values below zero.
- Upsampling the signals results in slightly better estimates than upsampling the discrete correlation when using the same factors and lowpass filters for both methods. This can be explained as follows: In the upsampling step, the approximation error results from the implemented non-

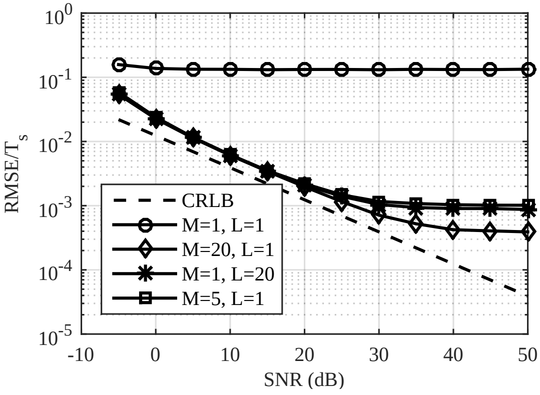


Figure 3.2: Performance of time delay estimation in AWGN

ideal filters, which are equal for both interpolations. The difference in the accuracy occurs at the correlator. The correlator described in (3.8) is an estimator of the continuous cross-correlation, where the integral is replaced by a sum and the observation length is limited. Therefore, the finer the resolution of the input signals, the higher the accuracy of the discrete correlator. Hence, upsampling the signals first results in smaller approximation errors for replacing the integral by a sum, thereby enhancing the overall accuracy.

- The noise floors caused by the parabolic misfit, starting at higher SNRs for more accurate estimators, can be seen in all curves.

- The curves for all estimators which fulfill the approximation requirements (i.e., all interpolation points are within the main lobe) show the same progression as the CRLB up to the point of the noise floor.

The next interesting question would be how the error is distributed at the output of the estimator. Depending on this distribution, appropriate position estimation methods and algorithms can be developed. Fig. 3.3 and 3.4 show the histograms using an upsampling factor of 20 and using 1000 symbols for an SNR of -5 dB and 20 dB. As a comparison, the mean and the variance of the estimation errors were determined and the probability density functions of the equivalent Gaussian distributions are shown. The figures show how, for a high upsampling factor and for a long observation window, the estimation error has

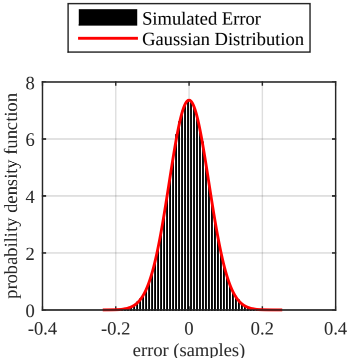


Figure 3.3: Error distribution at SNR=-5dB

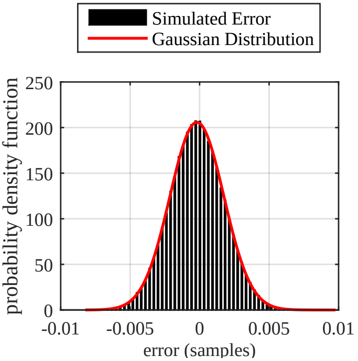


Figure 3.4: Error distribution at SNR=20dB

a zero-mean Gaussian-like distribution. In the next section, this will be the model used for position estimation algorithms.

3.2 Position Estimation

The aim of estimating the TDoAs accurately is to be able to utilize them, together with the known coordinates of the sensors, for estimating the position of the transmitter. As it has been demonstrated in the last section, the estimated TDoAs can be modeled as

$$\widehat{\Delta\tau}_{i,j} = \Delta\tau_{i,j} + \eta_{i,j} \quad \eta_{i,j} \sim \mathcal{N}(0, \sigma^2_{\Delta\tau_{i,j}}). \tag{3.17}$$

Assuming an efficient estimator, the equality in (3.15) holds and the variance of the TDoA estimate can be expressed as $\sigma^2_{\Delta\tau_{i,j}} = \sigma^2_{\tau_i} + \sigma^2_{\tau_j}$, where from (3.15),

$$\sigma^2_{\tau_i} = \frac{K\sigma^2_{w_i}}{\sum\limits_{k=0}^{K-1} 4\pi^2(\frac{k}{K})^2 \left|R_1[k]\right|^2}. \tag{3.18}$$

Hence, the single terms in the variance of the TDoA estimate in AWGN depend on the noise terms of the received signals $r_i[n]$ and $r_j[n]$. Originating from this model, the system model for the position estimation shall be discussed. Afterwards, different localization algorithms are presented and analyzed.

3.2.1 System Model

The relationship between the true source position $\boldsymbol{x} = [x_T, y_T]^T$ and the TDoA $\Delta\tau_{i,j}$, using the known sensor positions $\boldsymbol{p}_i = [x_i, y_i]^T$ and $\boldsymbol{p}_j = [x_j, y_j]^T$ is

$$\begin{aligned} c\Delta\tau_{i,j} &= c(\tau_i - \tau_j) = d_i - d_j = \Delta d_{i,j} \\ &= \sqrt{(x_i - x_T)^2 + (y_i - y_T)^2} - \sqrt{(x_j - x_T)^2 + (y_j - y_T)^2}. \end{aligned} \tag{3.19}$$

Throughout this section, the term Range Difference (RD), defined as the distance difference resulting from the TDoAs by multiplying them with the propagation speed ($\Delta d_{i,j} = c\Delta\tau_{i,j}$), will be used for the measurements, since it describes the input of the position estimation algorithms. For the estimation problem, the notation following Appendix A is used

$$\widehat{\Delta d}_{i,j} = f(\boldsymbol{p}_i, \boldsymbol{p}_j; \boldsymbol{x}) + \xi_{i,j} \tag{3.20}$$

with $\xi_{i,j} = c \cdot \eta_{i,j}$ denoting the error, which is zero-mean Gaussian with $\xi_{i,j} \sim \mathcal{N}(0, \sigma^2_{\Delta d_{i,j}})$ and $\sigma^2_{\Delta d_{i,j}} = c^2\sigma^2_{\Delta\tau_{i,j}} = \sigma^2_{d_i} + \sigma^2_{d_j}$. f is the function relating the transmitter position to the measured quantity given in (3.19). Without loss of generality, sensor 1 is defined as the reference sensor throughout this section ($j = 1$) and the estimation problem is reduced to

$$\widehat{\Delta d}_{i,1} = f(\boldsymbol{p}_i, \boldsymbol{p}_1; \boldsymbol{x}) + \xi_{i,1} \tag{3.21}$$

In vector form, the equation becomes

$$\begin{aligned} \widehat{\boldsymbol{\Delta d}}_1 &= \left[\widehat{\Delta d}_{2,1}, \widehat{\Delta d}_{3,1}, ..., \widehat{\Delta d}_{N,1}\right]^T \\ &= \boldsymbol{f}_1(\boldsymbol{p}; \boldsymbol{x}) + \boldsymbol{\xi}_1, \end{aligned} \tag{3.22}$$

where the index refers to the reference sensor. The noise vector $\boldsymbol{\xi}_1$ is now zero-mean with covariance matrix

$$\boldsymbol{C}_1 = \begin{pmatrix} \sigma^2_{d_1} + \sigma^2_{d_2} & \sigma^2_{d_1} & \cdots & \sigma^2_{d_1} \\ \sigma^2_{d_1} & \sigma^2_{d_1} + \sigma^2_{d_3} & \cdots & \sigma^2_{d_1} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma^2_{d_1} & \sigma^2_{d_1} & ... & \sigma^2_{d_1} + \sigma^2_{d_N} \end{pmatrix}. \tag{3.23}$$

Hence, the RDs have nonzero covariance outside the diagonal due to the common reference signal.

3.2.2 Least Squares Solution

For the estimation problem described above, the least squares solution is given as

$$\hat{\boldsymbol{x}}_{LS} = \arg\min_{\boldsymbol{x}} ||\widehat{\boldsymbol{\Delta d}}_1 - \boldsymbol{f}_1(\boldsymbol{p};\boldsymbol{x})||^2. \tag{3.24}$$

This describes a nonlinear equation system that needs to be solved for the unknown parameter $\boldsymbol{x} = [x_T, y_T]^T$. A direct nonlinear algorithm would be computationally costly. Therefore, many algorithms have been developed to reduce computational complexity. They vary in their estimation accuracies, their ability to utilize redundant measurements (e.g., more than two measurement in a two-dimensional system), their assumptions about sensor placement, their ability to produce exact solutions as well as their computational complexity. The approaches can be categorized as follows:

Iterative algorithms

Iterative algorithms employ a gradient search or, in some cases, techniques such as the Gauss Newton interpolation. For the TDoA estimation problem, [21] and [22] presented iterative solutions that start with an initial estimate and compute in each iteration step a new estimate with a smaller squared error. The main drawback of this approach is its dependency on the initial estimate, which needs to be close enough to the true location. As one of the most convenient iterative algorithms for TDoA estimation, the Taylor Series Estimation (TSE) was implemented and analyzed. It will be described later on.

Transformation algorithms

To simplify the equation system, an additional variable is defined in the system, which is the source-reference distance d_1, and is being treated as an independent variable. This is done by assuming that d_1 is fix and solving the resulting equation system as a function of d_1. This transformation linearizes the equation system, reducing complexity. Friedlander's method [23], the spherical interpolation method [24] as well as the spherical intersection method [25] all use this approach and develop different algorithms based on it. One drawback is the need of at least $dim + 1$ measurements instead of dim (dim being the dimension of the location to be estimated) for the original problem, but the main disadvantage is that it is suboptimal due to the fact that it ignores the dependency between d_1 and the location coordinates.

Closed-form algorithms

Compared to iterative algorithms, closed-form solutions have the obvious advantage of an exact solution, with no need for an initial guess. Fang presented an algorithm in [26] that proved to be ML-optimal for exactly determined systems (i.e., with two TDoA measurements for a two-dimensional location), making its main drawback the fact that it can not use additional measurements. Chan and Ho [27] used the spherical interpolation method, treating d_1 as an additional variable, but adding a second least squares estimation step to overcome that drawback by reintroducing the dependencies between d_1 and the parameter. Due to the second step, Chan and Ho's algorithm outperforms the transformation algorithms and additionally offers a closed-form solution. It is therefore chosen as an alternative to the iterative approach. Next, both the iterative and the closed-form algorithm are described.

Taylor Series Estimation

Presented in [21], the TSE-algorithm simplifies the problem by linearizing the least squares equation for every iteration. Using the linear terms of the Taylor series around a point $\boldsymbol{x}_0 = [x_0, y_0]^T$, f_1 can be approximated as

$$\begin{aligned}\tilde{f}_1(\boldsymbol{p}_j, \boldsymbol{x}) &\approx \sqrt{(x_j - x_0)^2 + (y_j - y_0)^2} - \sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2} + \boldsymbol{h}_j \boldsymbol{\delta} \\ &\approx f_1(\boldsymbol{p}_j, \boldsymbol{x}_0) + \boldsymbol{h}_j \boldsymbol{\delta}, \quad j = 2, 3, ..., N\end{aligned} \tag{3.25}$$

where $\boldsymbol{h}_j$ is the first derivative of $f_1(\boldsymbol{p_j}, \boldsymbol{x})$ at $\boldsymbol{x}_0$ given as

$$\boldsymbol{h}_j = \left[\frac{\partial f_1(\boldsymbol{p}_j)}{\partial x}|_{x=x_0} \quad \frac{\partial f_1(\boldsymbol{p}_j)}{\partial y}|_{y=y_0} \right] \tag{3.26}$$

and $\boldsymbol{\delta} = [\delta_x, \delta_y]^T$ is the so called shift vector. The matrix equation now becomes

$$\tilde{\boldsymbol{f}}_1(\boldsymbol{p}; \boldsymbol{x}) \approx \boldsymbol{f}_1(\boldsymbol{p}; \boldsymbol{x}_0) + \boldsymbol{H}_1 \boldsymbol{\delta}, \tag{3.27}$$

where

$$\boldsymbol{H}_1 = \begin{bmatrix} \frac{\partial f_1(\boldsymbol{p}_2;\boldsymbol{x})}{\partial x_T}|_{x_T=x_0} & \frac{\partial f_1(\boldsymbol{p}_2;\boldsymbol{x})}{\partial y_T}|_{y_T=y_0} \\ \vdots & \vdots \\ \frac{\partial f_1(\boldsymbol{p}_N;\boldsymbol{x})}{\partial x_T}|_{x_T=x_0} & \frac{\partial f_1(\boldsymbol{p}_N;\boldsymbol{x})}{\partial y_T}|_{y_T=y_0} \end{bmatrix} \tag{3.28}$$

$$= \begin{bmatrix} \frac{x_1-x_0}{d_1} - \frac{x_2-x_0}{d_2} & \frac{y_1-y_0}{d_1} - \frac{y_2-y_0}{d_2} \\ \frac{x_1-x_0}{d_1} - \frac{x_3-x_0}{d_3} & \frac{y_1-y_0}{d_1} - \frac{y_3-y_0}{d_3} \\ \vdots & \vdots \\ \frac{x_1-x_0}{d_1} - \frac{x_N-x_0}{d_N} & \frac{y_1-y_0}{d_1} - \frac{y_N-y_0}{d_N} \end{bmatrix}. \tag{3.29}$$

The TSE algorithm iteratively tries to minimize the approximated least squares equation. In each iteration step k, the estimation error is

$$\boldsymbol{e}[k] = \hat{\boldsymbol{d}}_1 - (\boldsymbol{f}_1(\boldsymbol{p}, \boldsymbol{x}_0[k]) + \boldsymbol{H}_1[k]\boldsymbol{\delta}[k]) = \boldsymbol{z}[k] - \boldsymbol{H}_1[k]\boldsymbol{\delta}[k] \tag{3.30}$$

and the least squares estimate for $\boldsymbol{\delta}$ at each step is

$$\boldsymbol{\delta}[k] = \left(\boldsymbol{H}_1[k]^T \boldsymbol{H}_1[k]\right)^{-1} \boldsymbol{H}_1[k]^T \boldsymbol{z}[k]. \tag{3.31}$$

The correction of the initial point from time step k to $k+1$ follows as:

$$\boldsymbol{x}_0[k+1] = \boldsymbol{x}_0[k] + \boldsymbol{\delta}[k]. \tag{3.32}$$

With a known or estimated noise covariance matrix $\boldsymbol{C}_1$ (3.23), the least squares estimate (which would also be the ML estimate in that case) at each iteration would be

$$\boldsymbol{\delta}[k] = \left(\boldsymbol{H}_1[k]^T \boldsymbol{C}_1^{-1} \boldsymbol{H}_1[k]\right)^{-1} \boldsymbol{H}_1[k]^T \boldsymbol{C}_1^{-1} \boldsymbol{z}[k]. \tag{3.33}$$

These steps are repeated with a new $\boldsymbol{x}_0$ in each iteration. When $||\boldsymbol{\delta}||$ is sufficiently small (e.g., smaller than one), the algorithm assumes convergence.

Parameters of this algorithm are the convergence criterion (i.e., when the algorithm should stop) and the initial position. With no *a priori* information about the position, an initial guess can be chosen for example as the center of all sensors. A bad initial estimate often results in convergence to a local, not global, minimum. Moreover, convergence of the algorithm is not guaranteed. These are the main setbacks of this algorithm.

Chan

As a comparison to the TSE estimator, the method after Chan and Ho gives a closed-form (i.e., one iteration) solution by presenting an enhanced algorithm for the spherical interpolation model. Two cases are to be distinguished:

1. Exactly dim TDoAs are available, (e.g., two for a two-dimensional localization) which is the minimum number needed. This means that the hyperbolas intersect in one point (sometimes two, which results in ambiguity). But in both cases, it is not an estimation problem anymore, rather a quadratic equation problem.

2. More than two TDoAs are available. In that case, the hyperbolas do not intersect in one point due to the estimation errors of the TDoAs. Methods like maximum likelihood or least squares estimation need to be utilized.

Minimum measurements available: This approach is based on the following two equations

$$d_i^2 = (x_i - x_T)^2 + (y_i - y_T)^2 \tag{3.34}$$

and

$$d_i^2 = (d_{i,1} + d_1)^2 \tag{3.35}$$

resulting from $d_{i,1} = d_i - d_1$. Using these equations, an explicit solution can be attained in two steps. In the first step, x and y can be solved in terms of d_1. This solution is entered in equation (3.34) at $i = 1$ and the positive square root of d_1 is used to obtain the values for x and y. Although the negative root usually yields a negative distance, sometimes, when the hyperbolas intersect in more than one point, both roots result in valid position estimates. This ambiguity can not be resolved without further information. The result is here equivalent to the spherical intersection method by [24] and yields the same results as the TSE algorithm when it converges.

Redundant measurements available: The closed-form solution is attained by applying a trick, where the three unknowns in (3.34) and (3.35) $[x_T, y_T, d_1]^T$ are treated as independent unknown variables as in all spherical interpolation methods. Under this assumption, a linear least squares solution can be obtained for these three parameters. This method reduces the complexity, but is not ML-optimal, since it does not consider all model dependencies. Chan and Ho added a step to overcome this inaccuracy, where the dependency between $\boldsymbol{x}$ and d_1 is reintroduced in a second step and another least-squares estimate is calculated to correct the estimates of the first step. It was shown that this algorithm presents a good approximation of the ML solution for low noise levels. High errors lead to biased solutions in the first step, therefore inaccurate estimates.

3.2.3 Kalman Filters

A signal of interest that is being located would be observed as long as needed. For example, if it is an unlicensed transmitter, its position needs to be tracked until action can be taken. Since the described scenario focuses mainly on static and slowly moving transmitters, it would be interesting to know how to ideally combine consecutive TDoA measurements to enhance the position accuracy in case that the transmitter stays in the same position, and in case that it moves.

On the other hand, it would be interesting to know how different kinds of measurements can be combined in an effective way. In Chapter 2, different localization methods based on various measurements were described. Even though TDoA fits best in the described scenario, sometimes, additional measurements are available that can help to enhance the estimation accuracy.

How to efficiently use sequential measurements as well as fuse different measurement systems is solved by the Kalman filter, which is described next. First, the Kalman filter equations are explained, then two modified Kalman filters for nonlinear systems are presented for the TDoA problem. Last but not least, a hybrid Kalman filter is presented for combining TDoA and RSSD measurements.

Kalman Filter Equations

The Kalman filter assumes the following Gauss-Markov model for the underlying state (i.e., the parameter or the signal that needs to be estimated) [28], [29]

$$\boldsymbol{x}[n] = \boldsymbol{A}\boldsymbol{x}[n-1] + \boldsymbol{w}[n], \tag{3.36}$$

where $\boldsymbol{x}[n]$ is the state vector at step n, $\boldsymbol{A}$ is the transition matrix, $\boldsymbol{w}[n]$ is the random process noise with $\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q})$. On the other hand, the observation equation is

$$\boldsymbol{z}[n] = \boldsymbol{H}\boldsymbol{x}[n] + \boldsymbol{v}[n], \tag{3.37}$$

where $\boldsymbol{H}$ is the observation matrix and $\boldsymbol{v}[n]$ is a noise vector with $\boldsymbol{v} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{C})$. Generalizing the model through time-variant $\boldsymbol{A}$ and $\boldsymbol{H}$ is possible but is not needed here.

For describing the Kalman filter steps, the predicted state at step n using the last step $n-1$ is denoted as $\hat{\boldsymbol{x}}[n|n-1]$ and the estimated state at step n is denoted as $\hat{\boldsymbol{x}}[n|n]$.

In each step, the Kalman filter aims at finding the Minimum Mean Square Error (MMSE) estimate using the previous state and the new observation following the next two steps:

1. **Prediction Step:** The predicted state $\hat{\boldsymbol{x}}[n|n-1]$ and the prediction covariance matrix $\hat{\boldsymbol{P}}[n|n-1]$ are calculated as

$$\hat{\boldsymbol{x}}[n|n-1] = \boldsymbol{A}\hat{\boldsymbol{x}}[n-1|n-1] \tag{3.38}$$

$$\hat{\boldsymbol{P}}[n|n-1] = \boldsymbol{A}\hat{\boldsymbol{P}}[n-1|n-1]\boldsymbol{A}^T + \boldsymbol{Q}, \tag{3.39}$$

   where $\hat{\boldsymbol{x}}[n-1|n-1]$ is the estimated state and $\hat{\boldsymbol{P}}[n-1|n-1]$ is the estimated covariance matrix at step $n-1$.

2. **Correction or Filter Step:** The predicted step is corrected using the Kalman gain $\boldsymbol{K}$ and the estimated state and covariance matrix are calculated as

$$\hat{\boldsymbol{x}}[n|n] = \hat{\boldsymbol{x}}[n|n-1] + \boldsymbol{K}[n](\boldsymbol{z}[n] - \boldsymbol{H}\hat{\boldsymbol{x}}[n|n-1]) \tag{3.40}$$

$$\boldsymbol{K}[n] = \hat{\boldsymbol{P}}[n|n-1]\boldsymbol{H}^T\left(\boldsymbol{H}\hat{\boldsymbol{P}}[n|n-1]\boldsymbol{H}^T + \boldsymbol{C}\right)^{-1} \tag{3.41}$$

$$\hat{\boldsymbol{P}}[n|n] = (\boldsymbol{I} - \boldsymbol{K}[n]\boldsymbol{H})\,\hat{\boldsymbol{P}}[n|n-1]. \tag{3.42}$$

The Kalman filter summarizes the effects of all past inputs to the system, allowing to estimate $\boldsymbol{x}[n]$ based on $\boldsymbol{z}[0], \boldsymbol{z}[1], ..., \boldsymbol{z}[n]$ as n increases. Such an operation is referred to as filtering whereas the Kalman gain is designed to optimally weight the innovation $\boldsymbol{z}(n) - \boldsymbol{H}\hat{\boldsymbol{x}}[n|n-1]$. It depends on the confidence in the new data sample compared to the confidence in the system. A noise free measurement has a large weight in the new estimate and vice versa. For a time-invariant parameter vector, where $\hat{\boldsymbol{x}}[n|n-1] = \hat{\boldsymbol{x}}[n-1|n-1]$, the Kalman filter is the sequential linear MMSE estimator [28]. For a static or a moving target, the Kalman filter integrates every new measurement $z[n]$ in an optimal way, assuming a known system model.

For the TDoA system, the state $\boldsymbol{x}$ would be the position of the transmitter with $\boldsymbol{x} = [x_T, y_T]^T$ and the transition matrix $\boldsymbol{A}$ would be the movement matrix. In the described scenario, it is assumed that the transmitter is either stationary or is slowly moving. For a stationary transmitter $\boldsymbol{A}$ is a two-dimensional identity matrix $\boldsymbol{I}_2$. For a moving transmitter with a known constant velocity vector, the relative shift per filter step can be defined as $\boldsymbol{\Delta} = [\Delta_x, \Delta_y]^T$ and the transition matrix becomes

$$\boldsymbol{A} = \begin{bmatrix} 1+\Delta_x & 0 \\ 0 & 1+\Delta_y \end{bmatrix}. \tag{3.43}$$

Unknown velocities can be estimated by expanding the state vector to include the velocity values as $\boldsymbol{x} = [x_T, y_T, \Delta_x, \Delta_y]$. The transition matrix becomes

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3.44}$$

The term $\boldsymbol{w}[n]$ includes all system uncertainties due to unknown velocity or various inaccuracies in the system model. For a perfectly known system evolution, $\boldsymbol{w}[n] = 0$. Unknown or erroneous system models include the model uncertainty through the process noise term $\boldsymbol{w}[n]$ and its covariance matrix $\boldsymbol{Q}$. $\boldsymbol{z}[n]$ is the observed value, here the TDoA estimate and $\boldsymbol{C}$ its covariance matrix, assumed to be known. The observation matrix $\boldsymbol{H}$ links the position with the TDoAs and requires a linear relation between the measurement and the state vector.

Hence, since the Kalman filter was designed for linear systems, modified Kalman filters are needed for the TDoA estimation problem, having a nonlinear relation between the measurement and the state.

Extended Kalman Filter

The Extended Kalman Filter (EKF) operates very similarly to the original filter. The main difference is that it linearly approximates the model in each step by replacing the nonlinear functions (here: the observation function) with their Jacobian matrix around the currently predicted state [30]. For the TDoAsystem, since $\boldsymbol{f}_1(\boldsymbol{p};\boldsymbol{x})$ is nonlinear, the observation matrix is approximated at step n by its Jacobian as

$$\boldsymbol{H}_1 = \begin{bmatrix} \frac{\partial f_1(\boldsymbol{p}_2;\boldsymbol{x})}{\partial x_T}\big|_{x_T=\hat{x}[n|n-1]} & \frac{\partial f_1(\boldsymbol{p}_2;\boldsymbol{x})}{\partial y_T}\big|_{y_T=\hat{y}[n|n-1]} \\ \cdots & \cdots \\ \frac{\partial f_1(\boldsymbol{p}_N;\boldsymbol{x})}{\partial x_T}\big|_{x_T=\hat{x}[n|n-1]} & \frac{\partial f_1(\boldsymbol{p}_N;\boldsymbol{x})}{\partial y_T}\big|_{y_T=\hat{y}[n|n-1]} \end{bmatrix}. \tag{3.45}$$

This matrix was already used in (3.28). The difference is that the expansion point here is the predicted state. Hence, the EKF approximates the nonlinear observation function with a first order Taylor series expansion about the predicted position estimate. The remaining operations are equal to (3.38)-(3.42).

Unscented Kalman Filter

In case of a highly nonlinear observation function, a linear approximation would lead to large estimation errors. The Unscented Kalman Filter (UKF) presented in [31] gives an alternative to the EKF with a more promising performance for highly nonlinear systems by replacing the linear measurement matrix $\boldsymbol{H}$ with the true measurement equation $\boldsymbol{f}_1(\boldsymbol{p};\boldsymbol{x})$ from (3.19).

The main idea of theUKF lies in generating the so called Sigma-points that have the same mean and covariance matrix as the current system state and propagating these points through the system dynamics. Since the state transition is assumed to be linear (i.e., either stationary or moving with constant velocity), the prediction step follows (3.38) and (3.39). For the filter step, the Sigma-points need to be generated.

Different methods for generating the Sigma-points are found in the literature. Here, the method of Julier and Uhlmann [32] is used. For a state dimension N_{dim}, Sigma-points $\boldsymbol{m}_i$ with $i = 1, 2, ..., 2N_{dim} + 1$ are generated using the predicted state $\hat{\boldsymbol{x}}[n|n-1]$ and the predicted covariance matrix $\hat{\boldsymbol{P}}[n|n-1]$ following

$$\begin{aligned}
\boldsymbol{m}_0 &= \hat{\boldsymbol{x}}[n|n-1] \\
\boldsymbol{m}_i &= \boldsymbol{m}_0 + \left[\sqrt{\frac{N_{dim}}{1-W_0}\hat{\boldsymbol{P}}[n|n-1]}\right]_i \\
\boldsymbol{m}_{i+N_{dim}} &= \boldsymbol{m}_0 - \left[\sqrt{\frac{N_{dim}}{1-W_0}\hat{\boldsymbol{P}}[n|n-1]}\right]_i \\
g_i &= \frac{1-g_0}{2N_{dim}}, \quad i = 1, 2, ..., N_{dim}.
\end{aligned} \tag{3.46}$$

$\left[\sqrt{\cdot}\right]_i$ denotes the ith row of the matrix square root of a positive semidefinite matrix. g_0 and g_i are the according weights of the generated points and g_0 is a tuning parameter that can take any value except for one.

Predicting Measurements: The Sigma-points are projected through the measurement function $\boldsymbol{f}_1(\boldsymbol{p};\boldsymbol{x})$, meaning that each Sigma-point is regarded as a transmitter position $\boldsymbol{m}_i = [x_{m_i}, y_{m_i}]^T$ and the corresponding RDs are calculated. The resulting measurement vector for Sigma-point i is

$$\boldsymbol{z}_i = \boldsymbol{f}_1(\boldsymbol{p};\boldsymbol{m}_i) = \frac{1}{c}\left[\widehat{\Delta d}_{2,1}, \widehat{\Delta d}_{3,1}, ..., \widehat{\Delta d}_{N-1,1}\right]^T$$
$$\widehat{\Delta d}_{k,1}(\boldsymbol{m}_i) = \sqrt{(x_k - x_{m_i})^2 + (y_k - y_{m_i})^2} - \sqrt{(x_1 - x_{m_i})^2 + (y_1 - y_{m_i})^2}. \tag{3.47}$$

The predicted measurement results from the weighted sum of the projected Sigma-points given as:

$$\boldsymbol{z}[n|n-1] = \sum_{i=0}^{2N_{dim}} g_i \boldsymbol{z}_i. \tag{3.48}$$

Calculating Innovation and Cross-Covariance Matrices: The predicted measurement covariance and cross-covariance matrix can now be calculated as

$$\begin{aligned} \boldsymbol{P}_{zz}[n] &= \sum_{i=0}^{2N_{dim}} g_i \left(\boldsymbol{z}_i - \boldsymbol{z}[n|n-1]\right)\left(\boldsymbol{z}_i - \boldsymbol{z}[n|n-1]\right)^T + \boldsymbol{C}_1 \\ \boldsymbol{P}_{xz}[n] &= \sum_{i=0}^{2N_{dim}} g_i \left(\boldsymbol{x}_i - \hat{\boldsymbol{x}}[n|n-1]\right)\left(\boldsymbol{z}_i - \boldsymbol{z}[n|n-1]\right)^T, \end{aligned} \tag{3.49}$$

with $\boldsymbol{C}_1$ being the covariance matrix of the RD estimates which is assumed to be known.

Filter Step: To update the state vector and covariance matrix, the following steps are applied

$$\begin{aligned} \boldsymbol{K}[n] &= \boldsymbol{P}_{xz}[n]\boldsymbol{P}_{zz}^{-1}[n] \\ \hat{\boldsymbol{x}}[n|n] &= \hat{\boldsymbol{x}}[n|n-1] + \boldsymbol{K}[n]\left(\boldsymbol{z}[n|n-1] - \boldsymbol{z}[n]\right) \\ \hat{\boldsymbol{P}}[n|n] &= \hat{\boldsymbol{P}}[n|n-1] + \boldsymbol{K}[n]\boldsymbol{P}_{zz}[n]\boldsymbol{K}^T[n]. \end{aligned} \tag{3.50}$$

Hybrid Unscented Kalman Filter

In addition to the ability of combining sequential TDoA measurements in an optimal way, the Kalman filter can also be used to fuse different kinds of measurements. Here, a hybrid system, presented in [33], that uses the UKF to combine TDoA and RSSD measurements is described and implemented. RSSD measurements can be obtained passively with the same system setup that has been used so far. The estimated received signal power at sensor i can be modeled as

$$\hat{S}_i = S_{\text{TX}} d_i^{-\gamma} l_i, \tag{3.51}$$

where S_{TX} is the unknown transmit power of the signal, γ is a path loss exponent and is assumed to be known, and d_i is the distance between the transmitter and sensor i. l_i denotes log-normal fading, i.e.,

$$l_i = 10^{\frac{n_i}{10}}, \quad n_i \sim \mathcal{N}(0, \sigma_{S_i}^2). \tag{3.52}$$

Expressing equation (3.51) in dB and subtracting the power measurements of two sensors, the relation between the RSSDs and the transmitter position can be established as

$$\begin{aligned} \hat{\Omega}_{i,k} &= \hat{\Omega}_i - \hat{\Omega}_k \\ &= 10\gamma \log_{10}(d_k) - 10\gamma \log_{10}(d_i) + n_i - n_k, \end{aligned} \tag{3.53}$$

where

$$\hat{\Omega}_i = 10 \log_{10}(\hat{S}_i) = M - 10\gamma \log_{10}(d_i) + n_i. \tag{3.54}$$

The equation system of the RSSDs describes circles that are defined by the ratio of the distance of two sensors to the transmitter. The equation system is highly nonlinear. Therefore, the UKF, being suitable for nonlinear estimation problems, can be utilized.

To include these measurements in the UKF equations, the predicted measurement vector is now expanded to

$$\boldsymbol{z}_i = [\hat{d}_{2,1}, \hat{d}_{3,1}, ..., \hat{d}_{N,1}, \hat{\Omega}_{2,1}, \hat{\Omega}_{3,1}, ..., \hat{\Omega}_{N,1}], \tag{3.55}$$

where

$$\begin{aligned} \hat{\Omega}_{k,1}(\boldsymbol{x}_i) = {} & 10\gamma \log_{10}(\sqrt{(x_1 - x_i)^2 + (y_1 - y_i)^2}) \\ & - 10\gamma \log_{10}(\sqrt{(x_k - x_i)^2 + (y_k - y_i)^2}). \end{aligned} \tag{3.56}$$

Assuming that the RSSDs are independent from the TDoA measurements, the measurement covariance matrix becomes

$$\boldsymbol{C} = \begin{pmatrix} \boldsymbol{C}_T & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{C}_P \end{pmatrix}, \tag{3.57}$$

where $\boldsymbol{C}_T$ is the covariance matrix of the RD measurements and $\boldsymbol{C}_P$ is the covariance matrix of the RSSD measurements. Assuming that the noise terms enter the measurements independently can be justified. Whereas for TDoA the main error source is the measurement noise, the RSSD error originates mainly from shadowing [34]. The remaining filter steps are the same as described in (3.49) and (3.50).

3.2.4 Geometrical Aspects and Estimation Bounds

For designing the position estimation system, it is important to understand what defines the attainable accuracy of the system. The estimation accuracy depends on three elements:

- The accuracy of the estimated TDoAs
- The implemented estimation method
- The geometrical setup of the sensors and the emitter

Based on this information, the following points need to be analyzed before designing the TDoA positioning system.

1. What are the attainable estimation bounds?
2. Which geometrical constellations are preferable?
3. What is the minimum number of sensors needed to achieve a certain accuracy?
4. How should the system choose the reference sensor?
5. What is the attainable bound for multiple sequential measurements?
6. What is the attainable bound when combining different measurement types?

Estimation Bounds

The first question plays a role for the system evaluation. After conducting a measurement and estimating a position, it is important to be able to state the expected accuracy of the estimated position. For that, the CRLB can be utilized again. It is well known that the CRLB for the TDoA position estimation problem is [27]

$$\sigma_{\boldsymbol{x}}^2 \geq \mathrm{Tr}\left(\boldsymbol{H}_1^T \boldsymbol{C}_1^{-1} \boldsymbol{H}_1\right)^{-1}, \tag{3.58}$$

where $\boldsymbol{H}_1$ is the Jacobian matrix given in (3.28) here at the true transmitter position ($\boldsymbol{x}_0 = \boldsymbol{x}$) and $\boldsymbol{C}_1$ is the covariance matrix given in (3.23). Another way to specify the Jacobian matrix is

$$\boldsymbol{H}_1 = \begin{bmatrix} \cos(\Phi_1) - \cos(\Phi_2) & \sin(\Phi_1) - \sin(\Phi_2) \\ \vdots & \vdots \\ \cos(\Phi_1) - \cos(\Phi_N) & \sin(\Phi_1) - \sin(\Phi_N) \end{bmatrix} \tag{3.59}$$

where $\Phi_1, ..., \Phi_N$ are the bearing angles from the sensors to the signal source. Observing (3.59), the absolute distances between the sensors and the transmitters do not matter to the system accuracy (apart from the fact that a further sensor usually results in lower SNR and thus, less accurate TDoAs). The accuracy depends on the bearing angles, specifically the difference between the angles. If two or more sensors are aligned with the transmitter, the variance would be ∞. The closer the angles, the larger the estimation variance.

Whereas the CRLB describes the attainable bound with a given measurement accuracy and a given setup, an alternative measure for the accuracy of a TDoA system is the Geometric Dilution of Precision (GDoP). It describes the effect of the TDoA error amplification due to the geometrical constellation. The GDoP is given as [22]

$$\mathrm{GDoP} = \frac{\sqrt{\mathrm{Tr}(H^T H)^{-1}}}{\sqrt{\bar{\sigma}_d^2}}, \tag{3.60}$$

where $\bar{\sigma}_d^2$ is the average variance of the RD estimates. For uncorrelated RDs with equal variances σ_d^2, $\bar{\sigma}_d^2 = \sigma_d^2$. The GDoP can be interpreted as the factor that scales the TDoA estimation error to the designated position estimation error. Larger GDoP values mean the localization accuracy decreases fast for growing TDoA error.

Preferred Setups

To answer the second question, Figure 3.5 shows an example of two different geometric constellations and their estimation performances. The figures show two constellations consisting of three sensors, one of which being chosen as reference sensor, and a transmitter. The two setups differ only in one sensor position. The figures show the hyperbolas using the true RDs, and, with the dashed lines, the hyperbolas shifted by an RD error of $\pm 20m$. The figures below zoom in on the same area size and show the uncertainty regions. In the constellation on the left, the hyperbolas intersect at sharp angles, since the sensors are well separated relative to the transmitter. On the right side, the sensors are not so well separated, specifically in the y-axis, causing the hyperbolas to intersect at flat angles. This spreads out the uncertainty region accordingly and reduces the estimation accuracy at equal TDoA error. Therefore, preferred constellations include angles that are as sharp as possible for all sensors. One such optimal constellation is to place the sensors equally spaced on a circle with the transmitter in the middle [35]. A general rule is to spread out the sensors as far as possible in both x- and y-axis (while considering coverage and SNR issues).

Minimal Number of Sensors

Considering the third question, assume an optimal geometrical setup and, to simplify the model, uncorrelated TDoA estimates with equal variances σ_d^2. The minimum value for the CRLB of an optimal setup is [36]

$$\sigma_{\text{min}}^2 = \sigma_d^2 \frac{27}{16(N-1)}, \tag{3.61}$$

yielding the minimum number of sensors needed. Hence, for a desired minimum accuracy σ_{min} and for an expected average RD accuracy σ_d in the best case scenario, $N \geq \frac{27}{16}\frac{\sigma_d^2}{\sigma_{\text{min}}^2} + 1$, $N \geq dim + 1$ where dim is the system dimension.

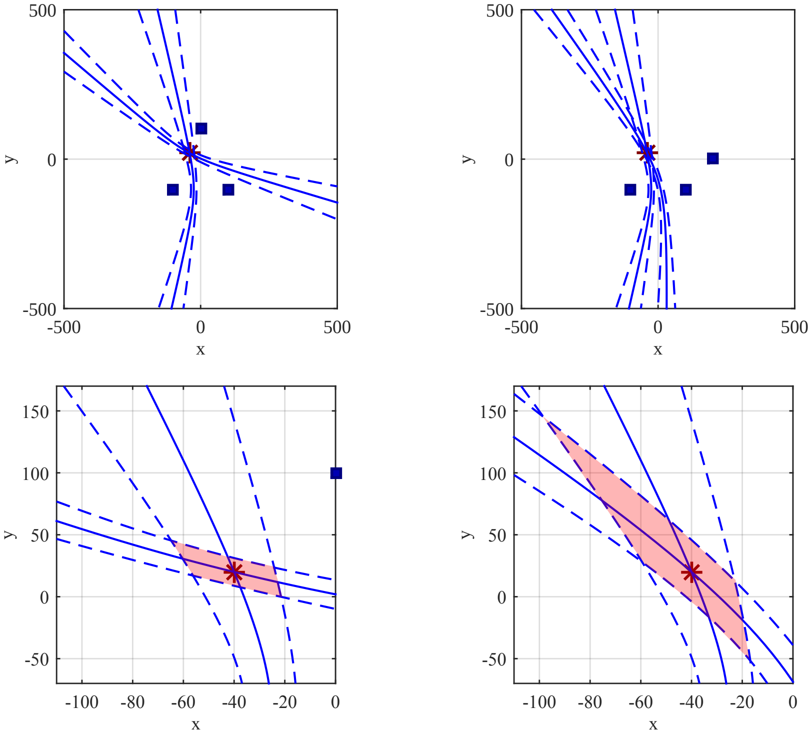


Figure 3.5: Uncertainty regions resulting from different TDoA geometrical setups

Choosing the Reference Sensor

Given a TDoA system with an initial guess about the sensor position, how should the reference sensor be chosen? Figure 3.6 shows a contour graph on how to chose the reference sensor for a given setup with a diagonal covariance matrix with equal variances. The regions mark the transmitter positions that are best to be estimated when choosing the according sensor. For example, if the transmitter is somewhere in the upper right corner of the plotted area, then the red sensor should be chosen as the reference sensor. Therefore, given the sensor positions, an initial emitter position estimate and, if possible, an estimate of the RD covariance matrix, the method for choosing the reference

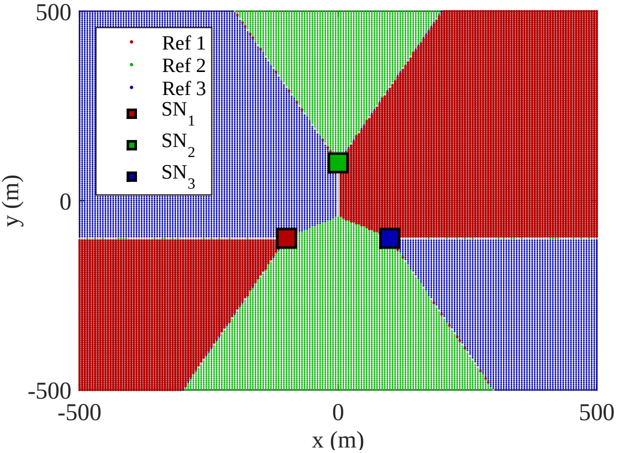


Figure 3.6: Best reference regions depending on the transmitter position

sensor is

$$Ref = \arg\min_i \left[\mathrm{Tr} \left[\left(\boldsymbol{H}_i^T \boldsymbol{C}_i^{-1} \boldsymbol{H}_i \right)^{-1} \right] \right], \tag{3.62}$$

where the index of the matrix indicates the reference sensor number.

Sequential Measurements

Due to the whiteness of the delay estimation error, two consecutive measurements are independent, meaning that the Fisher information of the two measurements is the sum of the Fisher information of the single measurements. Assuming a minimum variance of $\sigma_{\boldsymbol{x}}^2$ for the accuracy of the position estimate using one measurement, the Cramér Rao bound for M consecutive TDoA measurements becomes

$$\sigma_{\boldsymbol{x},M}^2 \geq \frac{1}{M} \sigma_{\boldsymbol{x}}^2. \tag{3.63}$$

Hybrid System Bound

The same principle applies to different kinds of measurements, assuming they are independent. For the described TDoA/RSSD measurement system, the overall Fisher information matrix can be described as [33]

$$\boldsymbol{F_{Hybrid}} = \boldsymbol{F}_T + \boldsymbol{F}_P, \tag{3.64}$$

where $\boldsymbol{F}_T = \boldsymbol{H}_1^T \boldsymbol{C}_1^{-1} \boldsymbol{H}_1$ is the Fisher information from the TDoA estimates (see eq. (3.58)) and $\boldsymbol{F}_P$ is the Fisher information available through the RSSD measurements given by [6]:

$$\boldsymbol{F}_P = \left(\sum_{k=1}^{N} \sigma_{S_k}^2\right) \cdot \left(\frac{10\gamma}{\ln 10}\right)^2 \cdot \begin{pmatrix} \xi - \chi^2 & \rho - \chi \cdot \nu \\ \rho - \chi \cdot \nu & \psi - \nu^2 \end{pmatrix}, \tag{3.65}$$

where

$$w_k = \left[\sum_{i=1}^{N} \sigma_{S_i}^{-2}\right]^{-1} \cdot \sigma_{S_k}^{-2} \tag{3.66}$$

$$a_k = \frac{x - x_k}{d_k^2}, \quad b_k = \frac{y - y_k}{d_k^2} \tag{3.67}$$

$$\chi = \sum_{k=1}^{N} w_k a_k, \quad \nu = \sum_{k=1}^{N} w_k b_k \tag{3.68}$$

$$\xi = \sum_{k=1}^{N} w_k a_k^2, \quad \psi = \sum_{k=1}^{N} w_k b_k^2 \tag{3.69}$$

$$\rho = \sum_{k=1}^{N} w_k a_k b_k. \tag{3.70}$$

The terms result from applying (A.10) to the measurement equation (3.53), i.e., by calculating the derivatives of the measurement equation over the parameters x_T, y_T. As was shown in [6], the expression does not depend on the choice of the reference sensor, as opposed to the bound based on TDoA measurements.

3.2.5 Simulation Results and Analysis

After presenting the positioning algorithms and the attainable estimation bounds, simulation results can give more insight into how to design the TDoA system. In this section, different scenarios are analyzed to compare the aforementioned algorithms regarding accuracy, robustness and efficiency. The presented results, together with the results of the first section, lead to a first conceptual design of the designated TDoA system.

The following points need to be examined regarding the position estimation algorithms in AWGN:

- How do the two chosen Least Squares (LS)-algorithms perform compared to each other and compared to the CRLB? By answering this question, the choice of the algorithms is justified.
- When a transmitter can be observed over a long period, sequential measurements are available. How can we make use of the newly gained information? When is it reasonable to apply a Kalman filter instead of a simple averaging over available estimates?
- How does the proposed hybrid UKF perform compared to the CRLB? When does the system benefit from the additional RSSD measurements?
- If the transmitter is moving with an unknown velocity, how does the Kalman filter perform?

One Shot Least Squares Algorithm Performance

At first we evaluate the two least squares algorithms presented in Section 3.2.2, depending on the measurement error as well as the geometry for the simple case of one-shot estimation, i.e., only one measurement is used for one position estimation. To compare the algorithms, a scenario consisting of five sensors which are distributed in a plane is chosen. A transmitter position is generated (randomly or deterministically) and noise-free TDoAs are calculated using one of the sensors as reference sensor. An error vector is generated as a zero-mean Gaussian process with covariance matrix

$$\boldsymbol{C}_1 = \begin{pmatrix} 2\sigma_n^2 & \sigma_n^2 & \sigma_n^2 & \sigma_n^2 \\ \sigma_n^2 & 2\sigma_n^2 & \sigma_n^2 & \sigma_n^2 \\ \sigma_n^2 & \sigma_n^2 & 2\sigma_n^2 & \sigma_n^2 \\ \sigma_n^2 & \sigma_n^2 & \sigma_n^2 & 2\sigma_n^2 \end{pmatrix}, \tag{3.71}$$

i.e., by applying (3.23) and assuming equal noise variances for all sensors. Using this setup and varying the geometrical setup as well as σ_n^2, the two least squares algorithms (TSE and Chan) can be compared to each other as well as to the CRLB.

Figure 3.7 shows the performance of the algorithms with a known covariance matrix for a growing noise standard deviation σ_n from (3.71). Different setups were chosen on a 2000m $\times$ 2000m plane. On the left side, the Root Mean Square Error (RMSE), which can be compared to the square root of the CRLB is plotted. On the right side, two performance criteria are shown: the failure rates due to numerical instabilities in the algorithms and the direct comparison between both algorithms. This direct comparison is for the cases where both algorithms do not fail. $\epsilon_{TSE} < \epsilon_{Chan}$ means that the estimation error using the TSE is smaller than that using the Chan algorithm. Both comparisons are shown, since they do not add up to 100 $\%$ due to the failures.

Three different geometrical setups are chosen. The first one is the optimal setup with the sensors placed on a circle around the transmitter. The TSE and the Chan algorithm have a failure rate of almost zero for all error values. The TSE curve is aligned with that of the CRLB. Chan, on the other hand, shows a better performance than the CRLB. The reason behind this behavior is that Chan's algorithm is not an unbiased estimator for the position as was shown in [37]. In fact, if the transmitter position is close to the center, Chan's algorithm estimates the center, on average. Therefore, Chan's algorithm can not be directly compared to the CRLB.

The second geometrical setup consists of the five sensors placed on a circle with a radius of 1000 meters and a randomly placed transmitter for each realization. For every noise value, 10^5 realizations were generated. This time, the TSE shows better results for a growing noise variance and even outperforms the CRLB. Chan is almost aligned with the CRLB. On the right side, it can be seen that the failure rate of the TSE increases and goes up to 18 % and that of Chan's algorithm goes up to 12 % which explains why the curves are better than the CRLB curve. Bad geometrical constellations, where two or more sensors are aligned with the transmitter, lead to extremely large GDoPs and CRLB, respectively. In these cases, the algorithms simply fail to obtain an estimate. On the other hand, when both algorithms do not fail, the TSE leads to more accurate results in $50\% - 60\%$ of the time.

In the third setup, random positions are chosen for the transmitter as well as for the sensors in each realization. Not all random geometries were allowed in the simulation. Every constellation where one or more sensors were aligned with the line connecting the reference sensor to the transmitter, as well as within ± 5 degrees of that line was eliminated. These setups have very high GDoPs that, on the one hand, yield extremely high CRLBs (i.e., having one realization in the magnitude of 10^9 of the others) while, on the other hand, these scenarios do not resemble real scenarios. In a realistic setup, the sensors would be well separated in the area of interest. Some real scenarios can include a case where two sensors and the transmitter are (approximately) aligned, nevertheless, the actual performance depends on the fact whether the reference sensor is one of these two sensors. Since the reference sensor can be changed any time, this would be a perfect example of when to change it. The direct comparison between both algorithms shows, again, how the TSE outperforms Chan's algorithm.

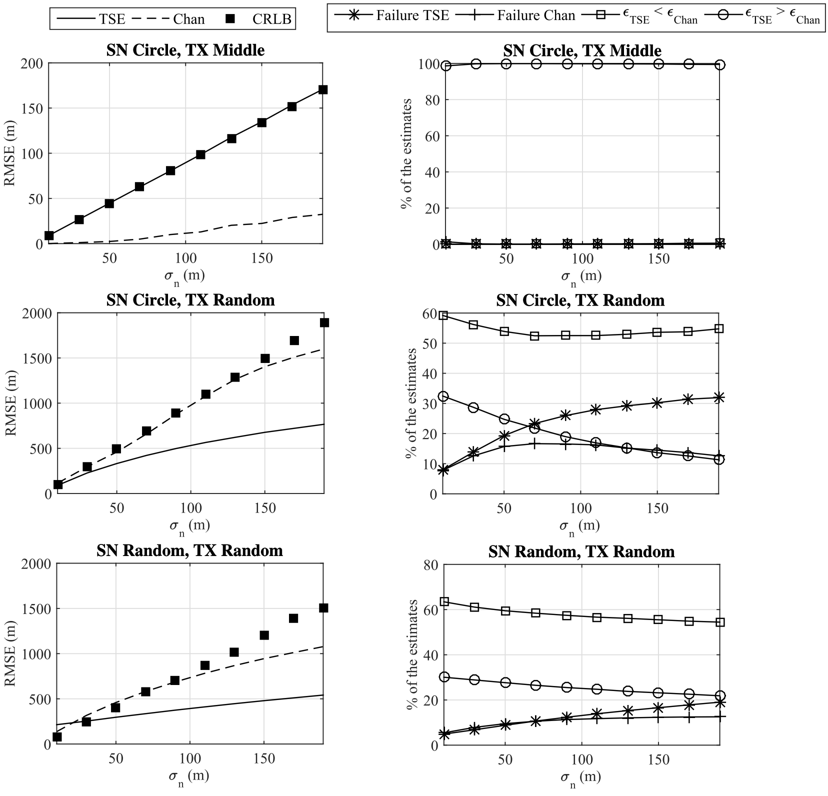


Figure 3.7: Performance of the LS algorithms compared to the CRLB. Left: RMSE, right: failure rates and direct comparison between algorithms

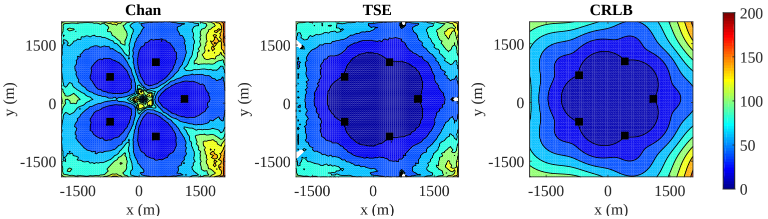


Figure 3.8: Contour graph of RMSE for different transmitter positions and five sensors on a circle. Left to right: Chan, TSE, CRLB

Figure 3.8 shows a contour graph for the RMSE at different transmitter positions for the circle setup described before. The noise standard deviation was kept fixed at 50 meters. The figure compares the TSE and the Chan algorithm with known covariance matrix. The white areas in the TSE plot belong to canceled values due to very high estimation variance. This plot shows how the TSE has a better performance for most of the transmitter positions except for the small white areas.

To conclude, the TSE has shown better estimation results in general, i.e., for most geometrical setups, but tends to fail more often (except for the optimal constellation). Therefore, aiming at higher accuracy, the TDoA system should be designed so that it first tries estimating the position using the TSE, then if the algorithm fails, to apply Chan's algorithm.

Combining Sequential Measurements

In most localization applications, the goal is to track the device or the signal source for a longer time period (e.g., until action can be taken against unlicensed users), even in the static transmitter case. The simulated scenario consists of sequentially estimated TDoAs originating from a static transmitter. To compare the algorithms, an initial estimate is used for the EKF and the UKF with known measurement covariance matrix and an estimated initial state covariance matrix. For a fair comparison with the TSE, estimated RDs are averaged in each step and used as an input to the TSE, together with the true measurement covariance matrix.

Figure 3.9 shows the performance of the algorithms over increasing step number, compared to the CRLB. In the left picture, normally distributed measure-

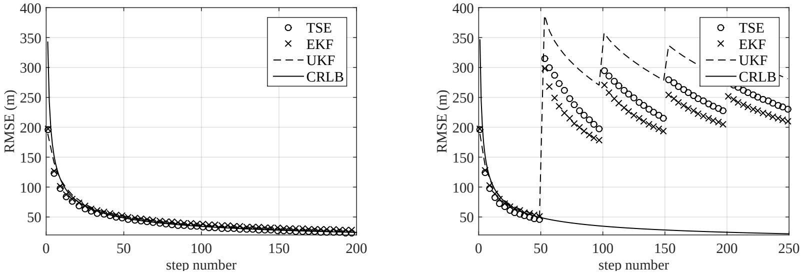


Figure 3.9: Performance of different sequential measurement combining methods left: without outliers; right: with outliers

ments with covariance matrices like in (3.71) are used. On the right side, outliers are generated, independent from the true measurements, at step number 50, 100, 150 and 200. In the left figure, all algorithms perform equally well and nearly equal to the CRLB. On the right side, the EKF shows a smoother and more accurate result after the outliers, whereas the UKF shows the worst performance.

Not all results were considered in the picture, only results where the initial estimate lies within a range of 500 meters. The Kalman filters need a good initial estimate since they do not estimate a new position every time, they rather track and correct an already estimated position. A robust localization system can therefore be designed as follows:

1. For an amount of k (e.g., $k = 20$) sequential TDoA measurements, use TSE or Chan to estimate the position and save the TDoAs.

2. After k TDoAs, calculate the average measurement vector and estimate an initial position for the EKF using TSE or Chan.

3. Use the EKF with the known or estimated covariance matrices to keep track of the estimated position.

This approach is especially useful, since it is not known whether a transmitter is static or moving *a priori*. The EKF can therefore be extended directly to include an estimate for the velocity and track a moving transmitter, if needed.

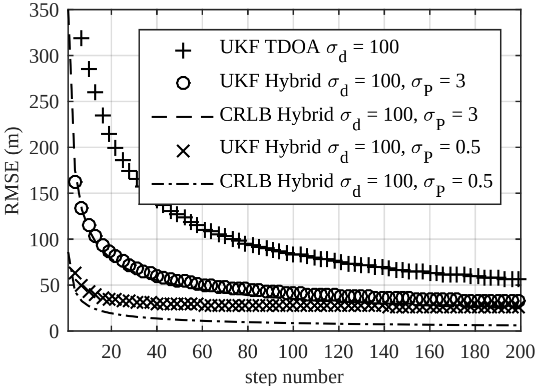


Figure 3.10: Comparison of the TDoA UKF and the hybrid UKF

Combining Different Measurement Types

Apart from combining successive measurements, the TDoA system needs, as a backup especially if the TDoA measurements are highly inaccurate, a way to combine different kinds of measurements. In Section 3.2.3, the UKF was applied to combine TDoA and RSSD measurements. It is of interest to know how much better the estimation can get using the additional measurements and whether the UKF is a good estimator for that case.

A simulated scenario consisting of five sensors placed on a circle with a radius of 1000 meters and a transmitter that was placed randomly for every realization is defined. For the initial estimate of the UKF, 20 averaged TDoA vectors are used in a least squares algorithm. Figure 3.10 shows the RMSE over growing number of sequential measurements for the UKF with TDoAs only, the hybrid UKF as well as the hybrid CRLB. It can be seen that for a small standard deviation of the RSSD measurements, there is an obvious improvement compared to using TDoA only. On the other hand, the UKF approaches the CRLB for hybrid measurements, meaning that it is a good candidate for combining the measurements. For the case of small RSSD errors, the UKF shows an error floor of 25 meters and is therefore not an efficient estimator.

On the left side, figure 3.11 shows the RMSE of the estimate for growing RD standard deviation and growing RSSD standard deviation after 100 Kalman steps; on the right side, the square root of the CRLB can be seen. For large

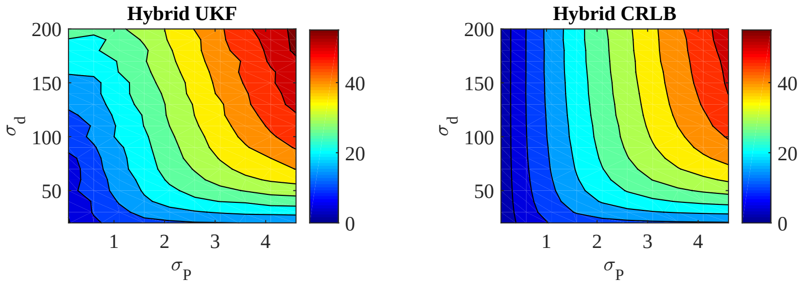


Figure 3.11: RMSE of the hybrid UKF at different TDoA and RSSD standard deviations compared to the CRLB

errors in RD measurements, using additional RSSD measurements improves the result. In all cases, statistically, it will never be worse than using TDoAs only since information is being added to the estimation problem. On the other hand, for reliable RDs with small standard deviations, the additional RSSDs make a small difference. A realistic scenario where applying the hybrid system is needed, would be a defect in the system clock that results in large RD errors, while RSSD measurements can produce good results. The smart TDoA system should detect when its RD measurements are highly erroneous to compensate that by using other measurements.

Tracking

The last scenario to be analyzed is for the case of a slowly moving transmitter. As mentioned in Chapter 1 and 2, the TDoA system should be able to localize stationary and slowly moving transmitters. For the static scenario, using a Kalman filter has no benefit in the estimation accuracy, unless large outliers are expected. But in case the transmitter is moving, Kalman filters continue to be a good tracking method. Figure 3.12 shows two tracks for the transmitter, where the positions are around 1.5 meters apart. The EKF is given TDoA measurements with $\sigma_d = 50$ m as well as an initial estimate obtained from averaging the first 20 estimated TDoAs. In case of an unknown velocity, which is used in the scenario, the EKF constantly assumed process noise since the unknown velocity keeps changing. Therefore, the accuracy stays at a level of 23 meters for both tracks. A known velocity or a smaller measurement noise would increase the accuracy.

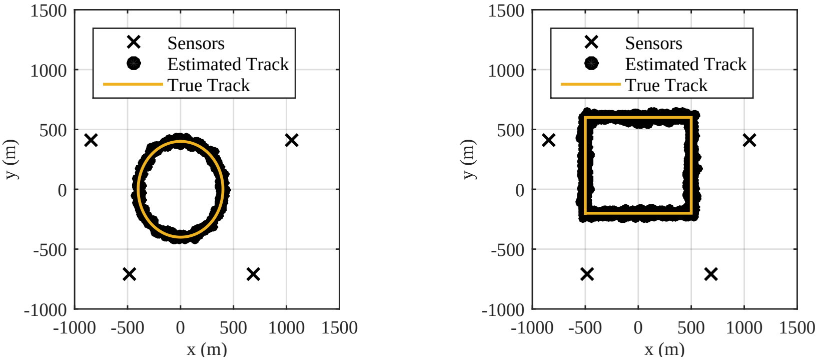


Figure 3.12: Tracking using the EKF

3.3 System Design

Based on the gained knowledge about different aspects that need to be considered, a smart TDoA system can be designed for the AWGN scenario. Figure 3.13 shows the block diagram of the smart TDoA system.

The system receives N time-synchronized and sampled signals $r_1[n], r_2[n], ...,$ $r_N[n]$. In the first step, these signals are preprocessed. For the case of unknown signal spectra, a lowpass filter can be implemented, reducing the noise around the signal spectrum. Additionally, the signals can be upsampled to achieve a higher time estimation accuracy. The preprocessed signals are then used to estimate TDoAs by cross-correlating and interpolating around the detected maximum. For choosing an initial reference sensor, all signals are cross-correlated and the signal that leads to the highest correlation peaks is chosen as reference sensor. Later on, the estimated position as well as the estimated covariance matrix can be used to choose the best reference as was described in (3.62). After estimating the RDs and the corresponding covariance matrix, the position can be obtained by applying the TSE-algorithm and, if it fails, using Chan's algorithm. For a number of M sequential measurements, the new position is estimated after averaging over all estimated RDs for each sensor. The Mth estimate is then used as an initial Kalman state to track the signal by re-estimating the position and the velocities for each measurement. Whenever the TDoA estimation accuracy is too low (e.g., for defect clocks), RSSD measurements can be used to track the transmitter using the UKF as was described.

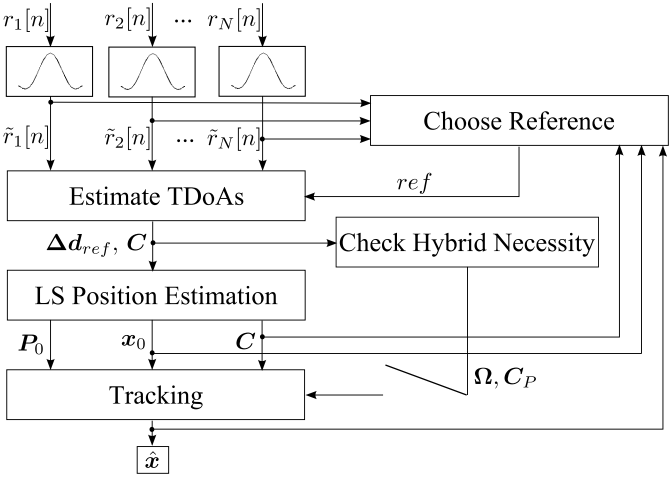


Figure 3.13: Smart TDoA system in AWGN

4 TDoA in Multipath Channels and NLoS

Chapter 3 dealt with utilizing signals that propagate through AWGN channels to passively estimate the position of a signal source using TDoA. In the AWGN scenario, the delay estimation accuracy is affected by noise, reducing the overall positioning accuracy. Aiming at designing a system that can be applicable for a variety of scenarios, the AWGN channel model, assuming no reflection on buildings or other objects, can not be considered solely.

This chapter handles the second and more challenging scenario facing the positioning system: multipath propagation. In the same manner as the last chapter, this problem is treated by observing its affect on the TDoA estimation and on the position estimation problem separately.

4.1 Time Delay Estimation in Multipath Channels

The task of estimating TDoAs from noise-corrupted signals is feasible. By cross-correlating signal pairs and searching for the correlation peak, unbiased

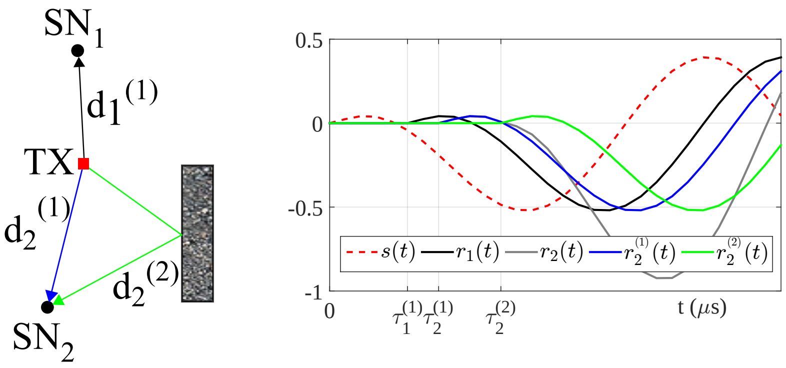


Figure 4.1: A multipath scenario with $d_1^{(1)} = 150$ m , $d_2^{(1)} = 210$ m , $d_2^{(2)} = 330$ m

TDoA estimates can be obtained. In multipath scenarios, this turns into a challenging task, especially when the signal paths arrive closely in time, compared to the corresponding signal correlation time.

Consider for instance a scenario depicted in Figure 4.1, where a signal is transmitted from source TX and received at both sensors SN_1 and SN_2. The signal is composed of random lowpass filtered data, generated at a rate of 1 MSps. SN_1 receives the direct signal path only, while SN_2 receives two signal replica arriving from a direct and a reflected path. The distance traveled by the pth signal component arriving at the lth sensor (whether it is the direct way or the detour due to reflection) is given below the figure as $d_l^{(p)}$. On the right side of the figure, signals received by the two sensors can be seen. While SN_1 receives only one component, SN_2 receives two replica of the signal. The figure shows the single components $r_2^{(1)}(t), r_2^{(2)}(t)$ as well as the actually received signal $r_2(t)$ which results from summing both components. Throughout this chapter, the subscript $(.)^{(p)}$ will be used to indicate parameters of the pth replica of the propagated signal, resulting from multipath.

Figure 4.2 shows the cross-correlation of the single components as well as the factual cross-correlation of the received signals at SN_1 and SN_2. The resulting correlation is the sum of both single correlations. However, since the single components are not known, they can not be separated. Therefore, a conventional cross-correlation and peak search would lead to the wrong delay estimation value, even in the noise-free case, since the sum of both correlation

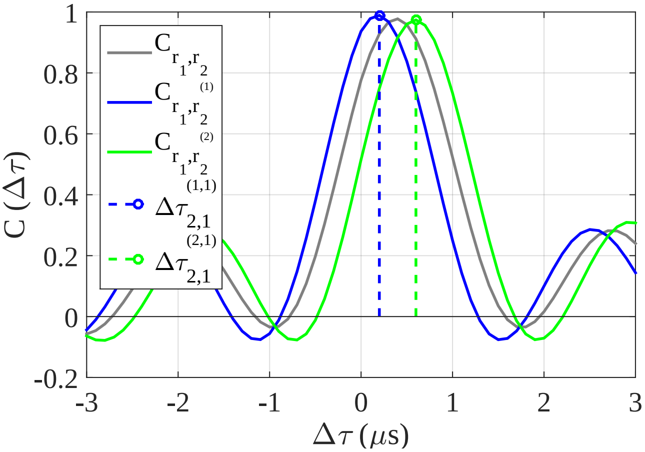


Figure 4.2: The effect of multipath on the cross-correlation

components leads to a peak shift. The closer the delays and/or the wider the correlation peak, the higher the error caused by additional arriving components. If the second component had a much larger detour distance, the cross-correlation would have two obvious correlation-peaks. The separation of the replica would be an easy task and the error caused by the superposition of both components would be negligible. On the other hand, if the transmitted signal had a larger bandwidth, its correlation peak would be narrower, also leading to an obvious peak separation. Since the localization scenario does not assume wide-band signals, the separation of multipath replica is one of the biggest challenges of the system.

In this section, methods aiming at resolving multipath delays are investigated. The ML solution for passive TDoA (i.e., with an unknown transmitted signal) as well as the CRLB are derived. Two applicable algorithms for the passive system are presented. Finally, specific scenarios are simulated to compare the algorithms to each other as well as to the theoretical bound.

4.1.1 System Model

A signal impinging on a sensor i after propagating through a multipath channel can be modeled as [38]

$$r_i(t) = \sum_{p=1}^{P_i} \alpha_i^{(p)} s\left(t - \tau_i^{(p)}\right) + w_i(t), \tag{4.1}$$

where $s(t)$ is the transmitted signal, P_i is the number of multipath components, $\alpha_i^{(p)}(t)$ and $\tau_i^{(p)}$ are the fading coefficient and the delay value of the pth path and $w_i(t)$ is a zero-mean white circular Gaussian noise. Since the described positioning scenario is concerned with stationary or slowly moving transmitters, the underlying assumption behind the model in (4.1) is that the channel parameters are constant throughout the observation period.

Assuming that $\tau_i^{(1)} < \tau_i^{(2)}, ..., < \tau_i^{(P_i)}$, the desired TDoA fulfilling the geometrical equation, results from the delay difference of the first, direct path delays as

$$\begin{aligned}\Delta d_{i,j}(\boldsymbol{x}) &= d_i^{(1)} - d_j^{(1)} \\ &= c \cdot \Delta\tau_{i,j}^{(1,1)} \\ &= c \cdot \left(\tau_i^{(1)} - \tau_j^{(1)}\right).\end{aligned} \tag{4.2}$$

Methods aiming at resolving the multipath delays can be found in the literature, varying in their computational complexity, their required *a priori* information, their ability to resolve closely spaced paths as well as their accuracy. The different approaches to solving the problem of time delay estimation of multipath propagated signals can be categorized in [2]

- Deconvolution-based methods: The deconvolution approach requires a known signal $s(t)$. Using this knowledge, the channel information (including the delays) can be extracted by dividing the received signal by the known one in the frequency domain [39]. Apart from the required signal knowledge, this method suffers from noise amplification as a result of the division operation, thus, requiring additional processing.

- Subset-based methods: Based on the covariance matrix model for sinusoidal signals in white noise [40], the subset-based methods extract the sinusoidal signal model by dividing the received signal by the known one in the frequency domain. This results in noise coloring, defying the important premise of white noise. Popular algorithms include the MUSIC [41] and the ESPRIT [42] algorithm.

- ML-based methods: The ML solution is the only one not requiring *a priori* information [43], making it a good candidate for the passive localization system. Although the original method requires an extensive multidimensional search over all unknown parameters (i.e., all delays and gains), different approaches have been presented reducing the computational complexity of the search. Therefore, two ML-based algorithms are implemented for the passive TDoA system. Next, the ML-system model is derived.

For the derivation of the ML-model, the case of an active localization system (i.e., with a known transmit signal) is described first. Assuming that all delays are different multiples of the sampling interval (i.e., $\tau_i^{(p)} = nT_{\text{samp}}, n \in \mathbb{Z}$), the DFT of the received signals can be calculated, enabling the differentiation of the signal by the parameters, leading to

$$R_i[k] = \sum_{p=1}^{P_i} \alpha_i^{(p)} S[k] \mathrm{e}^{\frac{-\mathrm{j}2\pi\tau_i^{(p)}k}{K}} + W_i[k], \quad k = 0, 1, ..., K-1, \tag{4.3}$$

where $W_i[k] \sim \mathcal{CN}(0, \sigma_{W_i}^2)$ and $\sigma_{W_i}^2 = K\sigma_{w_i}^2$. In vector form, (4.3) can be rewritten as

$$\boldsymbol{R}_i = \boldsymbol{\Phi}_A(\boldsymbol{\tau}_i)\boldsymbol{\alpha}_i + \boldsymbol{W}_i, \tag{4.4}$$

where

$$\boldsymbol{\alpha}_i = \left[\alpha_i^{(1)}, \alpha_i^{(2)}, ..., \alpha_i^{(P_i)}\right]^T \tag{4.5}$$

$$\boldsymbol{\tau}_i = \left[\tau_i^{(1)}, \tau_i^{(2)}, ..., \tau_i^{(P_i)}\right] \tag{4.6}$$

$$\boldsymbol{\Phi}_A(\boldsymbol{\tau}_i) = \left[\boldsymbol{a}_A\left(\tau_i^{(1)}\right), \boldsymbol{a}_A\left(\tau_i^{(2)}\right), ..., \boldsymbol{a}_A\left(\tau_i^{(P_i)}\right)\right] \tag{4.7}$$

$$\boldsymbol{a}_A(\tau_i^{(p)}) = \left[S[0], S[1]\mathrm{e}^{\frac{-\mathrm{j}2\pi\tau_i^{(p)}}{K}}, ..., S[K-1]\mathrm{e}^{\frac{-\mathrm{j}2\pi\tau_i^{(p)}(K-1)}{K}}\right]^T. \tag{4.8}$$

The probability density function of the observed signal can be described as a function of the unknown parameters $\alpha_i^{(p)}, \tau^{(p)}$ as

$$p(\boldsymbol{R}_i|\boldsymbol{\alpha}_i, \boldsymbol{\tau}_i) = \frac{1}{\pi^K |\boldsymbol{\Sigma}_i|} \mathrm{e}^{-(\boldsymbol{R}_i - \boldsymbol{\Phi}_A(\boldsymbol{\tau}_i)\boldsymbol{\alpha}_i)^H \boldsymbol{\Sigma}_i^{-1}(\boldsymbol{R}_i - \boldsymbol{\Phi}_A(\boldsymbol{\tau}_i)\boldsymbol{\alpha}_i)}, \tag{4.9}$$

where

$$\boldsymbol{\Sigma}_i = \boldsymbol{I}_K \sigma^2_{W_i}. \tag{4.10}$$

The lower index $(.)_A$ refers to the case of a known signal, i.e., the active case. Two possible approaches emerge from this model when dealing with an unknown signal:

- The unknown signal can be included in the unknown parameters of the likelihood function $p(\boldsymbol{R}_i|\boldsymbol{\alpha_i}, \boldsymbol{\tau_i}, \boldsymbol{S})$, leading to a far more complex problem that is unsolvable.
- The unknown signal can be replaced by a reference signal, i.e., a signal received by one of the sensors. To simplify the notation, SN_1 is chosen as the reference sensor. This approach is more feasible and is therefore chosen and described next.

Assuming that the reference signal has only one component, i.e., $r_1(t) = \alpha_1^{(1)} s\left(t - \tau_1^{(1)}\right) + w_1(t)$, (4.1) can be rewritten as

$$R_j[k] = \sum_{p=1}^{P_j} \beta_j^{(p)} R_1[k] \mathrm{e}^{\frac{-\mathrm{j}2\pi\Delta\tau_j^{(p)} k}{K}} + \tilde{W}_j[k], \quad j = 2, 3, ..., N, \tag{4.11}$$

where

$$\beta_j^{(p)} = \frac{\alpha_j^{(p)}}{\alpha_1^{(1)}} \tag{4.12}$$

$$\Delta\tau_j^{(p)} = \tau_j^{(p)} - \tau_1^{(1)} \tag{4.13}$$

$$\tilde{W}_j[k] = W_j[k] - \sum_{p=1}^{P_j} \beta_j^{(p)} W_1[k] \mathrm{e}^{\frac{-\mathrm{j}2\pi\Delta\tau_j^{(p)} k}{K}}. \tag{4.14}$$

The assumption about the reference signal is not required to derive this model. Without it, the number of unknown parameters would merely grow since the number of unknown channel parameters in (4.11) is $2P_jP_1$. This assumption has the advantage of making the TDoA estimation problem a direct result of the estimation of the channel parameters. Under this premise, the desired TDoA is always given by $\Delta\tau_{j,1}^{(1)}$ since larger delay differences originate from longer traveled distances from the transmitter to the sensors. Without this assumption, a higher delay difference can not be directly linked to larger distances from the transmitter to the sensors, only to larger distance differences. Therefore, if this condition does not hold, additional algorithms are needed to identify the delay corresponding to the true TDoA as it was done in [44] and [45]. In this work, it is assumed that at least one of the sensors has a strong Line-of-Sight (LoS) component and that this sensor can be identified and defined as the reference sensor.

Analyzing the expressions in (4.11), $\tilde{W}_j[k]$ is a complex zero mean white Gaussian noise term with $\tilde{W}_j[k] \sim \mathcal{CN}\left(0, \sigma^2_{W_j} + \sigma^2_{W_1}\left|\sum_{p=1}^{P_j} \beta_j^{(p)} \mathrm{e}^{\frac{-\mathrm{j}2\pi\Delta\tau_j^{(p)}k}{K}}\right|^2\right)$ (see Appendix B). Hence, the noise on each frequency bin of the discrete Fourier transformed signal is white, but each bin has a different noise variance consisting of the noise on the jth signal as well as the noise of the reference signal multiplied by a factor, depending on the frequency bin and on the channel parameters.

The resulting likelihood function for the parameter vector $\boldsymbol{\theta}_j = \left[\boldsymbol{\Delta\tau}_j, \boldsymbol{\beta}_j^T\right]$ is given by

$$p(\boldsymbol{R}_j|\boldsymbol{\theta}_j) = \frac{1}{\pi^K|\tilde{\boldsymbol{\Sigma}}_j|}\mathrm{e}^{-(\boldsymbol{R}_j - \boldsymbol{\Phi}_P(\boldsymbol{\Delta\tau}_j)\boldsymbol{\beta}_j)^H \tilde{\boldsymbol{\Sigma}}_j^{-1}(\boldsymbol{R}_j - \boldsymbol{\Phi}_P(\boldsymbol{\Delta\tau}_j)\boldsymbol{\beta}_j)}, \tag{4.15}$$

where the elements of the covariance matrix $\tilde{\boldsymbol{\Sigma}}_i$ are

$$\tilde{\boldsymbol{\Sigma}}_j = \begin{pmatrix} \sigma_{\tilde{W}_j} & & & \\ & \sigma^2_{\tilde{W}_j}[1] & & \\ & & \ddots & \\ & & & \sigma^2_{\tilde{W}_j}[K-1] \end{pmatrix} \tag{4.16}$$

and

$$\sigma^2_{\tilde{W}_j}[k] = \sigma^2_{W_j} + \sigma^2_{W_1}\left|\sum_{p=1}^{P_j} \beta_j^{(p)} \mathrm{e}^{\frac{-\mathrm{j}2\pi\Delta\tau_j^{(p)}k}{K}}\right|^2. \tag{4.17}$$

$\boldsymbol{\Phi}_P(\boldsymbol{\Delta\tau}_j)$ denotes the signal vector in case of a passive system, resulting from replacing $S[k]$ in (4.8) by $R_1[k]$, the known reference signal.

Compared to the likelihood in case of an active system, the expected value is now a function of the reference signal. Delays and gains are replaced by the delay differences and the gain ratios. The main difference between both models is the covariance matrix.

One practical challenge emerging from the maximization of (4.15) is the parameter dependent covariance matrix, increasing the complexity of the problem. By assuming $\tilde{\boldsymbol{\Sigma}}_j \approx \boldsymbol{I}_K \tilde{\sigma}^2$, an approximated ML solution is given by minimizing the following cost function

$$\hat{\boldsymbol{\theta}}_j = \arg \min_{\boldsymbol{\Delta\tau}_j, \boldsymbol{\beta}_j} ||\boldsymbol{R}_j - \boldsymbol{\Phi}_P(\boldsymbol{\Delta\tau}_j)\boldsymbol{\beta}_j||^2 . \tag{4.18}$$

Hence, due to the approximations behind (4.18), it can not be labeled the ML solution but it will be referred to as an ML-based solution. In fact, (4.18) is a least squares solution based on the passive model in (4.11).

A straightforward solution to (4.18) would be a grid search algorithm, with the grid points being all possible parameter vectors. Such a search would be in a $2P_j$-dimensional space (assuming that all β_j are real, otherwise a $3P_j$-dimensional space) and would require a large number of grid points. For this reason, algorithms based on the above model try to reduce computational cost, while maintaining a high estimation accuracy. Two well used search algorithm classes are Monte Carlo and iterative algorithms.

The basic idea of the Monte Carlo solution is to simplify the search by generating samples or realizations from the function of interest (here: (4.15)) and estimating the maximum using these samples. Popular techniques are importance sampling [46], pure random search [47] and its faster version, the accelerated random search [48]. For the TDoA system, a method based on the importance sampling technique was chosen. This method presented in [49] is suitable for the passive scenario, showing a promising accuracy and an acceptable computational complexity. The approach will be described in Section 4.1.2.

As an alternative, iterative algorithms often start with an initial estimate and iteratively search for the minimum of (4.18). Common algorithms in that category are the Expectation Maximization (EM) algorithm [50], simulated annealing [51], and a relaxation-based algorithm [52]. Although the EM algorithm is known to be able to obtain the most accurate results, its performance is highly dependent on the initial guess, making it less reliable. For the TDoA system, the relaxation-based algorithm presented in [52] is chosen since it operates closely to the EM algorithm with no requirement of an initial estimate. It will be presented in Section 4.1.3.

4.1.2 ML-based Solution Using Importance Sampling

The idea of the Importance sampling algorithm was presented in [53] for direction of arrival estimation. In [49] and [54], this idea was applied to time delay estimation. It proceeds from (4.18) by assuming that the nuisance parameter vector $\boldsymbol{\beta}_j$ is deterministic but unknown, making it possible to solve for $\boldsymbol{\beta}_j$ at the true delay vector as

$$\hat{\boldsymbol{\beta}}_j(\boldsymbol{\Delta\tau}_j) = (\boldsymbol{\Phi}_P^H(\boldsymbol{\Delta\tau}_j)\boldsymbol{\Phi}_P(\boldsymbol{\Delta\tau}_j))^{-1}\boldsymbol{\Phi}_P^H(\boldsymbol{\Delta\tau}_j)\boldsymbol{R}_j. \tag{4.19}$$

Inserting (4.19) in (4.18) and omitting all irrelevant terms results in

$$L_c(\boldsymbol{\Delta\tau}_j) = \boldsymbol{R}_j^H\boldsymbol{\Phi}_P(\boldsymbol{\Delta\tau}_j)(\boldsymbol{\Phi}_P^H(\boldsymbol{\Delta\tau}_j)\boldsymbol{\Phi}_P(\boldsymbol{\Delta\tau}_j))^{-1}\boldsymbol{\Phi}_P^H(\boldsymbol{\Delta\tau}_j)\boldsymbol{R}_j, \tag{4.20}$$

depending only on the delay vector. For active systems, this function is referred to as the compressed likelihood function. The search is now reduced to a P_j-dimensional search, thus still remaining a challenging task. Two mathematical tools are utilized to solve this problem. The first tool is Pincus theorem presented in [55], stating that the global maximum of the multidimensional function $L_c(\boldsymbol{\Delta\tau})$ is given by

$$\widehat{\Delta\tau}^{(p)} = \lim_{\rho\to\infty} \frac{\int_J ... \int_J \Delta\tau^{(p)} \exp{(\rho L_c(\boldsymbol{\Delta\tau}))} d\boldsymbol{\Delta\tau}}{\int_J ... \int_J \exp{(\rho L_c(\widetilde{\boldsymbol{\Delta\tau}}))} d\widetilde{\boldsymbol{\Delta\tau}}}, \quad p = 1, 2, ..., P_j, \tag{4.21}$$

where J is the interval including all possible values for $\boldsymbol{\Delta\tau}$.

Defining

$$L'_{c,\rho_0}(\boldsymbol{\Delta\tau}) = \frac{\exp{(\rho_0 L_c(\boldsymbol{\Delta\tau}))}}{\int_J ... \int_J \exp{(\rho_0 L_c(\widetilde{\boldsymbol{\Delta\tau}}))} d\widetilde{\boldsymbol{\Delta\tau}}}, \tag{4.22}$$

the optimal value of $\Delta\tau^{(p)}$ is

$$\widehat{\Delta\tau}^{(p)} \approx \int_J ... \int_J \Delta\tau^{(p)} L'_{c,\rho_0}(\boldsymbol{\Delta\tau}) d\boldsymbol{\Delta\tau}, \quad p = 1, 2, ..., P_j, \tag{4.23}$$

where ρ_0 is some large number. (4.23) has the form of a mean operation with L'_{c,ρ_0} being the according pseudo probability density function. If one is able to generate random samples from L'_{c,ρ_0}, then (4.23) can be solved by approximating the integral using a sum over generated samples from L'_{c,ρ_0}.

Still, generating samples from a multidimensional probability distribution function is not an easy task. Here, the second mathematical tool, a Monte Carlo technique called importance sampling [46] is used, enabling the generation of random samples from the computationally complex function L'_{c,ρ_0} by approximating it using a well defined function, the so called importance function. The approach is based on the following transformation of (4.23) to

$$\widehat{\Delta\tau}^{(p)} = \int_J \dots \int_J \boldsymbol{\Delta\tau} \frac{L'_{c,\rho_0}(\boldsymbol{\Delta\tau})}{g'(\boldsymbol{\Delta\tau})} g'(\boldsymbol{\Delta\tau}) d\Delta\tau. \tag{4.24}$$

This equation can be interpreted as the mean of $\Delta\tau^{(p)} \frac{L'_{c,\rho_0}(\boldsymbol{\Delta\tau})}{g'(\boldsymbol{\Delta\tau})}$ when $\boldsymbol{\Delta\tau}$ is generated according to $g'(\boldsymbol{\Delta\tau})$. Therefore, by defining a well chosen function as $g'(.)$, the maxima of L_{c,ρ_0} can be searched for by approximating the integral in (4.24) by

$$\widehat{\Delta\tau}^{(p)} = \frac{1}{M} \sum_{m=1}^{M} \boldsymbol{\Delta\tau}[m] \frac{L'_{c,\rho_0}(\boldsymbol{\Delta\tau}[m])}{g'(\boldsymbol{\Delta\tau}[m])}, \tag{4.25}$$

where $\boldsymbol{\Delta\tau}$ is generated according to $g'(.)$.

Since the correlation behaves similarly to the likelihood (i.e., peaking around the true delay parameters), it was chosen as the foundation of the importance function, thus, reducing the dimension to one. Overall, the algorithm steps for estimating the relative delays $\boldsymbol{\Delta\tau}_j$ of the jth sensor while using sensor one as reference sensor are [49], [54]:

1. Compute the DFT of the two signals $r_1[n], r_j[n]$.

2. Evaluate the following function for all possible delay values in the interval J

$$I(\Delta\tau[l]) = \left| \sum_{k=0}^{K-1} R_1[k] R_j^*[k] \exp\left(\frac{\mathrm{j}2\pi k \Delta\tau[l]}{K} \right) \right|^2, \quad \Delta\tau[l] \in J. \tag{4.26}$$

3. Define the sampled one-dimensional probability density function $g_{\rho_1}(.)$ as

$$g_{\rho_1}(\Delta\tau[l]) = \frac{\exp\left(\rho_1 I(\Delta\tau[l])\right)}{\sum\limits_{k=1}^{L} \exp\left(\rho_1 I(\Delta\tau[k])\right)}, \quad l = 1, 2, ..., L, \tag{4.27}$$

where L is the total number of points in the interval J, determining the resolution of the importance function.

4. Generate M sample vectors of the form $\boldsymbol{\Delta\tau}[m] = \left[\Delta\tau[m]^{(1)}, \Delta\tau[m]^{(2)}, \ldots, \Delta\tau[m]^{(P_j)}\right]$, $m = 1, 2, ..., M$ using $g_{\rho_1}(.)$ as the probability density function.

5. Calculate a weighting factor for each generated $\boldsymbol{\Delta\tau}[m]$-vector depending on the likelihood-based value of (4.20), i.e., delay vectors resulting in higher likelihood values obtain larger weights, as

$$\begin{aligned} F(\boldsymbol{\Delta\tau}[m]) = \exp\Bigg\{ & \rho_0 L_c(\boldsymbol{\Delta\tau}[m]) - \rho_1 \sum_{p=1}^{P_j} I\left(\Delta\tau[m]^{(p)}\right) \\ & - \max_{1 \le l \le M} (\rho_0 L_c(\boldsymbol{\Delta\tau}[l]) - \rho_1 \sum_{p=1}^{P_j} I\left(\Delta\tau[l]^{(p)}\right) \Bigg\}. \end{aligned} \tag{4.28}$$

The weighting factor results from reforming the factor remaining in (4.25) $\frac{L'_{c,\rho_0}(\boldsymbol{\Delta\tau}[m])}{g'(\boldsymbol{\Delta\tau}[\boldsymbol{m}])}$ to avoid numerical instabilities.

6. Use the generated parameter vectors and the weights to calculate the circular mean, resulting in the estimated delay vector following

$$\widehat{\Delta\tau}_j^{(p)} = \frac{\tau_{\text{max}}}{\pi} \angle \left[\frac{1}{M} \sum_{m=1}^{M} F(\Delta\tau[m]) \exp\left(\frac{\mathrm{j}2\pi\Delta\tau[m]^{(p)}}{\tau_{\text{max}}} \right) \right], \tag{4.29}$$

where τ_{max} is the maximum possible delay value. Applying the circular mean instead of the linear mean was justified by the fact that the range of possible delay values is normally known and can be incorporated to enhance the estimation accuracy.

The parameters of the algorithms are ρ_0 and ρ_1. They determine the performance of the algorithm and must be chosen carefully. Their effect as well as the best values will be discussed in the result section. One drawback of this algorithm is that it requires knowledge of the number of paths P_j. This problem can be solved by using the Minimum Description Length (MDL) algorithm, which will be presented in Section 4.1.4.

4.1.3 ML-based Solution Using the WRELAX Algorithm

As an alternative to the Monte Carlo based method in Section 4.1.2, an iterative approach is presented. Opposed to most iterative methods, the chosen algorithm presented in [52] does not require an initial estimate and is therefore, another good candidate for the passive localization scenario. It was described for the active scenario, i.e., with a known signal, but can be extended directly to the passive scenario. Based on a relaxation approach, the method proceeds by simplifying the multidimensional search in (4.18) to a one-dimensional search. Based on a weighted Fourier transform and relaxation approach, the algorithm was referred to as the Weighted Fourier Transform and Relaxation (WRELAX) algorithm. The idea of the algorithm is based on minimizing the following cost function as a function of the parameter vector $\boldsymbol{\theta}_j = [\boldsymbol{\Delta\tau}_j, \boldsymbol{\beta}_j^T]$

$$D(\boldsymbol{\theta}_j) = ||\boldsymbol{R}_j - \boldsymbol{\Phi}_P(\boldsymbol{\Delta\tau}_j)\boldsymbol{\beta}_j||^2 . \tag{4.30}$$

The key equation for this approach is

$$\hat{\boldsymbol{R}}_j^{(l)} = \boldsymbol{R}_j - \sum_{p=1,p\neq l}^{P_j} \beta_j^{(p)} \boldsymbol{a}_P \left(\Delta\tau_j^{(p)}\right), \tag{4.31}$$

where $\left\{\beta_j^{(p)}, \Delta\tau_j^{(p)}\right\}_{p=1,p\neq l}^{P_j}$ are assumed to be given [1]. (4.30) can be expressed by

$$D\left(\Delta\tau_j^{(l)}, \beta_j^{(l)}\right) = \left|\left|\boldsymbol{R}_j^{(l)} - \beta_j^{(l)} \boldsymbol{a}_P \left(\Delta\tau_j^{(l)}\right)\right|\right|^2, \tag{4.32}$$

where $D\left(\Delta\tau_j^{(l)}, \beta_j^{(l)}\right)$ refers to the cost-function depending on the parameters of the lth multipath component. By solving for $\beta_j^{(l)}$ at the true $\Delta\tau^{(l)}$ and reinserting in (4.32), minimizing the cost function yields

$$\widehat{\Delta\tau}_j^{(l)} = \arg\max_{\Delta\tau} \left|\left|\boldsymbol{a}_P \left(\Delta\tau^{(l)}\right)^H \boldsymbol{R}_j^{(l)}\right|\right|^2 \tag{4.33}$$

$$\hat{\beta}_j^{(l)} = \left.\frac{\boldsymbol{a}_P\left(\Delta\tau\right)^H \boldsymbol{R}_j^{(l)}}{\boldsymbol{R}_1^H \boldsymbol{R}_1}\right|_{\Delta\tau=\widehat{\Delta\tau}_j^{(l)}} . \tag{4.34}$$

Based on this one-dimensional search, the algorithm proposed in [52] is demonstrated in Figure 4.3.

Assuming L signal components, $\hat{\beta}_j^{(L)}, \widehat{\Delta\tau}_j^{(L)}$ can be obtained by applying (4.31), (4.34) and (4.33). Afterwards, previously obtained parameters for $\hat{\beta}_j^{(1)}$,

[1] Note that $\boldsymbol{R}_j^{(l)}$ here refers to the lth-multipath component of the received signal at sensor j.

$\widehat{\Delta\tau}_j^{(1)}, \ldots, \hat{\beta}_j^{(L-1)}, \widehat{\Delta\tau}_j^{(L-1)}$ can be re-estimated using the same equations. Starting $L = 3$, this step is repeated iteratively m times, until the enhancement in the cost function from D_m to D_{m+1} (see (4.30)) is small (e.g., by setting $\epsilon = 0.001$). The value of L is incremented until it reaches the given number of signal components P_j.

Larger ϵ values lead to faster convergence but result in less accurate estimates. The WRELAX algorithm, like the Importance Sampling (IS) algorithm requires a known number of paths. By integrating the MDL algorithm within the WRELAX, the number of paths can be estimated. Therefore, the MDL is described next for both algorithms.

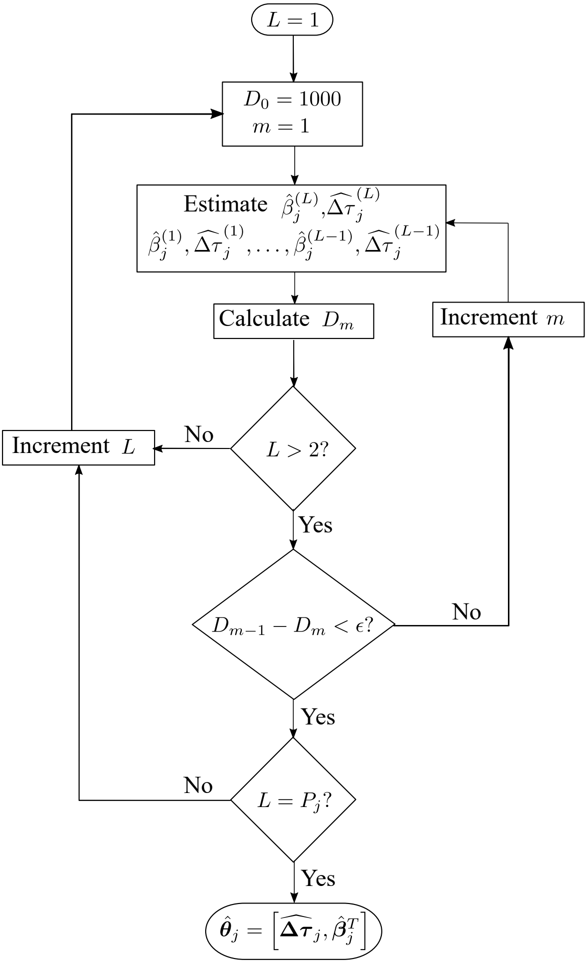

$L = 1$
$D_0 = 1000$
$m = 1$
Estimate $\hat{\beta}_j^{(L)}, \widehat{\Delta\tau}_j^{(L)}$
$\hat{\beta}_j^{(1)}, \widehat{\Delta\tau}_j^{(1)}, \ldots, \hat{\beta}_j^{(L-1)}, \widehat{\Delta\tau}_j^{(L-1)}$
Calculate D_m
Increment m
Increment L
No
$L > 2?$
Yes
$D_{m-1} - D_m < \epsilon?$
No
Yes
No
$L = P_j?$
Yes
$\hat{\boldsymbol{\theta}}_j = \left[\widehat{\boldsymbol{\Delta\tau}}_j, \hat{\boldsymbol{\beta}}_j^T\right]$

Figure 4.3: WRELAX algorithm

4.1.4 The Minimum Description Length Method

The MDL method is, like its name indicates, a method to find the minimum number of variables needed to describe an observed sequence. It was originally presented by Rissanen in [56]. Later on, its application to multipath channel parameter estimation was presented in [57]. The minimum description length for a received signal sequence $\boldsymbol{x} = x_1, x_2, ..., x_K$ and an unknown parameter vector $\boldsymbol{\theta}$ is found by minimizing the following expression

$$M(L) = -\log\left[p(\boldsymbol{x}|\boldsymbol{\theta}_{\text{max},L})\right] + \frac{1}{2}L\log\left(K\right), \tag{4.35}$$

where L is the number of unknown parameters and $p(\boldsymbol{x}|\boldsymbol{\theta}_{\text{max},L})$ is the maximum likelihood value, assuming L parameters. The relation to the likelihood value is intuitive. The more the received signal fits to the model using $\boldsymbol{\theta}_L$, the more likely this parameter vector is the true one. On the other hand, the second term in the sum is a penalty term to avoid over-parameterization.

For the two described algorithms in this section, (4.35) results to

$$\tilde{M}_{\text{IS}}(P) = A + \frac{1}{2}P\log K \tag{4.36}$$

$$\tilde{M}_{\text{WRELAX}}(P) = A + P\log K, \tag{4.37}$$

where

$$A = \left(\boldsymbol{R} - \boldsymbol{\Phi}_P(\widehat{\boldsymbol{\Delta\tau}})\hat{\boldsymbol{\beta}}\right)^H \hat{\boldsymbol{\Sigma}}^{-1}\left(\boldsymbol{R} - \boldsymbol{\Phi}_P(\widehat{\boldsymbol{\Delta\tau}})\hat{\boldsymbol{\beta}}\right). \tag{4.38}$$

Since the WRELAX estimates the gains and delays, the number of unknown parameters is $2P$, which is why the second term in the sum is doubled compared to the IS.

The MDL can be integrated simply in both algorithms. Assuming a maximum number of paths P_{max}, the complete algorithm for time delay estimation in multipath channels with no knowledge about the number of multipath components is presented in Figure 4.4. After estimating the parameters $\tilde{\boldsymbol{\theta}}(P)$ assuming P components and using one of the two previously described algorithms, $M(P)$ can be calculated by (4.37) or (4.36), depending on the applied algorithm. When the maximum number of paths is reached, $\hat{P}$ is chosen as the value that minimizes $M(P)$ and the parameter vector is the corresponding vector assuming $\hat{P}$ components.

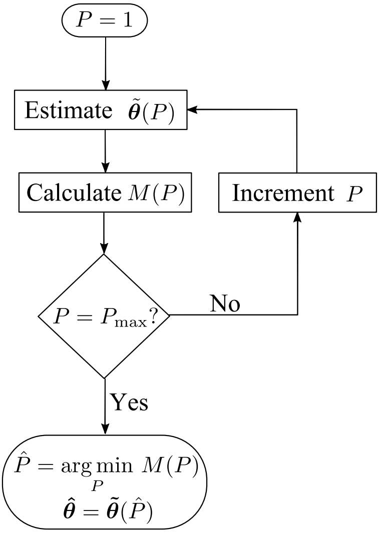

$P = 1$
Estimate $\tilde{\boldsymbol{\theta}}(P)$
Calculate $M(P)$
Increment P
$P = P_{\max}$?
No
Yes
$\hat{P} = \arg\min_{P} M(P)$
$\hat{\boldsymbol{\theta}} = \tilde{\boldsymbol{\theta}}(\hat{P})$

Figure 4.4: Time delay estimation with unknown number of multipath components

4.1.5 Estimation Bounds

In Section 3.1.4, the estimation bound for time delay in AWGN channels was discussed. The CRLB was chosen due to its simple form and its popularity, making it suitable for general comparisons. For multipath propagation, the evaluation of the expression (A.6) is a challenging task and can not be solved simply. Here, the likelihood function needed for the derivation of the CRLB does not depend on one parameter, but on a parameter vector $\boldsymbol{\theta}$ consisting of all unknown delays and gains defined here as $\boldsymbol{\theta} = [\theta_1, \theta_2, ..., \theta_M]$.

We follow here the derivation of the CRLB for a complex Gaussian probability density function of the form given in (A.8) presented in [28] and apply this to the multipath model in (4.15).

The CRLB for the ith parameter of the vector $\boldsymbol{\theta}$ is given as

$$\sigma_i^2 \geq [\boldsymbol{F}(\boldsymbol{\theta})^{-1}]_{ii}, \tag{4.39}$$

where $\boldsymbol{F}$ is the $2P \times 2P$ Fisher Information Matrix (FIM), assuming real-valued $\beta^{(p)}$. For complex-valued gains, the FIM becomes a $3P \times 3P$ matrix, where the real and imaginary parts of the gains are treated separately. Since the derivation is equal for both scenarios, the case of real-valued gains is presented here. The entries of the FIM for the model in (4.15) are given as [28]

$$[\boldsymbol{F}(\boldsymbol{\theta})]_{kl} = \operatorname{Tr}\left[\tilde{\boldsymbol{\Sigma}}^{-1}\frac{\partial\tilde{\boldsymbol{\Sigma}}}{\partial\theta_k}\tilde{\boldsymbol{\Sigma}}^{-1}\frac{\partial\tilde{\boldsymbol{\Sigma}}}{\partial\theta_l}\right] + 2\Re\left\{\frac{\partial\boldsymbol{\mu}^H(\boldsymbol{\theta})}{\partial\theta_k}\tilde{\boldsymbol{\Sigma}}^{-1}\frac{\partial\boldsymbol{\mu}(\boldsymbol{\theta})}{\partial\theta_l}\right\}, \tag{4.40}$$

where, resulting from the likelihood function in (4.15), the mean value is given by

$$\boldsymbol{\mu}(\boldsymbol{\theta}) = \boldsymbol{\Phi}_P(\boldsymbol{\Delta\tau})\boldsymbol{\beta} \tag{4.41}$$

and the covariance matrix $\tilde{\boldsymbol{\Sigma}}$ equals that in (4.16).

This means that, to be able to calculate the CRLB for one parameter (e.g., $\Delta\tau_1^{(1)}$), all entries of the FIM need to be computed for the inversion. Reorganizing the parameter vector elements to have a clearer view, the FIM of the parameter vector $\boldsymbol{\theta} = [\Delta\tau^{(1)}, \beta^{(1)}, \Delta\tau^{(2)}, \beta^{(2)}, ..., ..., \Delta\tau^{(P)}, \beta^{(P)}]$ has the form

$$\boldsymbol{F} = \begin{bmatrix} F_{\Delta\tau^{(1)}\Delta\tau^{(1)}} & F_{\Delta\tau^{(1)}\beta^{(1)}} & \cdots & \cdots & F_{\Delta\tau^{(1)}\Delta\tau^{(P)}} & F_{\Delta\tau^{(1)}\beta^{(P)}} \\ F_{\beta^{(1)}\Delta\tau^{(1)}} & F_{\beta^{(1)}\beta^{(1)}} & \cdots & \cdots & F_{\beta^{(1)}\Delta\tau^{(P)}} & F_{\beta^{(1)}\beta^{(P)}} \\ F_{\Delta\tau^{(2)}\Delta\tau^{(1)}} & F_{\Delta\tau^{(2)}\beta^{(1)}} & \ddots & & F_{\Delta\tau^{(2)}\Delta\tau^{(P)}} & F_{\Delta\tau^{(2)}\beta^{(P)}} \\ F_{\beta^{(2)}\Delta\tau^{(1)}} & F_{\beta^{(2)}\beta^{(1)}} & & \ddots & F_{\beta^{(2)}\Delta\tau^{(P)}} & F_{\beta^{(2)}\beta^{(P)}} \\ \vdots & \vdots & & & \vdots & \vdots \\ F_{\Delta\tau^{(P)}\Delta\tau^{(1)}} & F_{\Delta\tau^{(P)}\beta^{(1)}} & ... & ... & F_{\Delta\tau^{(P)}\Delta\tau^{(P)}} & F_{\Delta\tau^{(P)}\beta^{(P)}} \\ F_{\beta^{(P)}\Delta\tau^{(1)}} & F_{\beta^{(P)}\beta^{(1)}} & ... & ... & F_{\beta^{(P)}\Delta\tau^{(P)}} & F_{\beta^{(P)}\beta^{(P)}} \end{bmatrix}. \tag{4.42}$$

Defining $\Omega_k = \frac{2\pi k}{K}$, the entries of the first summand of (4.40) are calculated as

$$\tilde{\boldsymbol{\Sigma}}^{-1} = \begin{bmatrix} \frac{1}{\sigma^2_{\tilde{W}}[0]} & & & \\ & \frac{1}{\sigma^2_{\tilde{W}}[1]} & & \\ & & \ddots & \\ & & & \frac{1}{\sigma^2_{\tilde{W}}[K-1]} \end{bmatrix} \tag{4.43}$$

and

$$\frac{\partial \tilde{\mathbf{\Sigma}}}{\partial \theta_i} = \begin{bmatrix} \frac{\partial \sigma^2_{\tilde{W}}[0]}{\partial \theta_i} & & & \\ & \frac{\partial \sigma^2_{\tilde{W}}[1]}{\partial \theta_i} & & \\ & & \ddots & \\ & & & \frac{\partial \sigma^2_{\tilde{W}}[K-1]}{\partial \theta_i} \end{bmatrix}, \tag{4.44}$$

with

$$\frac{\partial \sigma^2_{\tilde{W}}[k]}{\partial \Delta\tau^{(i)}} = -2\sigma_1^2 \Omega_k \Im\left\{ \sum_{p=1}^{P} \beta^{(i)} \beta^{(p)} \mathrm{e}^{-\mathrm{j}\Omega_k \left(\Delta\tau^{(p)} - \Delta\tau^{(i)}\right)} \right\} \tag{4.45}$$

$$\frac{\partial \sigma^2_{\tilde{W}}[k]}{\partial \beta^{(i)}} = 2\sigma_1^2 \Re\left\{ \sum_{p=1}^{P} \beta^{(p)} \mathrm{e}^{-\mathrm{j}\Omega_k \left(\Delta\tau^{(p)} - \Delta\tau^{(i)}\right)} \right\}. \tag{4.46}$$

The entries of the second summand of (4.40) are

$$\frac{\partial \boldsymbol{\mu}}{\partial \theta_i} = \left[\frac{\partial \mu[0]}{\partial \theta_i}, \frac{\partial \mu[1]}{\partial \theta_i}, \ldots, \frac{\partial \mu[K-1]}{\partial \theta_i} \right]^T \tag{4.47}$$

$$\frac{\partial \mu[k]}{\partial \Delta\tau^{(i)}} = -j\Omega_k \beta^{(i)} S[k] \mathrm{e}^{-\mathrm{j}\Omega_k \Delta\tau^{(i)}} \tag{4.48}$$

$$\frac{\partial \mu[k]}{\partial \Delta\beta^{(i)}} = S[k] \mathrm{e}^{-\mathrm{j}\Omega_k \Delta\tau^{(i)}}. \tag{4.49}$$

The resulting entries for the FIM can be found in Appendix C.

Although the expressions for the single bounds, i.e., $\left[\boldsymbol{F}^{-1}\right]_{ii}$, are not tractable, three conclusions can be drawn directly from the above equations.

- The bound does not depend on the absolute delays, but solely on the delay differences. The expressions for the entries are a function of the delay differences. Therefore, the resulting bounds for the parameter estimation are also a function of the delay differences.
- Compared to the case of an active system that was presented in [58] and [59], the bounds for the passive system model (i.e., with an unknown transmitted signal) introduce additional dependencies between the parameters. The significant difference is the first summand in (4.40), being a result of the parameter dependent covariance matrix. In the second summand, the difference lies in the frequency dependent covariance matrix.

- Despite the differences between the active and the passive system model, one conclusion holds for both scenarios. The bounds on the first k parameters of $k+m$ parameters are not less than the bounds when there are only k parameters. This was proven thoroughly in [58] using the positive semi-definite property of the FIM. This statement implicates that adding the Pth multipath component, the bound on the last $P-1$ multipath parameters are at least as high as the bounds with the additional component. Hence, a larger number of paths reduces the estimation accuracy for all path components.

In the next section, the estimation bound is evaluated numerically and compared to the estimated parameters using the presented algorithms.

4.1.6 Simulation Results and Analysis

So far in 4.1, two ML-based algorithms aiming at estimating the time delays in a multipath scenario were presented. The bound on the minimum variance of an unbiased estimator was derived for the passive system model. To evaluate the previously presented algorithms, we focus on answering the following questions:

1. How do the two presented algorithms perform compared to correlation-based algorithms that require much less computations?

2. How do both algorithms perform compared to each other and compared to the theoretical bound depending on the following parameters:

    - SNR

    - Delay differences

    - The number of path components

    - Path gains

3. How would the algorithms perform when implemented for the described localization scenario?

Simulation Setup

For the simulation, 50 symbols are generated randomly with a symbol duration of T_s. A multipath scenario with P multipath components is defined by choosing fixed values for $\beta^{(p)}$ and $\Delta\tau^{(p)}$ ($p = 1, 2, ..., P$). The signal energy of the one-path reference signal as well as the multipath signal are normalized to 1. The normalization makes sure that the results differentiate both the error caused by random noise as well as the one resulting from the later arriving replica. Without the normalization, increasing a path gain would additionally increase the signal power, hindering a distinct analysis of the influence of the gain.

All results concern the accuracy of the first path delay estimate only $\left(\Delta\tau^{(1)}\right)$, since this is the desired parameter for localization. The variables that play the biggest role in determining the output accuracy are: (i) the delay difference between the first and later arriving paths, (ii) the strength of the first path $\beta^{(1)}$, (iii) the strength of the next arrived paths, (iv) the total number of arrived paths (v) as well as the SNR.

The following algorithms are compared, labeling them by the names in the brackets:

- The importance sampling algorithm (IS) with the best case concerning the choice of the parameter values ρ_0 and ρ_1. The simulations have shown a robust behavior of the algorithm for ρ_0, so the value was chosen for all simulations to be $\rho_0 = 7$. As to the choice of ρ_1, different simulations have shown an overall best performance for $\rho_1 = 10$. The value of ρ_1 influences the estimation more than that of ρ_0. With ρ_1, the correlation is amplified around obvious peaks and is attenuated elsewhere. For low SNR values, increasing ρ_1 helps generate an importance function that peaks solely around the true delay values and that is almost zero elsewhere. On the other hand, choosing a high ρ_1 value bears the risk of attenuating weaker signal replicas. Simulation results as a function of the parameters ρ_0 and ρ_1 are demonstrated in Appendix E.

- The WRELAX algorithm (WRELAX) with a convergence value of $\epsilon = 0.001$.

- The cross-correlation and peak interpolation as described in Section 3.1 (Strongest Peak).

- The cross-correlation followed by a first peak detection (First Peak). Since the correlation can result in multiple peaks, a threshold γ is defined and the first peak above that threshold is detected and interpolated. Similar to the IS algorithm, results using the best γ value are shown. If no peak exists above the threshold, then the strongest peak is chosen.

Performance over SNR:

First, it is investigated how the algorithms behave in terms of RMSE over increasing SNR. The error is expected to decrease with increasing SNR. However, the larger error source here is the multipath propagation. Precisely, the overlapping of the multipath components, causing the correlation peak shift. Therefore, one can expect different behaviors for differently spaced signal replica. Figure 4.5 shows the RMSE of the first path delay estimate over increasing SNR. The scenario consists of three equally-gained multipath components. The delays are chosen as $\boldsymbol{\Delta\tau} = [2, 2.5, 9]T_s$, aiming at investigating the behavior of the algorithms for closely spaced and largely overlapping paths.

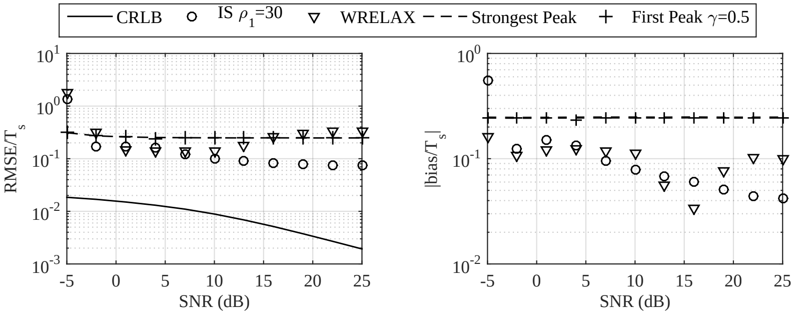


Figure 4.5: Performance of first path estimation algorithms over increasing SNR at overlapping multipath

All algorithms perform considerably worse than the CRLB, especially at higher SNR values. The improvement in performance becomes negligible after a threshold around 5 dB. The algorithms seem to assume that the peak around $\frac{\Delta\tau_2+\Delta\tau_1}{2}$ belongs to one path component. This can be further proven by observing the bias behavior. For SNR values higher than 5 dB, all algorithms continue to show a bias and, hence, can not be compared to the CRLB that assumes unbiased estimates. Taking a look at each algorithm individually, the following facts can be concluded.

- The IS algorithm performs best for closely spaced paths. For higher SNR values, the bias converges to 0.07 T_s. Since the generated probability distribution function peaks at 2.25 T_s and since therefore, most sampled delays are around that value, the bias can be reduced but not eliminated. Looking at the histogram for an SNR of 25 dB in Figure 4.6, the estimation error is within ± 0.4, having a positive bias.
- The WRELAX algorithm shows the highest RMSE for higher SNR. This is because the algorithm tends to estimate outliers for higher SNR while assuming that the peak around 2.25 T_s belongs to one component. At lower SNR values, other local maxima that are closer to the true delay value are found, while for higher SNR, two errors are dominant, the error around 0.2 and -0.5, explaining the increase in RMSE. Depending on the cross-correlation and the resulting side lobes, one of the side lobes can be mistaken for a signal replica, resulting in higher error occurrences at specific values.
- The two correlation-based algorithms perform close to the WRELAX algorithm and are equally good. Searching for the first peak ends up in the same correlation peak as searching for the strongest peak for a threshold of 0.6. This is due to the overlapping replica that merge the two components to one correlation peak. Both correlation-based algorithms show an expected bias of 0.25 that can be further observed in the histogram. The error has a Gaussian-like distribution around the bias.

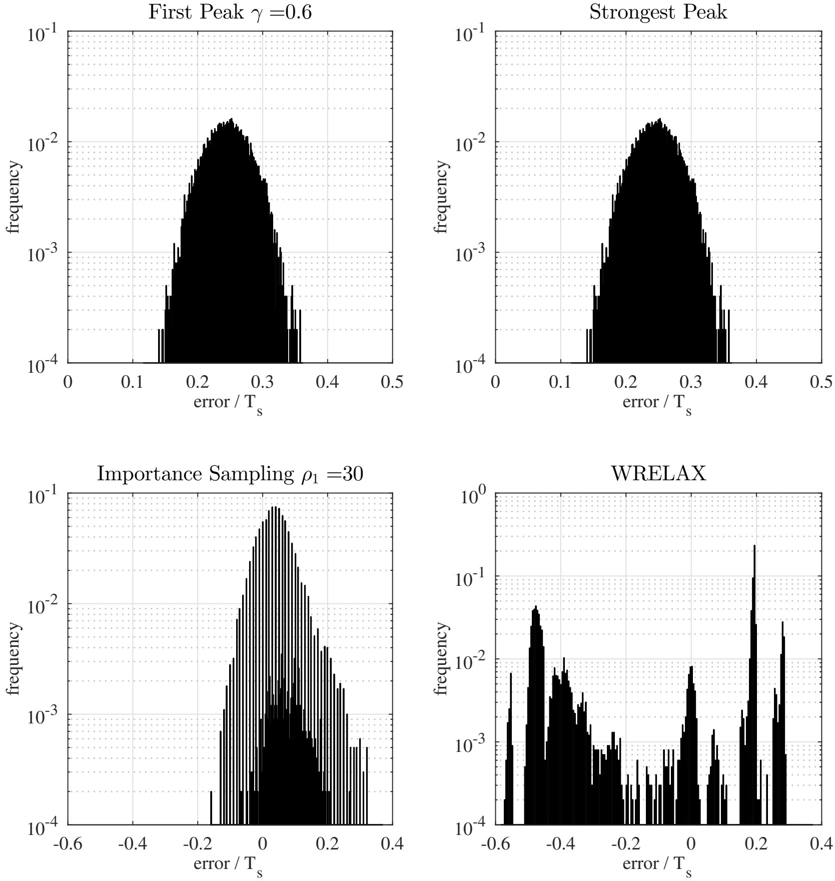


Figure 4.6: Histograms of the estimation error at SNR = 25 dB

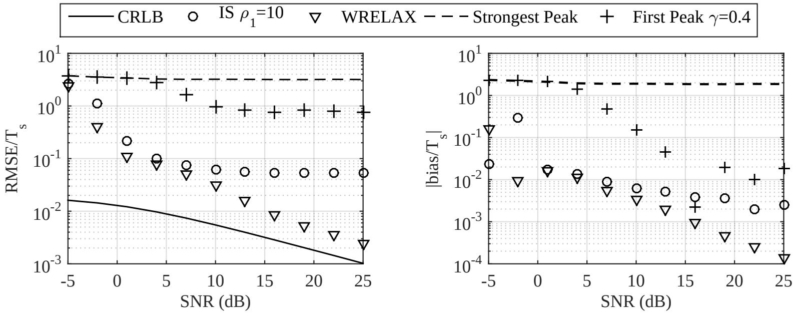


Figure 4.7: Performance of first path estimation algorithms over increasing SNR at mild multipath conditions

Now looking at the performance for well separable paths in Figure 4.7, the behavior is totally different. Here, the delays are chosen as $\boldsymbol{\Delta\tau} = [2, 6, 9]T_s$. The WRELAX algorithm shows the overall best performance and the Strongest Peak method shows the worst one. Again, the algorithms are analyzed separately:

- Considering that interpolation errors are still relevant in the WRELAX algorithm, its result approaches the CRLB for high SNR values. Recall how the algorithm uses an approximated likelihood solution by replacing the transmit signal with the reference signal. The histogram in Figure 4.8 shows how the estimate is largely concentrated around the true value with a small variance.

- The IS algorithm is the second best one. Compared to the WRELAX algorithm, its performance improvement is marginal for higher SNR values. Looking at the histogram, one can see how the estimates are concentrated around the true value, but with a higher variance. This is due to the nature of the algorithm. While the WRELAX algorithm searches directly for the maximum of the cost-function (i.e., the correlation), the IS algorithm uses the importance function to generate samples of the cost function. To enhance accuracy, the algorithm would have to generate more samples from the importance function, resulting in higher computational time.

- The First Peak method outperforms the Strongest Peak Method. The histograms show how the Strongest Peak algorithm shows estimation errors around $[0, 4, 7]T_s$ which corresponds to the delay differences in the simulation scenario. Since all replicas are equally gained, the strongest peak can be random for each noise realization since, due to noise, one of the three peaks can become higher than the other two. Searching for the First Peak with a threshold of $\gamma = 0.4$, most of the estimates lie around the true delay, but this method comes with the risk of outliers, resulting from mistakenly identifying side lobes as signal replica, like for example the values -7 and -2.

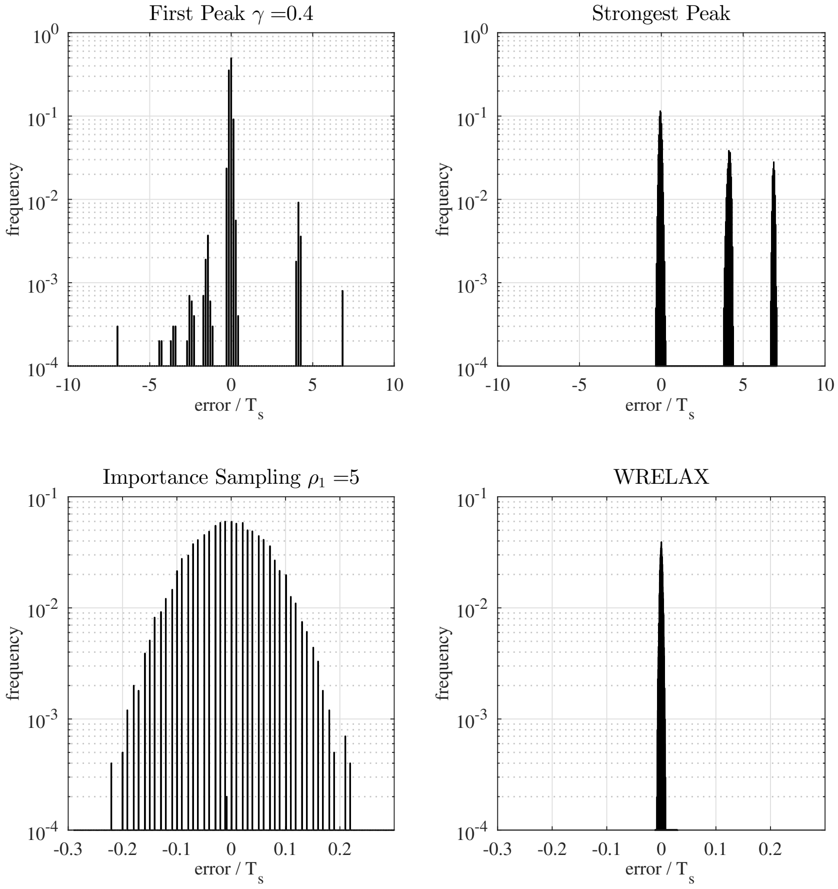


Figure 4.8: Histograms of the estimation error of Figure 4.7 at SNR = 7 dB

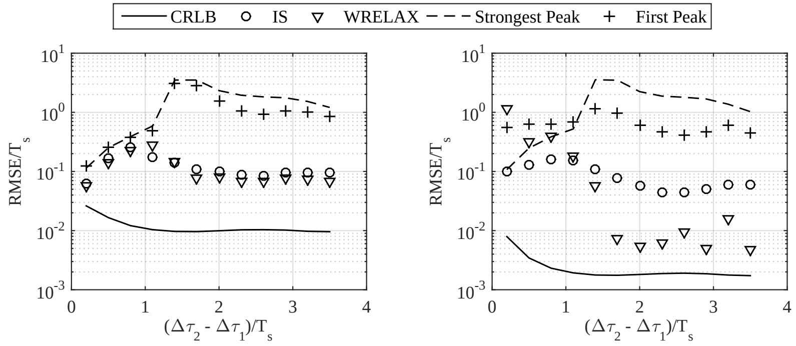


Figure 4.9: Performance of first path estimation algorithms over increasing delay difference. Left: SNR = 5 dB , right: SNR = 20 dB

Performance over Delay Differences:

Starting from the detailed observations for increasing SNR at different multipath scenarios, the behavior over growing $\Delta\tau^{(2)} - \Delta\tau^{(1)}$ in Figure 4.9 can now be explained.

The results reaffirm that the algorithms can not resolve closely spaced signal replicas, while they succeed in doing so for higher delay differences, especially the WRELAX algorithm. For sufficiently high SNR values, the algorithm succeeds in separating the multipath components and estimating their parameters starting at a delay difference of $1.5T_s$. If there is no information about how close the signal replicas will be, then the WRELAX algorithm is expected to obtain the overall best estimates, whereas for closely spaced replicas, all algorithms perform considerably worse than the attainable bound. In that case, correlation-based algorithms are preferable due to the reduced computational complexity.

Unexpectedly, the CRLB as well as the WRELAX algorithm partly increase for a larger delay difference. This is because of the specific signal form. Consider the main lobe of a Si-function while adding a second delayed version of the same function and increasing the delay. The outcome is periodic since the function has such behavior. Hence, although the overall influence of the second added function decreases when the delay is increased, throughout one period, the influence decreases before it increases again. Thus, a larger delay

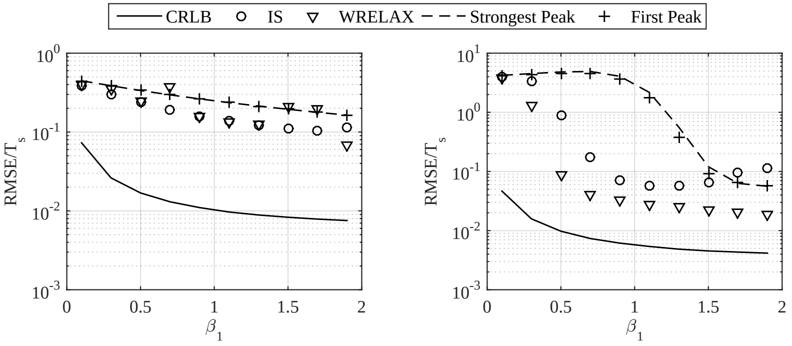


Figure 4.10: Performance of first path estimation algorithms over increasing path gain. Left: $\mathbf{\Delta\tau} = [2, 2.5, 9]T_s$, right: $\mathbf{\Delta\tau} = [2, 6, 9]T_s$

difference is, in general, beneficial for the estimation of the main lobe (i.e., the first delay), but the opposite effect can be observed when looking at a small scale increase in the delay difference.

Performance over Path Gain:

The influence of an increasing relative gain of the first path can be seen in Figure 4.10. On the left side, the scenario with closely spaced paths from the first result is depicted and the gain of the first path is increased, while the other gains remain equal to 1. For closely spaced paths, the increase in accuracy for higher $\beta^{(1)}$ and lower $\beta^{(2)}$ is expected. It reduces the bias resulting from the overlapping of the multipath components, but, as can be seen, still failing to completely resolve the single replicas. For well separable paths, the results on the right side show how the algorithms are able to distinct the first component and estimate its delay starting a gain of 0.5.

Increasing Path Components:

To verify whether the calculated CRLB behaves as expected for an increasing number of signal replicas, and to compare its behavior to that of the algorithms, a scenario consisting of 7 signal replicas with delays $\mathbf{\Delta\tau} = [2, 3, 5, 5.5, 8, 13, 16]T_s$ and gains $\boldsymbol{\beta} = [1, 1, 1, 1, 1, 1, 1]$ was simulated. The number of used

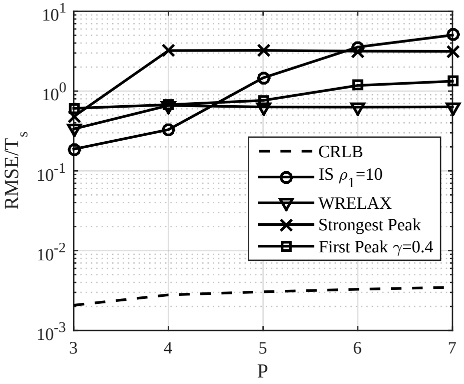


Figure 4.11: Performance over growing number of received signal replica

signal replicas is incremented from 3 to 7. The estimation accuracy of each delay is expected to decrease with a growing number of unknown parameters. Figure 4.11 shows the result of the algorithms compared to the CRLB for an increasing number of multipath components. It can be seen how the IS algorithm has a problem with a growing number of multipath components in general. The algorithm tends to estimate side lobes, leading to large errors. The WRELAX algorithm is not influenced by the additional 5th to 7th multipath component. The main error reason here is the bias resulting from the superposition of the first two paths. The CRLB behaves as expected and is much lower than the estimation results.

Complete Estimation using two Cost-Channels over SNR:

After analyzing the performance of the algorithms for different scenarios, the question whether it would be worth it to implement these algorithms for the smart TDoA system and which errors to expect need to be answered to be able to plan the further processing chain. For exemplary purposes, three COST 207 channel models that were standardized in [60] to enable simulating scenarios in urban channels are implemented. COST 207 offers different channel environments, making a comparison between various scenarios possible. From the available scenarios, a channel model for a rural areal (RA), one for a typical urban area (TU) and one for a bad urban (BU) area are chosen. These three resemble mild to worst case propagation conditions for the algorithms. The processing chain was extended to include the MDL algorithm, since in real scenarios, the number of multipath components is unknown. Table 4.1 shows the channel characteristics of the three chosen channels.

	P	Delays (μs)	Average Path Gains (dB)
RA	4	$[0, 0.2, 0.4, 0.6]$	$[0, -2, -10, -20]$
TU	6	$[0, 0.2, 0.6, 1.6, 2.4, 5]$	$[-3, 0, -2, -6, -8, -10]$
BU	12	$[0, 0.2, 0.4, 0.8, 1.6, 2.2,$ $3.2, 5, 6, 7.2, 8.2, 10]$	$[-7, -3, -1, 0, -2, -6,$ $-7, -1, -2, -7, -10, -15]$

Table 4.1: Characteristics of the chosen COST channels

While the rural area channel model has only 4 replica with the first one being the strongest one, the bad urban model has 12 multipath components and its strongest component is the 4th one.

The simulated transmit signal consists of random data sampled at 1 MHz, aiming at analyzing a narrow-band scenario. Figure 4.12 shows the performance of the algorithms as well as the CRLB. The results show the performance for a known number of multipath components as well as for the estimated number using the MDL algorithm.

Overall, the results show the best performance for the WRELAX algorithm, except for the rural area channel model, where the correlation-based algorithms perform best. This degradation is due to large outliers in that scenario. The IS algorithm does not show any advantage over the other algorithms. Additionally, the degradation due to the unknown number of multipath components is marginal.

Hence, assuming an urban environment, the TDoA system is faced with intermediate TDoA estimates that are biased and that have high variances. Depending on the channel density and the signal bandwidth, the biases can be in the magnitude of $T_s/2$ which, for a sample rate of 1 MHz, corresponds to a bias of 150 meters, making the time delay estimate highly inaccurate as an input to the localization algorithms. Lower signal bandwidths would further reduce the accuracy since their correlation peaks would be wider, resulting in larger overlapping areas, shifting the correlation peak even further.

As to the choice of the algorithm, on the one hand, most scenarios have shown a higher accuracy for the ML-based algorithms, especially for the WRELAX algorithm for high SNR scenarios with distance differences larger than 1.5 T_s.

Still, some large outliers degrade the algorithm performance greatly. Some of these outliers can be detected by adding intelligence to the algorithm, e.g., by eliminating too weak estimated replicas or by detecting an error if one of the detected replicas is too far away from the remaining ones.

Hence, for high SNR scenarios and large delay differences (or high signal bandwidths), the WRELAX offers promising estimation accuracies. For all other scenarios, the trade-off between the marginal improvement in accuracy and the enormous increase in computational complexity makes the WRELAX rather impractical.

On another note, the inability to model the estimation error in multipath scenarios is another challenge for the positioning system. Estimation errors using the correlation-based algorithms have shown Gaussian-like behaviors around one or more peaks (see Appendix D). However, using the ML-based algorithms, the error behavior is not predictable, since it highly depends on the channel parameters, thus, making the second step, the localization, even harder. All simulation results have shown that two largely overlapping signal replicas will continue to be estimated as one path, resulting in the same bias. This bias is the dominant source of error, making the estimates almost independent of the SNR.

In the next section, algorithms that deal with the TDoA estimation errors in multipath environments are presented. Specifically, the biased estimates are modeled and different approaches dealing with these biases are shown, aiming at the best possible passive localization in harsh multipath scenarios, while taking into account that the estimation error can not be easily modeled in those environments.

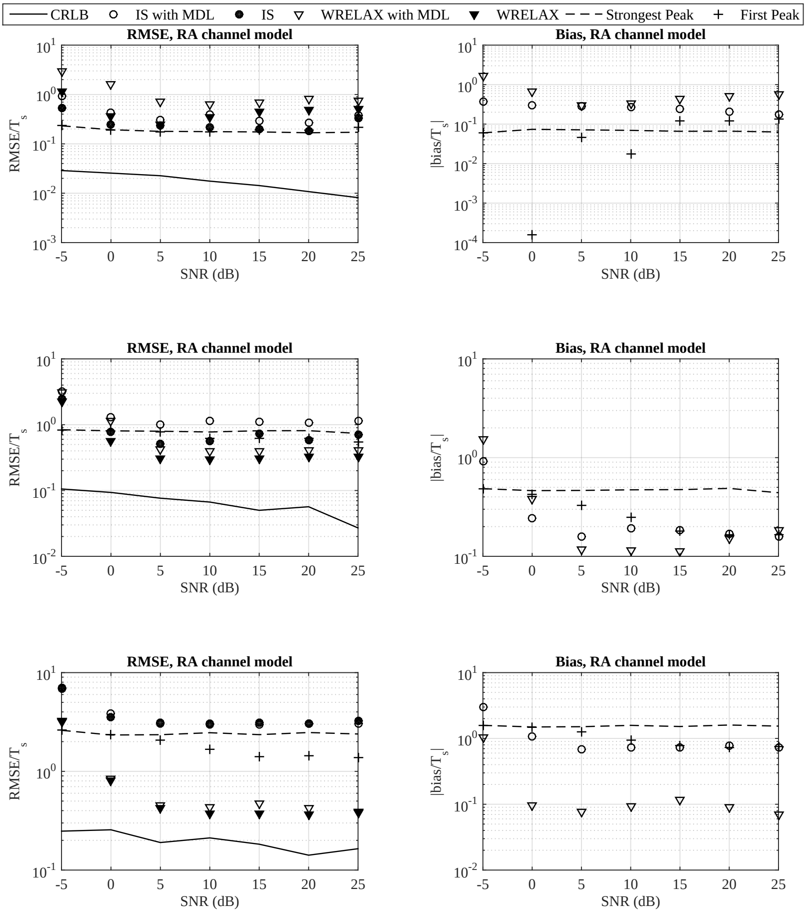


Figure 4.12: Algorithm performance using COST 207 channel models

4.2 Position Estimation in NLoS

Previously in this chapter, the difficulty of resolving multipath components in time was shown, especially when the correlation time is in the magnitude of the propagation time of the traveled distances. Particularly, signals having weak LoS components yield biased TDoA estimates. In urban scenarios, the direct path often gets attenuated by buildings or other objects standing in between the transmitter and the receiver. Thus, a later arriving reflected path can be stronger than the first attenuated one. This scenario can not be ignored when designing a localization system.

Except for Subsection 4.2.4, this section assumes that the time delay estimation yielded one or more biased TDoAs from the available sensors due to the absence of a strong direct peak or due to unresolvable multipath peaks. Both scenarios will be denoted NLoS scenarios, primarily indicating a bias in the estimated TDoA value.

Throughout the years, various approaches have emerged aiming at mitigating the NLoS effect. After describing the system model for the NLoS scenario, three algorithms based on this model are presented. In subsection 4.2.4, a different approach to dealing with tough multipath scenarios is described. Various simulated scenarios show and compare the performances of the algorithms at the end of this section.

4.2.1 System Model and the Effect of NLoS

The scenario of an unsuccessful time delay estimation due to unresolvable multipath components or due to a too weak direct path can generally be approximated by the following model

$$\widehat{\Delta\tau}_{i,j} = \Delta\tau_{i,j}^{(1,1)} + e_{i,j} + \eta_{i,j}, \tag{4.50}$$

where $\Delta\tau_{i,j}^{(1,1)}$ is the correct TDoA, $e_{\tau_{i,j}}$ is the bias resulting multipath propagation and $\eta_{i,j}$ is the noise term. In the last section, the histograms of the estimation error in different scenarios were shown. This model can be a good approximation for correlation-based algorithms, whenever the estimate is around one peak and not multiple peaks. For the ML-based algorithms, this model can not be assumed for most scenarios. However, although the model does not match the results of the last section, it is widely used in the literature and can be implemented easily when simulating the localization algorithms.

The following presented algorithms do not require this to be the estimation error model, they merely use it to describe the problem. Thus, while it is an approximation of the true TDoA estimate, the following approaches are independent of it.

While this model is widely used, modeling the bias $e_{i,j}$ deviates:

- $e_{i,j}$ is fix: This means that at each point in space $e_{i,j}$ takes a specific value and that this value is constant for at least a time period. This model was applied in [61] by using sequential observations to try to learn which sensors have NLoS. On the other hand, this can be used to try and learn the biases, as was shown in [62] and will be presented in Section 4.2.3.

- $e_{i,j}$ is random and follows known distributions [63], e.g., uniform, exponential or Gaussian. This information can be either used to help finding the NLoS induced measurements or, even better, it can be used to enhance the overall position estimation accuracy by applying the Maximum a Posteriori (MAP) criterion.

- Experimentally extracted characteristics of $e_{i,j}$: Some research studies have conducted extensive measurement campaigns to extract the behavior of the measurement system under NLoS and have used this information to identify and mitigate the NLoS error [64].

From the time model in (4.50), the range distance model is derived since, like in section 3.2, this is the input value of the following position estimation algorithms. The biased range difference estimates are modeled as

$$\widehat{\Delta d}_{i,j} = \Delta d_{i,j} + b_{i,j} + \xi_{i,j}, \quad \xi_{i,j} \sim \mathcal{CN}(0, \sigma^2_{\Delta d_{i,j}}). \tag{4.51}$$

The influence of NLoS on the position estimation: To be able to comprehend the effect of NLoS, Figure 4.13 shows how the bias affects the hyperbola constellation, with and without the noise term. Although noise also leads to a hyperbola shift, the shift caused by bias can be assumed to be more severe. The underlying statement behind that assumption is: whereas a noise term can be arbitrarily reduced by averaging, the bias term will remain the same and would thereby lead to an obvious, constant shift of the hyperbola in a certain direction. Based on this realization, possible approaches to mitigate the NLoS error are presented next.

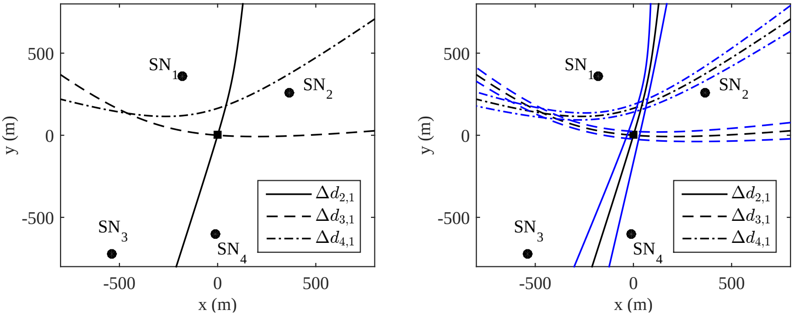


Figure 4.13: The bias effect on the hyperbola intersections with TX at the origin; left: without noise; right: with noise

4.2.2 NLoS Identification and Mitigation

A survey with a thorough discussion about possible NLoS mitigation approaches and techniques was provided in [65]. Here, the most important categories are listed:

- ML-based methods: For unknown bias distributions, the ML solution consists of identifying the NLoS induced measurements and discarding them from the position estimation. Examples can be found in [61] and [66]. A known bias distribution can further enhance the estimation by applying the MAP criterion. In that case, the NLoS measurements do not need to be eliminated as was shown in [67].

- Since the optimal ML solution consisting of an optimal identification of NLoS sensors and a localization using only LoS sensors is practically very challenging, another family of methods emerged as a suboptimal solution, the so called constrained techniques [68], [69]. For harsh NLoS scenarios, these methods oftentimes outperform ML-based solutions.

Here, two popular methods were chosen that assume no *a priori* information about the bias distribution and that have shown to perform well for a large variety of scenarios (i.e., simple and severe NLoS).

An alternative is to rely on the premise that the NLoS induced measurements have a distinctively higher variance than those with LoS [70], since the signals experience reflection and scattering effects. A measurement campaign conducted by Nokia research in [71] yielded standard deviation values for ξ in (4.51) of 150 m for LoS and of 409 m for NLoS. Relying on this assumption, the NLoS effect can be alleviated simply by using the covariance matrix to weight the algorithms, something that is implemented already for the LS algorithms as well as the Kalman filters in Section 3.2. However, by weighting the measurements according to their covariances, the bias can only be reduced and not removed completely from the estimate.

The Residual Weighting Algorithm

Looking at Figure 4.13, one of the sensors, SN_4 has no line-of-sight to the transmitter. A bias means that, on average, the estimated RD of SN_4 using SN_1 as reference, yields $\Delta d_{4,1} + b_{4,1}$. The figure shows how the bias affects the hyperbola constellation, especially considering that the errors caused by measurement noise can be kept small by averaging multiple estimated TDoAs. Hence, noise typically makes the unbiased hyperbolas intersect in a small region whereas the biased hyperbola lies far outside that region. Thus, estimating a position using biased measurements and applying the LS approach, the position estimate can be expected to lie far from the hypothetical TDoAs. This is the basic idea of the method after Chen that was presented in [72].

Chen uses this information by forming so called subsets, which are smaller segments from the set of available sensors with a minimum of three sensors (for a two-dimensional localization). For example, five sensors can be segmented into 16 subsets, where:

- $\binom{5}{3} = 10$ subsets use three sensors
- $\binom{5}{4} = 5$ subsets use four sensors
- $\binom{5}{5} = 1$ subset uses all five sensors.

Each of the possible subsets can be used to yield a position estimate, e.g., by running the TSE algorithm presented in 3.2. Assuming that at least three sensors have LoS, it is expected that their estimated position is *aligned* with their estimated TDoAs, i.e., their residuum is small compared to that using NLoS measurements. Thus, Chen weights the outcome of each subset by the inverse of its residuum. Originally developed for time of arrival measurements, the algorithm steps for TDoA are [73]:

1. Estimate a position for each subset $S_q \in \mathcal{Q}_3^N$ using a basic TDoA algorithm (e.g., TSE). S_q denotes the qth subset and $\mathcal{Q}_3^N$ denotes all possible sensor combinations with a minimum of three sensors and a maximum of N sensors. The estimate of subset S_q is $[\hat{x}_q, \hat{y}_q]^T$.

2. Calculate the residual R_q for each subset S_q following

$$R_q = \frac{\sum\limits_{i \in S_q, i \neq j} (\widehat{\Delta d}_{i,j} - \widetilde{\Delta d}_{i,j})^2}{|S_q| - 1}, \tag{4.52}$$

where

$$\widetilde{\Delta d}_{i,j} = \sqrt{(x_i - \hat{x}_q)^2 + (y_i - \hat{y}_q)^2} - \sqrt{(x_j - \hat{x}_q)^2 + (y_j - \hat{y}_q)^2}. \tag{4.53}$$

$\widehat{\Delta d}_{i,j}$ is the estimated RD, resulting from applying delay estimation algorithms to the received signals $r_i(t), r_j(t)$. On the other hand, $\widetilde{\Delta d}_{i,j}$ is the TDoA corresponding to the estimated position, yielded by using $\widehat{\Delta d}_{i,j}$ as well as at least one other estimated TDoA.

To clarify, consider the scenario depicted in Figure 4.14. On the left side, an ideal channel is assumed, resulting in error-free estimated TDoAs that yield an error-free estimated transmitter position $[\hat{x}_T, \hat{y}_T]^T$. The estimated point has zero distance to the hyperbolas, i.e., the residuum is zero. On the right side, the hyperbola shift caused by errors in the estimated TDoAs leads to three intersection points and the estimated position lies between them, having the least squares error to all intersection points. Hence, the hypothetical TDoAs resulting from the estimated point are not perfectly aligned with the truly estimated ones, yielding a residuum.

$|S_q|$ denotes the magnitude of the qth set, i.e., the number of sensors in the qth subset. Since different subsets have different reference sensors, the subscript $\widehat{\Delta d}_{i,j}$ denotes the delay difference of sensor i using sensor j as the reference sensor.

3. Form the weighted sum of the estimated transmitter positions as

$$[x_{\text{sum}}, y_{\text{sum}}]^T = \sum_{q \in \mathcal{Q}_3^N} \frac{1}{R_q} [\hat{x}_q, \hat{y}_q]^T. \tag{4.54}$$

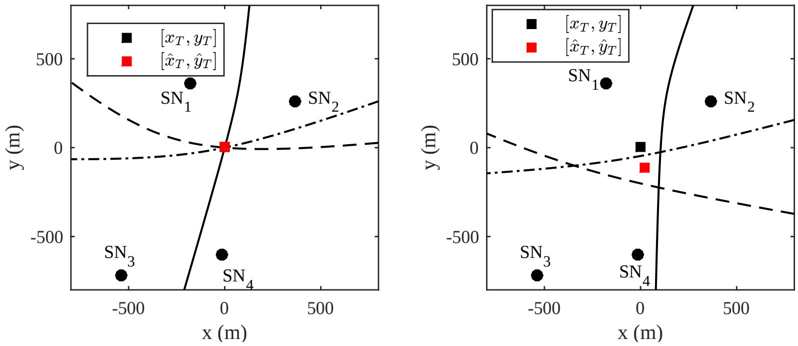


Figure 4.14: LS estimated positions using left: error-free TDoAs and right: noisy TDoAs

4. The final estimate is the weighted sum divided by the sum of weights

$$[\hat{x}_T, \hat{y}_T] = \frac{1}{\sum\limits_{q \in \mathcal{Q}_3^N} \frac{1}{R_q}} [x_{\text{sum}}, y_{\text{sum}}]^T. \tag{4.55}$$

Since Chen's method exploits the residuum to identify *good* TDoA estimates and *bad* ones, one thing needs to be considered when choosing the minimum number of sensors needed for a subset, following from Figure 4.14. The residual of one subset is defined as the sum of the square distances between the hyperbolas and the estimated position. A subset consisting of three sensors results in two hyperbolas, meeting in at least one point. A least-squares based algorithm would estimate the intersection point, resulting in zero residuum. Hence, for two hyperbolas (i.e., three sensors), the residuum is not a measure of the TDoA estimate quality. Thus, the minimum number of sensors in a subset must be set to four to fulfill the requirement of a residuum.

The disadvantage of this algorithm is the fact that the used basic positioning algorithm needs to be run $|Q_3^N|$ times. Another disadvantage is that, while this algorithm will never have a *false alarm*, i.e., it will never eliminate a sensor that has LoS, it will always have a biased outcome since the NLoS error might just be reduced, but not eliminated from the overall estimate. This and more will be discussed in the results section.

Cong Residual Algorithm

As an alternative, another residual based algorithm proposed by Cong in [74] and [75] overcomes two weaknesses of the previously described algorithm. The first weakness being the fact that it does not eliminate NLoS measurements since it is not about identifying them. The second weakness is that it does not consider the measurement noise while computing the residual. A high residual resulting from a measurement with high variance is less an indicator for NLoS than when a high residual results from low variance measurements. Cong solves this by defining a new residual. From (3.20), the basic conditional probability density function in case of LoS (since there is no available information about the bias distribution) is

$$p_{i,j}(\widehat{\Delta d}_{i,j} - f(\boldsymbol{p}_i, \boldsymbol{p}_j, \boldsymbol{x})|\text{LoS}) = \frac{1}{\sqrt{2\pi}\sigma_{\Delta d_{ij}}} \exp\left(\frac{-(\widehat{\Delta d}_{i,j} - f(\boldsymbol{x}))^2}{2\sigma^2_{\Delta d_{ij}}}\right). \tag{4.56}$$

For a measured range difference value $\widehat{\Delta d}_{i,j}$ and a reference location (i.e., an initial estimated position), the corresponding Cumulative Distribution Function (CDF) is a measure of how likely the error caused by noise is smaller than the measured error. Therefore, a higher CDF value means the observed $\widehat{\Delta d}_{i,j}$ is more likely to be a result of NLoS bias. The algorithm steps are as follows:

1. Estimate an initial guess $\boldsymbol{x}_0$ which will be the reference point for the calculation of the residuals. This can be done by utilizing all estimated TDoAs from all sensors and running the TSE algorithm.

2. Calculate the new residual for each sensor as

   $$res_{i,j} = 0.5 + 0.5\,\text{erf}\left(\frac{|\widehat{\Delta d}_{ij} - f(\boldsymbol{p}_i, \boldsymbol{p}_j, \boldsymbol{x}_0)|}{\sqrt{2}\sigma_{\Delta d_{i,j}}}\right) \tag{4.57}$$

   If the bias can be assumed positive (which is the case for correlation-based algorithms assuming a one-path reference sensor), then the residual can incorporate this information by eliminating the absolute operation in calculation of the residuum.

3. If the number of NLoS sensors is known, identify the N_{NLoS} TDoAs with the highest sensor residuals as NLoS.

4. If the number of NLoS sensors is unknown, identify TDoAs with residuals higher than a defined threshold γ as NLoS-caused residuals.

5. If there are enough LoS measurements, use only them to estimate the final position. If not, issue a warning.

The choice of γ as well as its effect on the estimation accuracy will be discussed in the simulation section.

4.2.3 NLoS mitigation using the Baum-Welch Algorithm

Consider a transmitter that is moving slowly in an area. At each different position, the multipath environment and hence, the bias, changes. However, if we stay in one position, the bias can be assumed fixed, at least for a time period (i.e., since the surrounding buildings and objects remain almost the same). This can actually be used to learn something about the bias and to improve the overall accuracy. In this model, a new aspect of the localization is addressed, the movement behavior of the transmitter. The approach was presented in [62]. Consider for example a very small part of the street map of Karlsruhe shown in Figure 4.15.

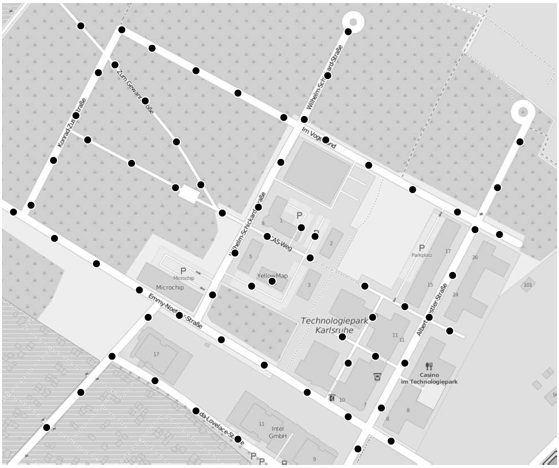


Figure 4.15: Street map of part of Karlsruhe with defined Markov states

Assuming the outdoor movement is limited to the streets, a slowly moving transmitter in the plotted area has a limited choice of changing its position from one time step to the next. Additionally, this transmitter can show a special behavior (e.g., going in circles around an area). In both cases, this information can be used effectively by modeling the movement using a hidden Markov model. A hidden Markov process is a stochastic model describing the behavior of an evolving chain of states, the so called Markov chain [76]. The hidden character results from the fact that the states can not be observed directly, but other parameters that include information about the state can be observed. For the described scenario, the different points on an area of interest, marked exemplary by the circles in Figure 4.15 can be modeled as the states of a Markov model. Groups of points can also be modeled as one state. The parameters of an M-state hidden Markov model are:

- $\boldsymbol{A}$: $M \times M$ transition matrix; $[A]_{ij} = a_{ij}$ is the transition probability from state i to state j. For the example in Figure 4.15, elements of the matrix representing non-adjacent locations are zero. The remaining elements depend on the movement behavior of the transmitter.

- $p_i(\boldsymbol{o}_t)$: conditional output probability, where $\boldsymbol{o}_t$ is the observed quantity and $p_i(\boldsymbol{o}_t)$ is the probability of observing $\boldsymbol{o}_t$ while being in state i. Here, the observed quantity is the estimated position at time step t, thereby, making the Markov model a higher layer model for the positioning system.

- π_i: the initial probability of being in state i.

Estimating the Markov state is in that case equivalent to estimating that the transmitter is in a specific point or within a group of points (if more than one point is to be defined as a state). The observed quantity are estimated positions from the received signals that are, due to multipath and NLoS-propagation, oftentimes biased.

Consider, for example, three states depicted in Figure 4.16. The three states (marked by squares) and their corresponding observations (marked by the equally-colored circles) can be seen. In a zero-mean error scenario, the observations would have the true state as their mean position. However, when the estimated TDoAs have biases, assumed fixed for each state, the mean estimated position is shifted according to those biases. Due to the remaining random noise term from (4.50), the observations have a Gaussian-like behavior and can therefore

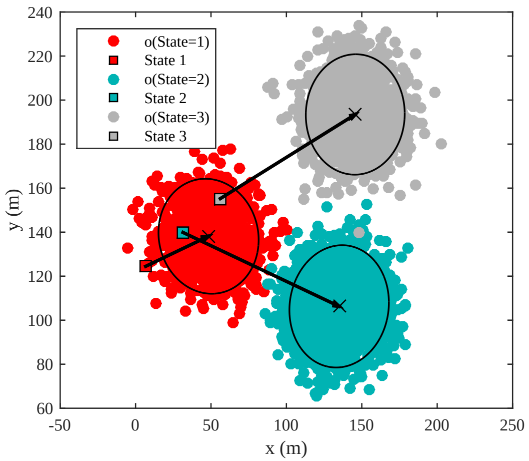


Figure 4.16: An example of an observed Markov chain with three states

be approximated by

$$p_i(\boldsymbol{o}) \approx \frac{1}{2\pi\sqrt{|\boldsymbol{P}_i|}} \mathrm{e}^{-\frac{1}{2}(\boldsymbol{o}-\boldsymbol{\mu}_i)^T \boldsymbol{P}_i^{-1}(\boldsymbol{o}-\boldsymbol{\mu}_i)}, \tag{4.58}$$

where $\boldsymbol{\mu}_i$ and $\boldsymbol{P}_i$ are the mean vector and covariance matrix of the ith state. Interpreting this model, the final position estimation error resulting from biased and/or noisy TDoAs is described by a bias vector and a covariance matrix.

This is an approximation that can be further enhanced by replacing the Gaussian probability density function by a Gaussian mixture, meaning that a position estimate resulting from erroneous TDoAs can be mapped on multiple Gaussian variables with different mean vectors and covariance matrices. For simplicity, the single Gaussian case is handled here but can be easily extended.

Based on the described Markov model, not only can the model parameters be used to greatly enhance the overall accuracy, but they can also be learned using the so called Baum-Welch algorithm that was presented in [77]. The following description of the Baum-Welch algorithm as well as the algorithm steps follow the tutorial by Rabiner in [78].

The Baum-Welch algorithm is an iterative algorithm searching for the maximum likelihood solution parameters of a Hidden Markov Model (HMM) given an observed sequence $\boldsymbol{o} = [\boldsymbol{o}_1, \boldsymbol{o}_2, ..., \boldsymbol{o}_T]$. The model parameters that can be estimated are $\boldsymbol{\theta} = (\boldsymbol{A}, \boldsymbol{p}(\boldsymbol{o}), \boldsymbol{\pi})$ with $\boldsymbol{p}(\boldsymbol{o}) = [p_1(\boldsymbol{o}), p_2(\boldsymbol{o}), ..., p_M(\boldsymbol{o})]^T$ and $\boldsymbol{\pi} = [\pi_1, \pi_2, ..., \pi_M]^T$. The algorithm requires as input: (i) initial values for the model parameters chosen randomly or using some information (e.g., the street map), (ii) the defined states and (iii) an observed sequence $\boldsymbol{o}_1, \boldsymbol{o}_2, ..., \boldsymbol{o}_T$ of T position estimates. The steps of the algorithm are:

1. Compute state probabilities using current model parameters $\boldsymbol{\theta}$: To calculate the state probabilities, the forward and backward probabilities of the sequence $\mathbf{o}_1, \mathbf{o}_2, ..., \mathbf{o}_T$ need to be calculated first. The forward probability $\nu_i(t)$ is the probability of being in state i at time t using the partial observation sequence $\mathbf{o}_1, \mathbf{o}_2, ..., \mathbf{o}_t$. It can be computed inductively using the currently estimated model parameters $\boldsymbol{\theta}$ as follows:

   - Initialization: $\nu_i(1) = \pi_i \cdot p_i(\mathbf{o}_1)$.
   - Induction : $\nu_i(t+1) = \left[\sum_{j=1}^{N} \nu_j(t) a_{ji}\right] p_i(\mathbf{o}_{t+1})$.

   The backward probability $\xi_i(t)$ is the probability of being in state i at time t given the partial observed sequence $\mathbf{o}_{t+1}, \mathbf{o}_{t+2}, ..., \mathbf{o}_T$ and the currently estimated model parameters $\boldsymbol{\theta}$. It can be solved for inductively as follows:

   - Initialization: $\xi_i(T) = 1$.
   - Induction: $\xi_i(t) = \sum_{j=1}^{N} a_{ij} p_j(\mathbf{o}_{t+1}) \xi_j(t+1)$.

   Using the forward and backward probabilities, the state probabilities are calculated as

$$\chi_i(t) = \frac{\nu_i(t)\xi_i(t)}{\sum_{j=1}^{N} \nu_j(t)\xi_j(t)}. \tag{4.59}$$

2. Re-estimate model parameters to $\hat{\boldsymbol{\theta}}$: The model parameters can now be re-estimated to $\hat{\boldsymbol{\theta}} = (\hat{\mathbf{A}}, \hat{\mathbf{p}}(\mathbf{o}), \hat{\boldsymbol{\pi}})$ by:

   - Calculating the probability of being in state i at time t and state j at time $t+1$ as

$$\psi_{ij}(t) = \frac{\nu_i(t) a_{ij} p_j(\mathbf{o}_{t+1}) \xi_j(t+1)}{\sum_{k=1}^{N} \sum_{l=1}^{N} \nu_k(t) a_{kl} p_l(\mathbf{o}_{t+1}) \xi_l(t+1)}. \tag{4.60}$$

- Updating the transition probabilities by

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1} \psi_{ij}(t)}{\sum_{t=1}^{T-1} \chi_i(t)}. \tag{4.61}$$

- Updating the mean vectors and covariance matrices of the Gaussian mixtures by

$$\hat{\boldsymbol{\mu}}_i = \frac{\sum_{t=1}^{T} \chi_i(t) \cdot \mathbf{o}_t}{\sum_{t=1}^{T} \chi_i(t)} \tag{4.62}$$

 and

$$\hat{\boldsymbol{P}}_i = \frac{\sum_{t=1}^{T} \gamma_i(t) \cdot (\mathbf{o}_t - \boldsymbol{\mu}_i)(\mathbf{o}_t - \boldsymbol{\mu}_i)^T}{\sum_{t=1}^{T} \chi_i(t)}. \tag{4.63}$$

3. Repeat steps 1 and 2 until convergence: The state probabilities can be re-calculated using updated model parameters and the model parameters can be re-estimated using the new probabilities. These steps are repeated until the maximization of the likelihood $p(\mathbf{o}|\boldsymbol{\theta}) = \sum_{i=1}^{N} \nu_i(T)$ has reached a desired convergence, i.e., when the improvement of the likelihood is negligible (e.g., < 0.001) from one step to the next.

The described Baum-Welch algorithm aims at estimating the model parameters. To estimate the position at each time step, the forward algorithm is used. The forward algorithm (FW) is the actual *state sequence* estimation algorithm. Unlike the Baum-Welch algorithm, the forward algorithm estimates the state sequence using given HMM parameters (known or learned through the Baum-Welch algorithm) and an observation sequence by calculating the forward probability given in step 1 at each time step. The estimated state at time t is the state with the highest calculated forward probability.

4.2.4 The Weighted Least Squares One-Step Localization

As an alternative to the previous algorithms, a completely different approach to the problem of position estimation in multipath channels is presented here. In Section 2, the one-step localization was mentioned as an alternative positioning approach. So far, the position estimation has been presented as a two-step process, the first step being the time delay estimation from the received signals and the second step being the position estimation using the obtained delays. Alternatively, the position can be yielded directly from the received signals using the so called one-step solution. This approach bears a much higher computational cost compared to the two-step approach in AWGN and offers a slightly higher accuracy in return, making two-step solutions more convenient. In case of harsh multipath scenarios, it is of interest to see whether the one-step solution enhances the estimation accuracy, especially since high resolution time delay algorithms tend to be computationally complex. This work was presented in [79].

First, the least squares solution is presented for the case of AWGN. The received and sampled signal at sensor i is again modeled as

$$r_i[n] = s(nT_s - t_0 - \tau_i) + w_i[n], \quad n = 0, 1, ..., K-1, \tag{4.64}$$

where $i = 1, 2, ..., N$. The propagation delay τ_i is related to the transmitter position $\boldsymbol{x} = [x_T, y_T]^T$ as

$$\tau_i(\boldsymbol{x}) = t_0 + \frac{\sqrt{(x_i - x_T)^2 + (y_i - y_T)^2}}{c} \tag{4.65}$$

with c being the propagation speed. Again, the frequency domain is favored since it separates the signal from the parameters that need to be estimated. After taking the DFT over the received samples (assuming that the delay is an integer in T_s), the least squares solution is given by minimizing the following cost function [80]

$$\begin{aligned}\hat{\boldsymbol{x}}_{\text{LS}} &= \arg\min_{\boldsymbol{x}} Q(\boldsymbol{x}) \\ Q(\boldsymbol{x}) &= \sum_{i=1}^{N}\sum_{k=0}^{K-1}\left|R_i[k] - \alpha_i S[k]\exp\left(\frac{-\mathrm{j}2\pi(t_0 + \tau_i(\boldsymbol{x}))k}{K}\right)\right|^2.\end{aligned} \tag{4.66}$$

Since the transmitted signal and the transmit time are unknown, this solution could result in ambiguity as was shown in [81]. Therefore, one sensor is defined again as reference sensor (here: SN_1). The least squares solution becomes:

$$\bar{Q}(\boldsymbol{x}) = \sum_{i=2}^{N}\sum_{k=0}^{K-1}\left|R_i[k] - \beta_i R_1[k]\exp\left(\frac{-\mathrm{j}2\pi\Delta\tau_i(\boldsymbol{x}))k}{K}\right)\right|^2, \tag{4.67}$$

where $\beta_i = \frac{\alpha_i}{\alpha_1}$ and $\Delta\tau_i = \tau_i - \tau_1$. Solving after β_i for the correct delay, reinserting in (4.67) and eliminating all terms independent of $\boldsymbol{x}$, the least squares solution is

$$\hat{\boldsymbol{x}}_{\mathrm{LS}} = \arg\max_{\boldsymbol{x}} \sum_{i=2}^{N} \frac{1}{||\boldsymbol{\Phi}_i(\boldsymbol{x})||^2} \left|\boldsymbol{R}_i^H \boldsymbol{\Phi}_i(\boldsymbol{x})\right|^2, \tag{4.68}$$

where

$$\boldsymbol{R}_i = [R_i[0], R_i[1], ..., R_i[K-1]]^T \tag{4.69}$$

$$\boldsymbol{\Phi}_i = \left[R_1[0], R_1[1]\mathrm{e}^{\left(\frac{-\mathrm{j}2\pi\Delta\tau_i(\boldsymbol{x})}{K}\right)}, ..., R_1[K-1]\mathrm{e}^{\left(\frac{-\mathrm{j}2\pi\Delta\tau_i(\boldsymbol{x})(K-1)}{K}\right)} \right]^T. \tag{4.70}$$

Results in [80] and [81] show an improved performance over regular two-step solutions for the AWGN scenario. The solution is grid-based, meaning that in the area of interest, grid points are defined and the value of the cost function is computed for each point. Since a high resolution needs a large number of grid points, the computational cost is much higher than that of usual two-step solutions, making the two-step solutions the reasonable choice for AWGN scenarios.

The use of the one-step method for multipath scenarios can, on the other hand, prove to be valuable. Compared to the AWGN signal model, there is an additional term that is added to the signal, and also to the cost function. This term is labeled interference in this method since the information about the unknown parameter $\boldsymbol{x}$ lies only in the first path. Again, assuming that the reference sensor receives one path ($P_1 = 1$). The cost function becomes

$$\bar{Q}(\boldsymbol{x}) = \sum_{i=2}^{N} \sum_{k=0}^{K-1} \left| R_i[k] - \beta_i^{(1)} R_1[k] \exp\left(\frac{-\mathrm{j}2\pi\Delta\tau_i^{(1)}(\boldsymbol{x})k}{K} \right) - I[k] \right|^2 \tag{4.71}$$

with $\beta_i^{(p)} = \frac{\alpha_i^{(p)}}{\alpha_1^{(1)}}$, $\Delta\tau_i = \tau_i^{(p)} - \tau_1^{(1)}$ and

$$I[k] = \sum_{p=2}^{P_i} \beta_i^{(p)} R_1[k] \exp\left(\frac{-\mathrm{j}2\pi\Delta\tau_i^{(p)}k}{K} \right). \tag{4.72}$$

The proposed algorithm is based on equations (4.68) and (4.71). Following, the steps are mentioned and a detailed explanation of each step follows:

1. If possible, estimate an initial position by cross-correlating the signals with the reference signal and identifying the first peak above a given threshold γ. Estimate the position using the TSE.

2. Eliminate the interference term $I[k]$ using a defined threshold γ.

3. Calculate weights for each sensor 2, ..., N depending on the outcome of step 2.

4. Use the interference-eliminated signals as well as the calculated weights to search for the weighted least squares solution according to (4.68). If step one was successful, the grid search area is reduced to a smaller area around the initial estimate. If not, the complete grid area is used.

In the first step, an attempt to reduce further computational cost is done by applying conventional two-step localization to obtain an initial guess. The grid search area can be reduced around the found guess, for example by defining a circle with a radius of 500 meters around it. Sometimes this step fails due to high errors caused by NLoS leading to matrix singularities.

In the second step, the cross-correlation between the reference sensor and all other sensors is calculated and peaks above a defined threshold are identified as received signal replica. Assuming one path component at the reference sensor, each later arriving signal path is labeled as interference and can be eliminated by estimating its delay $\widehat{\Delta\tau}$ and gain $\hat{\beta}$ and subtracting it from the received signal as

$$\bar{R}_i[k] = R_i[k] - \hat{\beta} R_1[k] \exp\left(\frac{-\mathrm{j}2\pi\widehat{\Delta\tau}k}{K}\right). \tag{4.73}$$

It must be noted here that this step results in noise amplification since the signal used to subtract the interference is noise corrupted. Therefore, this method should be applied whenever the reference signal has a high SNR and hence, $R_1[k] \approx \alpha_1 S[k] \mathrm{e}^{\frac{-\mathrm{j}2\pi\tau_1 k}{K}}$.

Figure 4.17 shows an example of a signal with three paths before and after step 2. The calculated correlation is normalized with the standard deviations of the signals. The correct term for this operation is the correlation coefficient. It is given by

$$C(\Delta\tau) = \frac{\mathbb{E}\left\{r_i(t), r_1^*(t+\Delta\tau)\right\}}{\sigma_{r_i}\sigma_{r_1}}. \tag{4.74}$$

Following the signal model in (4.1) while using $r_1(t)$ as the reference signal, the correlation coefficient at the delay belonging to the p-th multipath compo-

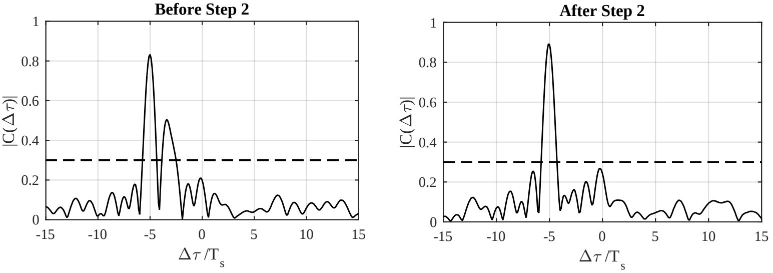


Figure 4.17: An example of a signal before and after step 2

nent $\Delta\tau_i^{(p)}$ results to

$$C(\Delta\tau) = \frac{\beta_i^{(p)}}{\sum\limits_{l=1}^{P_i} \left|\beta_i^{(l)}\right|^2}. \tag{4.75}$$

Taking values from 0 to 1 (after the absolute operation), a value close to one means that almost all the signal power is concentrated around the according delay value. Additional arriving signal replicas increase the denominator of the correlation coefficient, reducing it. Therefore, if the interference is eliminated correctly, the resulting correlation would have a high peak, optimally close to 1 but usually lower due to noise.

Thus, the weights are calculated as the correlation coefficient of the highest peak after applying the second step of the algorithm. In the example in Figure 4.17, the weight would be 0.85.

In the last step, the transmitter position is estimated by applying the grid-search algorithm

$$\hat{\boldsymbol{x}}_{\text{WLS}} = \arg\max_{\boldsymbol{x}} \sum_{i=2}^{N} \frac{a_i}{||\boldsymbol{\Phi}_i||^2} |\bar{\boldsymbol{R}}_i^H \boldsymbol{\Phi}_i(\boldsymbol{x})|^2, \tag{4.76}$$

where a_i is the weight of sensor i.

4.2.5 Estimation Bounds

Without knowledge about the NLoS error or its distribution, the lower bound for the position estimate in an NLoS scenario depends only on the LoS measurements as was proven in [82]. An optimal system would therefore require the identification of the biased measurements, followed by an ML solution using only LoS measurements.

However, this does not mean that any estimate using only LoS measurements will have a smaller RMSE than one using all available TDoAs. This is due to the fact that the CRLB describes the lower bound on the variance for an unbiased estimator. Estimators using all available measurements, including the biased ones, produce biased position estimates that can, however, have a smaller RMSE than the unbiased estimates. This is because the RMSE includes the standard deviation as well as the bias of an estimator. Using an additional NLoS measurement adds a bias that can, in some cases be smaller than the increase in standard deviation in case the measurement is not used.

4.2.6 Simulation Results and Analysis

Next, the previously described algorithms and different approaches to handling the position estimation in multipath propagation are verified using numerical simulations. Since the described three approaches can not be compared directly, their results are presented separately.

NLoS Identification and Elimination

The first part is concerned with mitigating NLoS by identifying the biased measurements and eliminating them or by weighting them accordingly. The results are compared to the case where all sensors are weighted equally as well as the case where only LoS sensors are used. For the simulation, N sensors are distributed on a circle with a radius of 1000 meters and the transmitter is placed on a grid shown in Figure 4.18. Each grid point was simulated 300 times, adding a random bias to N_{NLoS} randomly chosen measurements as well as a noise term with the covariance matrix form of (3.71). Generating positions within the shown square aims at avoiding bad geometries, which were discussed thoroughly in Section 3.2.

The RDs are generated following (4.51) with the bias and the noise standard deviation as parameters. The noise is generated according to (3.23) and assuming equal variances for all sensors ($\sigma_n^2 = \sigma_{d_1}^2 = \sigma_{d_2}^2 = \ldots = \sigma_{d_N}^2$). The biases are generated using exponential or uniform distributions, since these are widely used in the literature [72].

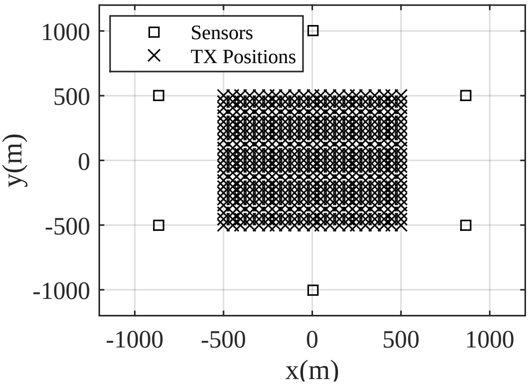


Figure 4.18: Simulated sensor and transmitter positions

The results compare the performance over growing noise standard deviation σ_n for the following algorithms:

- LS All: The TSE algorithm, or alternatively the Chan algorithm (in case the TSE fails) using all available measurements.
- Chen: Chen's algorithm using a minimum number of 4 sensors for a subset.
- Cong $\gamma = 0.9$: Cong's algorithm using a threshold of $\gamma = 0.9$.
- Cong known N_{NLoS}: Cong's algorithm using a known number of NLoS measurements.
- LS LoS: Like LSAll, but using only unbiased measurements.
- CRLB LoS: The CRLB using only unbiased measurements.
- CRLB All: The CRLB using all measurements (i.e., assuming they are all unbiased).

Aiming at differentiating the influence of various parameters on the estimation accuracy, a standard scenario is defined. By changing one parameter in the scenario, its influence can be demonstrated. The standard scenario is comprised of 6 sensors, 2 NLoS measurements and a bias generated using an exponential distribution with the mean value $\mu_b = 100$ m (see (4.51)). Figure 4.19 shows the performance of the mentioned algorithms over growing measurement noise.

The dashed line and the solid line show the CRLB using all TDoAs or using only unbiased TDoAs.

First let us compare the results of using all measurements to the performance using only unbiased measurements. For low measurement noise, eliminating the biased measurements does not largely degrade the estimation accuracy, since the hyperbolas almost intersect in one point at low noise and since there are enough LoS sensors (i.e., three for the two-dimensional case). With growing measurement noise standard deviation, the deterioration due to the lacking additional measurements becomes obvious. Looking at the RMSE of the LS estimates using all available measurements, including the biased ones, the curve crosses that of using only LoS measurements at $\sigma_n = 65$ m. After that, using the additional but biased TDoA improves the accuracy in terms of RMSE. However, recall that this method always leads to biased position estimates (even with the lower RMSE) while the other one obtains unbiased estimates with higher variance.

Chen's residual algorithm aims at reducing the bias and Cong's at identifying biased measurements and eliminating them. Comparing these two residual-based algorithms, Cong outperforms Chen, except for low noise scenarios. At low σ_n values, accurate TDoA estimates lead to an almost residuum-free subset using the unbiased measurements, while the biased subsets have a residuum and are therefore weighted much less.

In an optimal scenario, Cong's curve should be aligned with the LS LoS curve, but since Cong's residual does not always succeed in identifying the true biased measurements, it is not the case. Observing the difference between Cong's residual-based algorithm with a known number of biased measurements and with a threshold of $\gamma = 0.9$ (which obtained the best results from $0.6, 0.7, 0.8$, 0.9), knowing the number of biased measurements leads to slightly better results, except for low noise scenarios, where it clearly outperforms the threshold method. Table 4.2 shows the false alarm rate as well as the detection rate for two different thresholds as well as for a known number of NLoS measurements. The choice of $\gamma = 0.9$, leading to lower RMSE values is due to the low false alarm rate, which is more crucial for the overall performance. A false alarm means that an unbiased TDoA is mistakenly identified as a biased one and eliminated, while another biased TDoA can be kept for the position estimation, leading to a clear degradation of performance. Generally, Cong's algorithm often fails at identifying the biased measurements. This is due to the following: the assumed position for the calculation of the residuum in (4.56) is the true position. However, in real scenarios, a position is estimated and used as

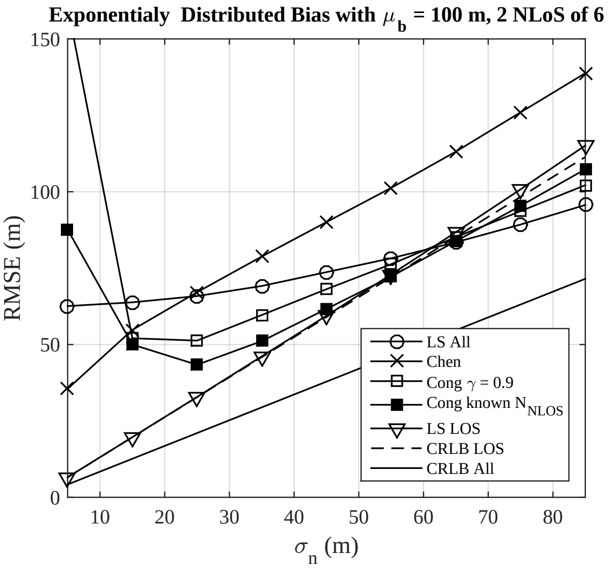


Figure 4.19: Position estimation performance for the standard scenario

	$\gamma = 0.6$	$\gamma = 0.9$	known N_{NLoS}
false alarm rate	21.36%	7.5%	17.75 %
detection rate	62.3 %	34.15 %	64.5 %

Table 4.2: False alarm and detection rate of the biased measurements using Cong's residual algorithm

reference for the calculation of the residuum. Since this reference is estimated using biased measurements, its residuals tend to be large, not providing enough information about whether they originate from NLoS measurements or from noise. There are two possible solutions for this problem: either to enhance the accuracy of the reference position, which is unrealistic since this is the goal of Cong's algorithm, or to incorporate the increased uncertainty caused by the inaccuracy of the reference position. This can be realized by enlarging the variance in (4.56). In Figure 4.20, the variance used for calculating the residuum was increased as $\tilde{\sigma_n}^2 = \sigma_n^2(1 + f)$, where f was the parameter. The figure shows how enlarging the variance enhances the accuracy, especially for low noise scenarios, in which mistaking a low-noise measurement for a biased one and eliminating it largely degrades the accuracy.

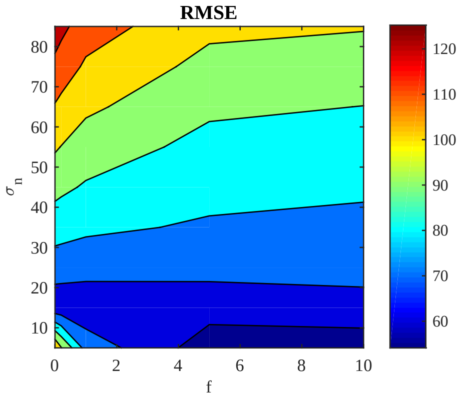


Figure 4.20: RMSE of the position estimate using Cong's algorithm with increased variance

In Figure 4.21, four simulation results are shown, each having one changed parameter from the standard scenario. The four main parameters determining the accuracy in NLoS scenarios are: the bias defined by its (i) mean as well as its (ii) distribution, (iii) the number of available sensors as well as (iv) the number of NLoS sensors.

By increasing the mean of the bias to $\mu_b = 200$m, the RMSE of the LS using all available measurements is almost doubled. A clear degradation can also be seen for Cong's algorithm for low noise standard deviation. Again, this is due to the fact that the residuum is unable to differentiate the biased measurements and almost randomly eliminates measurements, meaning that it oftentimes eliminates unbiased measurements and leaves the highly biased ones, resulting in the large RMSE. For higher noise standard deviation, the differentiation improves, leading to an obvious improvement over LS All.

By adding another biased sensor, the performance worsens for all residual-based algorithms, even though the number of unbiased measurements stays the same. The probability of identifying the true biased measurements for Cong's algorithm is lower and Chen's algorithm obtains estimates with higher biases since there is one additional biased TDoA. On the other hand, LS All slightly improves, crossing the LS LoS curve at 55 m instead of 60 m. This is mainly because oftentimes, the biases of the different hyperbolas can partly cancel out each other.

Given the same bias mean, exponentially distributed biases lead to poorer performance than uniformly distributed biases. This is due to the unbounded nature of the bias originating from an exponential distribution.

Increasing the number of biased measurements from two to three clearly worsens the results, especially that now the number of unbiased measurements is three, being the minimum number needed for a two-dimensional localization. That is why the performance of the LS LoS and the CRLB degrade, crossing the other algorithms at a much lower standard deviation (e.g., the LS All at $\sigma_n = 30m$).

Overall, the improvement by applying residual-based algorithms compared to using all available measurements depends on the bias scenario. For high bias values and sufficient unbiased measurements, Cong's algorithm can be applied to mitigate the error caused by NLoS.

Learning the Bias Using the Baum-Welch Algorithm

Introduced as a higher layer model and algorithm, the improvement by modeling the movement behavior using a hidden Markov model and the effect of learning the model parameters is analyzed here. For that, the map of a small part of Karlsruhe in Figure 4.15 on page 91 is utilized for the simulation scenario and the marked 68 points are defined as the Markov states. Neighboring states have a distance of 30 meters, leading to a quantization error. For the scenario, a Markov chain is generated randomly using the information about adjacent states (i.e., setting all non-adjacent transitions to zero and uniformly distributing the transition probability on all non-zero transitions). Assuming five sensors (i.e., four TDoAs), four biases are generated using an exponential distribution with mean value μ_b) for each of the 68 points. These are assumed to be fixed for each state throughout one Markov chain simulation. To generate the observations for the simulation of the Markov chain, the range differences resulting from the generated states are extracted using one sensor as reference. The according biases (being fixed for each state) as well as random noise with a standard deviation of 50 m are added following (4.51). The observation, being an estimated position, is then obtained by applying a LS algorithm to the resulting erroneous RDs.

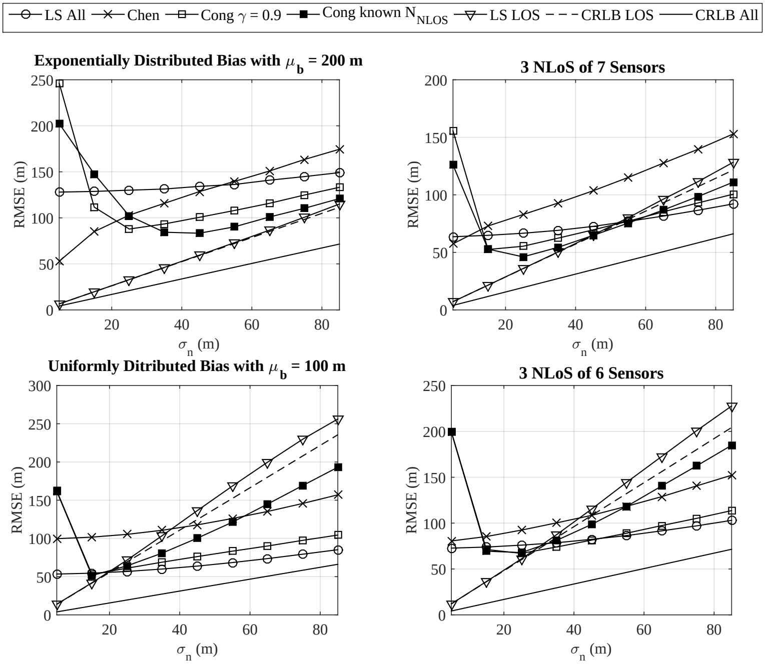


Figure 4.21: The influence of various parameters on the NLoS scenario

As a comparison, the results of the one-shot LS algorithm are shown, meaning that the biased and noisy range measurements are used as input at each time step to estimate the position, without considering any movement model. Recall that these estimates are the observations of the hidden Markov model and show directly the improvement by using this model. To compare to the conventional methods, the results using a Kalman filter are shown. The filter is given information about the movement as an uncertainty in its movement model, meaning that the Kalman filter is told that the movement in each step is within 30 meters of the current state. By that, a fair comparison can be made with the hidden Markov model, with the only difference being that the Markov model additionally incorporates adjacent and non-adjacent transitions in the various directions.

The results in Figure 4.22 show three curves for the Baum-Welch algorithm:

- BW Step 0: This is the result of using the forward algorithm (with no learning algorithm), showing the improvement solely by using the information about adjacent and non-adjacent transitions that is given from the street map.
- BW Step 9: Aiming at learning the model parameters (here, the error distribution of the observations as well as the initial states), this result shows the additional improvement by applying the Baum-Welch algorithm to an observation sequence. At each BW step, one sequence consisting of 1000 observations is used to try and learn the model parameters.
- BW Step 9 with Feedback: Since the Baum-Welch algorithm is an iterative algorithm that is initialized with random parameters, it bears the risk of converging to a local maximum of the likelihood function, having more than one local maximum. Therefore, giving the algorithm some feedback in the form of information about the true state can help prevent the algorithm from converging to a local maximum. This was done here by randomly choosing 50 of the 1000 observations that are fed to the algorithm as input and by giving the information about the true state of these observations. This can be interpreted as a form of training the model, but without an extensive training campaign.

The results in the figure show the RMSE of the estimate using the above described methods over growing mean value of the range difference bias. By adding the information given to the Kalman filter, the RMSE decreases from 325 m to 250 m using the Kalman filter for a bias mean of $\mu_b = 350\text{m}$. By incorporating the map information using the Markov model, this result is improved to 150 m. Additionally, learning by observing longer sequences slightly improves the result to 130 m. Finally, helping the Baum-Welch algorithm by giving it feedback largely improves the result up to 53 m.

One-Step Localization

As an alternative approach to localize a transmitter in a multipath environment, the one-step algorithm that was presented in [79] and described in Section 4.2.4 is analyzed. Its main idea is to combine the different signals after eliminating the interference caused by the later arrived signal replicas. By doing so, any remaining interference is entered as soft information in the position location. For correlation-based algorithms (e.g., First Peak detection), this interference caused by multipath propagation results directly in biased TDoA estimates,

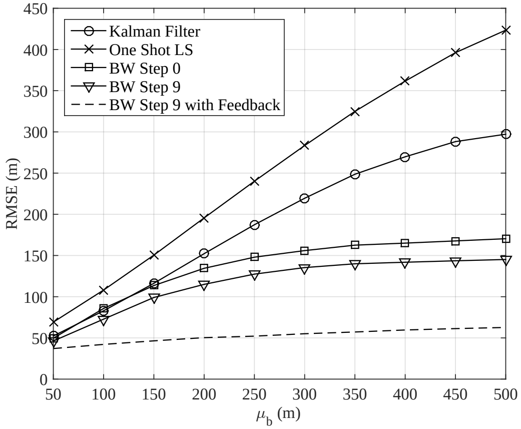


Figure 4.22: Improved positioning performance by using and learning the hidden Markov model

resulting in a position bias. Compared to a two-step localization using the ML-based WRELAX algorithm, the one-step approach does not fully resolve the signal replica, it rather reduces the interference. Hence, its advantage over a two-step localization with WRELAX lies in the reduced computational complexity.

Running the simulation on a 3GHz-AMD Athlon II X2 250 desktop PC, a scenario with six sensors, five sensors being propagated through a multipath channel each having six path components, was tested. The WRELAX algorithm was given the true number of multipath components (i.e., the additional computation time by assuming a higher number of components and searching for the minimum description length is neglected, see Section 4.1.4)). The two-step localization using the WRELAX algorithm for four signals, followed by the TSE algorithm for position estimation needs, on average, 29 seconds. The one-step localization described in 4.2.4, on the other hand, requires 6.8 seconds. A correlation-based algorithm followed by the TSE needs 0.04 seconds.

Figure 4.23 shows the cumulative distribution function of the estimation error for two scenarios. On the left side, a signal with a symbol length of $T_s = 2\mu s$ is received by six sensors, five having multipath propagation. On the right side, the symbol duration is increased to $4\mu s$, making its bandwidth half of the sig-

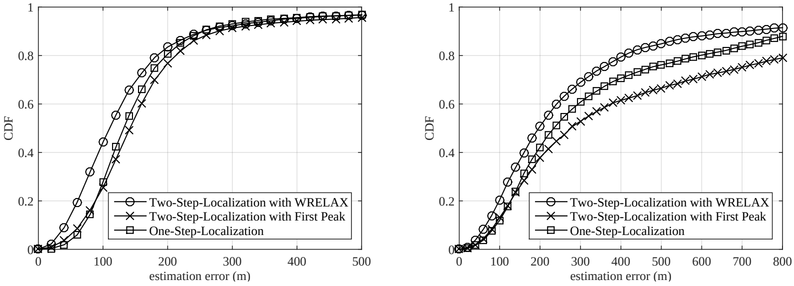


Figure 4.23: Performance of the one-step localization algorithm in multipath environments left: $T_s = 2\mu s$; right: $T_s = 4\mu s$

nal in the left scenario. The multipath channels are chosen randomly from four Cost 207-models for urban scenarios, each having six multipath components. On the left side, the algorithms perform almost equally, with the two-step localization with WRELAX having a slight improvement over the two other algorithms and the one-step localization being slightly better than the two-step algorithm. On the right side, however, the difference between the algorithms is more obvious. While the estimates of the two-step algorithm with WRELAX have an error below 300 meters 70 % of the time, only 61% of the one-step algorithm estimates and only 52 % of the estimates using the two-step algorithm with First Peak show this behavior. Hence, if the system requires a real-time position estimation and does not have the computational capacity to meet these requirements while applying the WRELAX algorithm, the one-step solution can be used instead, offering a trade-off between computational complexity and accuracy, especially for harsh multipath environments.

4.3 System Design

To sum up this chapter, Figure 4.24 shows the processing chain in multipath scenarios. Here, the priorities have changed for choosing the reference sensor. Now it primarily depends on the number of multipath components in the received signals, making the geometry a marginal aspect. After determining the reference sensor, a trade-off between accuracy and computational complexity decides which algorithms follow. If there are no restrictions on the computational power, the WRELAX algorithm, combined with the MDL algorithm should be applied for the time delay estimation. If it is too cumbersome for the computing entity and if the reference signal has a high SNR, then one-step localization shall be realized. In the worst case scenario, correlation-based methods are to be used.

Depending on the applied algorithm for time delay estimation and on the associated expected error, the position estimation can be carried out using LS solutions like the TSE. A suspected biased outcome of the delay estimation can be mitigated using Cong's algorithm.

In the last step, the estimated positions are tracked. The default algorithm here is the Kalman filter. However, whenever certain assumptions can be made about the movement behavior of the transmitter (e.g., limiting its movement to the streets), the accuracy can be enhanced using the hidden Markov model. Optimally, the transmitter can be observed long enough to learn the parameters of the environment.

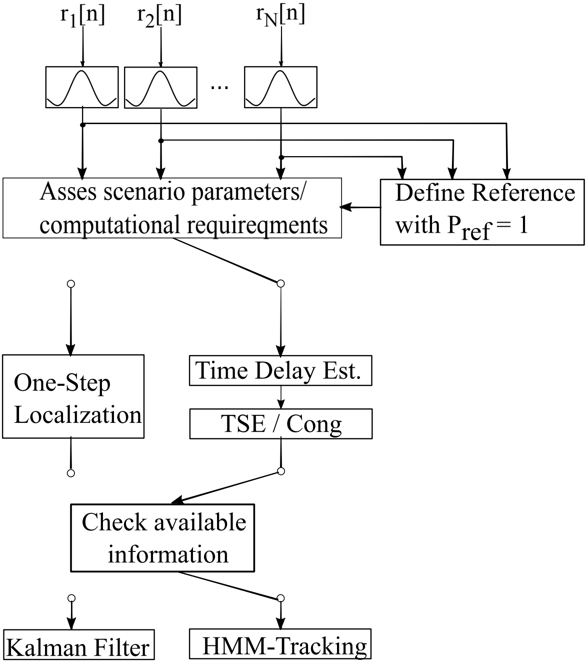

r1[n]
r2[n]
rN[n]
...
Asses scenario parameters/ computational requireqments
Define Reference with Pref = 1
One-Step Localization
Time Delay Est.
TSE / Cong
Check available information
Kalman Filter
HMM-Tracking

Figure 4.24: Smart TDoA system in multipath propagation

5 System Setup and Verification

To assess the applicability of a low cost hardware setup as the core of a smart TDoA system, 5 identical sensors were designed and installed on KIT's south campus. The sensors were developed within the scope of the research project *Intelligente Sensoren für die Digitale Dividende (ISDD)*, together with the project partner *LStelcom* and sponsored by the German Federal Ministry of Education and Research [1]. By having synchronized clocks as well as the ability to operate on variable carrier frequencies and sampling rates, the sensors fulfilled the requirements of a TDoA system.

In this chapter, the configuration of the measurement system, including the sensor components, the geometrical setup as well as the steps between receiving the signal and applying the described algorithms are discussed. In [83] and [84], some results were presented using this setup. Here, two measurements with different scenarios are depicted to demonstrate the performance of the designed TDoA system.

[1] Grant 01BU1031

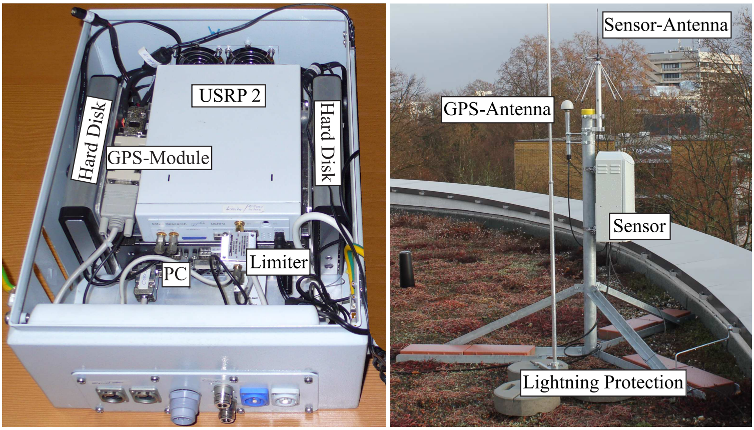


Figure 5.1: TDoA sensors, reprinted with permission from [85]

5.1 Measurement Configuration

5.1.1 Sensor Setup

Figure 5.1 shows one of the five identical sensors that were built for the measurements. On the left side, the core components can be seen. The main element of the sensor is the second generation Universal Software Radio Peripheral (USRP) [86]. This software radio consists of an exchangeable daughterboard, handling the analog operations such as the down and up conversions as well as filtering, and a motherboard handling the baseband processing of the signal. The incoming signal from the respective daughterboard is received by the motherboard and sampled at a rate of 100 MSps. Each yielded complex IQ-sample consists of 32 bits. Further processing of the signal either occurs on an FPGA on the motherboard or on the host computer, whereas the data is transferred via Ethernet. Therefore, decimation filters are applied to reduce the sampling rate, having a decimation factor ranging from 4 to 512, allowing the effective bandwidth to be in the range of 195.3 kHz to 25 MHz. For the measurements, a WBX-daughterboard [87], operating in the range of 50 MHz to 2.2 GHz, was chosen.

The second and more crucial component of the TDoA system is the GPS module. It serves as the common time reference for all sensors, enabling precise time synchronization. This is done by tagging the sampled data with time stamps using a PPS signal, a pulse that accurately signalizes the start of a second. Additionally, the module is used to discipline the internal clock of the USRPs, since timing errors caused by local oscillator offsets can lead to large inaccuracies. The chosen module is the Mini-T module from Trimble [88], offering a PPS signal accuracy of 15 ns for the time stamps, as well as a 10 MHz reference signal for the local oscillators with an accuracy of 1.16×10^{-12} (one day average). The GPS module is also connected to the PC through a RS-232 port, providing the information about the position of the sensor.

Each USRP is connected to and controlled by a compact PC, setting the following parameters to record the IQ-data:

- Carrier frequency
- Sampling rate
- Receive gain
- Number of samples per saved file

During a measurement, the PC receives the time-stamped IQ-data from the USRP via Ethernet and stores them in the hard disks, each having a capacity of 2 TByte to enable longer recording times.

The connections between the different sensor components are displayed in Figure 5.2.

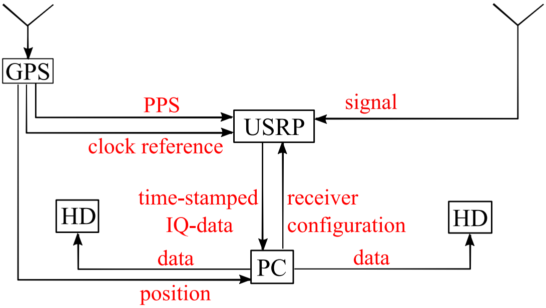


Figure 5.2: System components of the installed sensors

5.1.2 Geometrical Setup

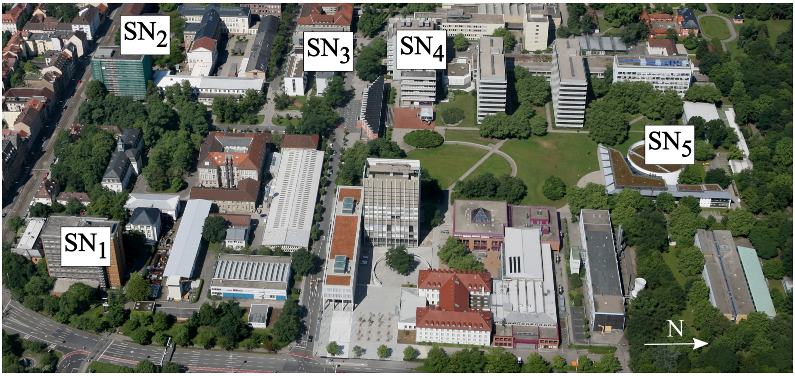


Figure 5.3: Installed sensors on campus buildings as seen from the east

The positions of the installed sensors as well as TX determine the GDoP (3.60) of the system, defining how much a measurement error degrades the position estimation accuracy. As was discussed in Section 3.2.4, flat bearing angles result in higher GDoP and hence, larger errors. An optimum constellation, on the other hand, where the sensors are aligned on a circle around the transmitter, results in the best possible GDoP. Oftentimes, due to the structure of buildings or other limitations, the optimal setup can not be achieved. This was also the case when the sensors were being installed on the university rooftops. Some buildings did not have a suitable space on their rooftops, others were being renovated. Thus, the final setup as can be seen in Figure 5.3 is a realistic setup that is far from the optimal constellation.

Figure 5.4 shows the setup in its true orientation as well as the hyperbola constellation for the shown transmitter position, a relatively centered position. On the right side, the uncertainty region for RD-errors between ±20 m can be seen. The uncertainty in the x-axis is higher due to the flatness of the intersection angles in the x-axis.

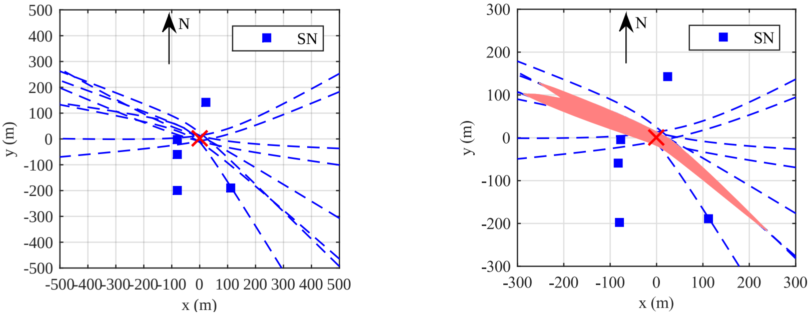


Figure 5.4: Hyperbola constellation of the sensor-transmitter geometrical setup with a 20 m error

5.1.3 From the Sensors to the TDoA System

Two aspects need to be considered before applying the estimation algorithms on the received data:

1. Synchronizing the time-stamped IQ-data gathered from all the sensors.
2. Transforming the received coordinates of the sensors to enable using them in the described algorithms.

The recorded data is gathered from all sensors by a central unit for further processing. Each measurement from one sensor, defined by a given data length, results in three files:

- *info(recording time).json*: This file contains the sensor parameters, including carrier frequency, sampling rate, sensor name and its GPS-position.
- *meta(recording time).bin*: This file contains the GPS-time stamps given in unixtime and paired together with the sample index belonging to each time stamp
- *data(recording time).bin*: This file contains the according IQ-data of the sensor.

To synchronize the data, a block consisting of K samples is imported from each sensor using the recording time given in the file names as well as the time-stamps included in the .meta-file. This results in sample-synchronized data blocks from all sensors. The remaining subsample time-offset (being smaller than the sampling rate) can be corrected when the TDoAs are estimated.

Concerning the transformation of the coordinates, geographic coordinates of the sensors are given in latitude, longitude and height, according to the World Geodetic System 1984 (WGS 84). This system uses a reference ellipsoid with the semi-major axis $a = 6378138$ m, semi-minor axis $b = 6356752.314245$ m and an inverse flattening $1/f = 298.257223563$. Using these parameters, a point on or close to the earth is given as function of longitude, latitude and height $P(\phi, \lambda, h)$, where h is the height above mean sea level. Transforming that to the geocentric Cartesian coordinates, the so called Earth-Centered Earth-Fixed (ECEF), the point is then given as $P(X, Y, Z)$ by using the following transformation:

$$\begin{aligned} X &= (N(\phi) + h)\cos(\phi)\cos(\lambda) \\ Y &= (N(\phi) + h)\cos(\phi)\sin(\lambda) \\ Z &= \big(N(\phi)(1 - e^2) + h\big)\sin(\phi), \end{aligned} \tag{5.1}$$

where

$$\begin{aligned} e^2 &= 2f - f^2 \\ N(\phi) &= \frac{a}{\sqrt{1 - e^2\sin^2(\phi)}}. \end{aligned} \tag{5.2}$$

Aiming at having a system that is intuitively pointing in the right direction, i.e., aligned with the geographic north and east and with the altitude pointing in the right direction, the origin of the system can be shifted to a point near the earth plane in the observed region and an additional rotation can transform the geocentric system to a so called local tangent system. To do that, a reference point $\boldsymbol{X}_0$ in the area of interest (e.g., the center of all sensors) is defined and, using its latitude λ_0 and longitude ϕ_0, the final transformed coordinates $[X', Y', Z']^T$ resulting from $[X, Y, Z]^T$ can be calculated as

$$\begin{pmatrix} X' \\ Y' \\ Z' \end{pmatrix} = \begin{pmatrix} -\sin(\phi_0) & \cos(\phi_0) & 0 \\ -\cos(\phi_0)\sin(\lambda_0) & -\sin(\lambda_0)\sin(\phi_0) & \cos(\lambda_0) \\ \cos(\lambda_0)\cos(\phi_0) & \cos(\lambda_0)\sin(\phi_0) & \sin(\lambda) \end{pmatrix} \begin{pmatrix} X - X_0 \\ Y - Y_0 \\ Z - Z_0 \end{pmatrix} \tag{5.3}$$

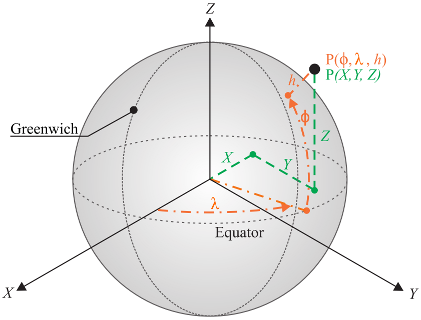


Figure 5.5: WGS 84 and ECEF coordinate systems

5.1.4 Transmitter and Ground Truth

The transmitter comprises another USRP2 with a WBX daughterboard connected to an amplifier, enabling a quality of the receive signal even for larger distances. The signal is generated and preprocessed on a host laptop, and transferred to the USRP via Ethernet for transmission.

To be able to evaluate the estimation accuracy, a Garmin GPS tracker [89] accompanied the transmitter to serve as ground truth for the estimated position (see Figure 5.6).

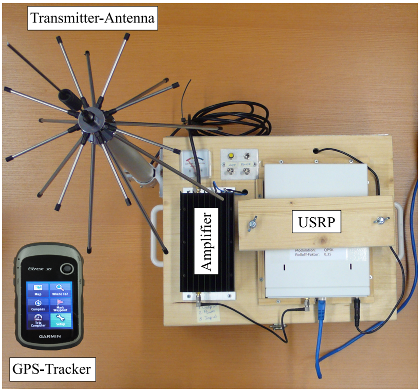


Figure 5.6: Transmitter components and ground truth

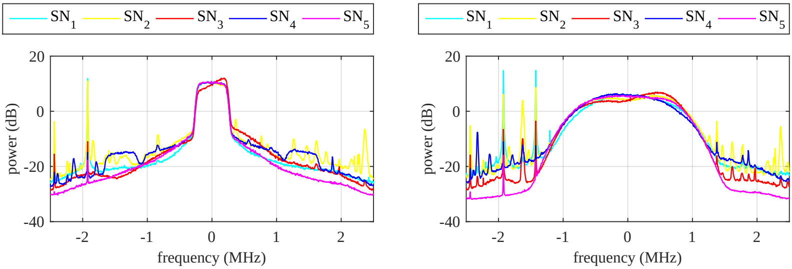


Figure 5.7: Spectrum of the received signals at the sensors; Left: 500 kHz signal, right: 2 MHz signal

5.2 Scenario 1: Stationary Transmitter in AWGN

The first measurement aims at verifying the applicability of low-cost sensor networks for a smart passive TDoA system as well as at testing the basic functionality of the positioning system. A simple scenario was therefore chosen for these tasks, consisting of a stationary transmitter standing in the position shown in Figure 5.4 expected to have LoS to three out of the five operating sensors and NLoS to SN_1 and SN_3.

The transmitted signal consisted of random, lowpass filtered data with two different symbol rates (500 kHz and 2 MHz) and a carrier frequency of 431 MHz. The distances between the transmitter and the sensors varied between 80 m and 220 m. Figure 5.7 shows the spectra of the received signals. Apart from the transmitted signal, spectral peaks resulting from other signals can be seen. Sensors having larger distances to the signal source receive a weaker signal as can be seen in the figure. Additionally, fading can be observed at SN_3 and SN_1.

The slope of the spectra outside the signal of interest makes the assumption of white noise a vague one. Nevertheless, the noise level can be approximately seen to be 20 dB lower than the 500 kHz signal. At a received bandwidth of 5 MHz and a signal bandwidth of 500 kHz, this results to an SNR of 10 dB. For the 2 MHz signal, assuming the noise level at around -15 dB, the signal is 20

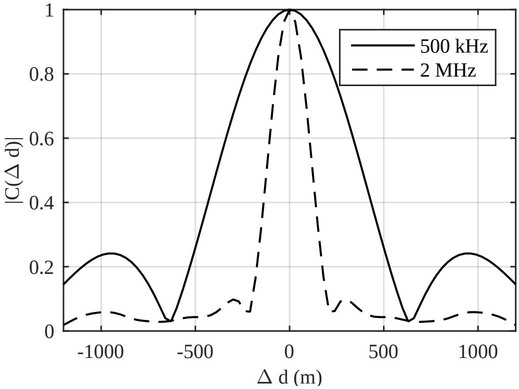


Figure 5.8: Cross-correlation of signals with different bandwidths

dB higher than the noise, resulting in an SNR of 16 dB before filtering. Thus, after filtering, both signals should have approximately equal SNRs of 20 dB.

To show the influence of a larger bandwidth, we compare the results of the 500 kHz and the 2 MHz signals, using an equally long observation window of 1 ms at a sensor sample rate of 5 MHz, i.e., 5000 samples per estimate. Noise and other unwanted signals are filtered, leading to the previously discussed nearly equal SNRs. Afterwards, the signals are cross-correlated using a random sensor as reference at first. The TDoA estimates using the 2 MHz signal are expected to have a lower estimation variance due to the higher bandwidth and hence, the position estimate is expected to have a lower variance than the 500 kHz signal. Figure 5.8 demonstrates the effect by showing an example correlation of both signals. The x-axis shows the respective range difference that will be used in the position estimation algorithm. An error caused by noise or other sources results in a slight peak shift of the correlation. However, a slight peak shift means that it lies in the magnitude of the peak width, which would mean a four times higher error for the 500 kHz signal at an equal relative shift in the peak.

Figure 5.9 shows the histograms of the estimation errors of the range differences $\Delta d_{4,5}, \Delta d_{1,5}$, i.e., using SN_5 as the reference sensor. These two were chosen to represent the case of an AWGN channel estimate $\Delta d_{4,5}$ and a multipath channel estimate $\Delta d_{1,5}$. As expected, estimates resulting from the 2MHz signal have a smaller variance than those resulting from the 500 kHz signal,

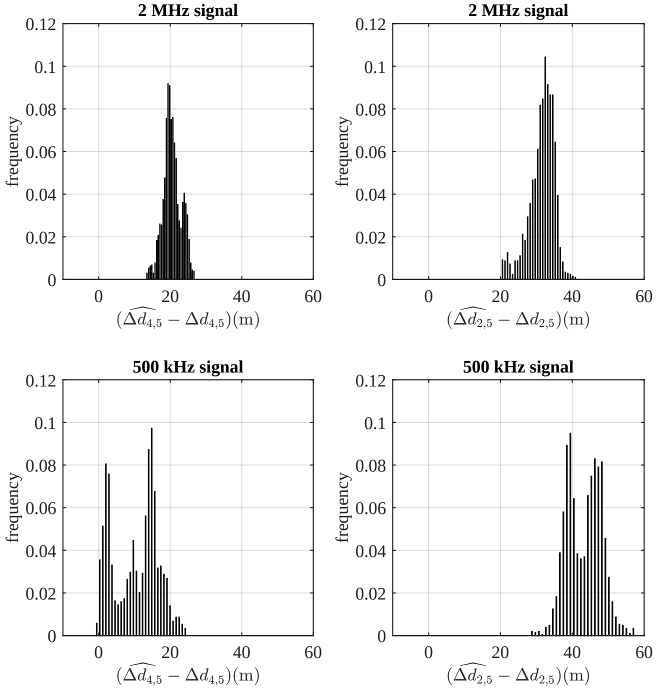


Figure 5.9: Histograms of the range difference error at two different sensors

since an equally large observation window is used. In an ideal measurement system, the left side estimates should be centered around a zero-error, and the right side estimates are expected to be biased. However, all estimates are biased. This is due to the fact that additional measurement inaccuracies occur, primarily due to the following:

- Timing inaccuracies: these can occur due to drifts of internal clocks of the receivers as well as the inaccuracy of the PPS-pulse (up to 30 ns resulting from adding the timing inaccuracies at two sensors).
- Position inaccuracies: the reference values for the error estimation are calculated using the information about the transmitter positions, resulting from the GPS tracker measurements, as well as the sensor positions,

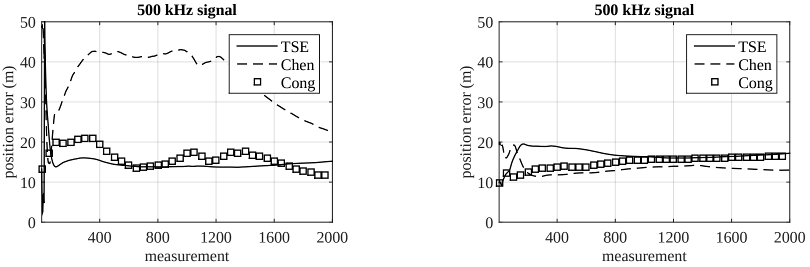


Figure 5.10: Position estimation error; Left: 500 kHz signal, right: 2MHz signal

given by the sensor GPS module. These measurements are not perfect, leading to imperfect reference values.

Hence, a realistic modeling of the estimated RDs is

$$\widehat{\Delta d}_{i,j} = \Delta d_{i,j} + b_{\text{NLoS}} + b_{\text{measurement}}. \tag{5.4}$$

After having an initial estimate of the position, (3.62) can be applied to redetermine the reference sensor to maximize the estimation accuracy. The estimated TDoAs are used, together with the transformed coordinates of the sensors, to calculate the position using the TSE algorithm, optionally accompanied by NLoS mitigation algorithms (Chen or Cong's residual-based algorithms presented in Section 4.2). Since the transmitter is not moving, sequential measurements are averaged to enhance the accuracy and stabilize the estimate.

Keeping in mind the TDoA biases, Figure 5.10 shows the position estimation error over increasing number of measurements.

The results show robust behaviors both for the TSE as well as Cong's algorithm. Since all measurements are biased and since there are only five sensors, there is an error floor around 15 meters for both measurements. Although the TDoA estimates have higher variance using the 500 kHz signal, the estimated positions are equally, or even more accurate than for the 2 MHz signal. The key factors in this measurement, since the SNR of the signal is 20 dB and since the observation window is relatively long, are the remaining biases resulting from measurement inaccuracies, which are independent of the signal bandwidths.

Figure 5.11 shows the true position on the map as well as the estimated positions using the TSE algorithm.

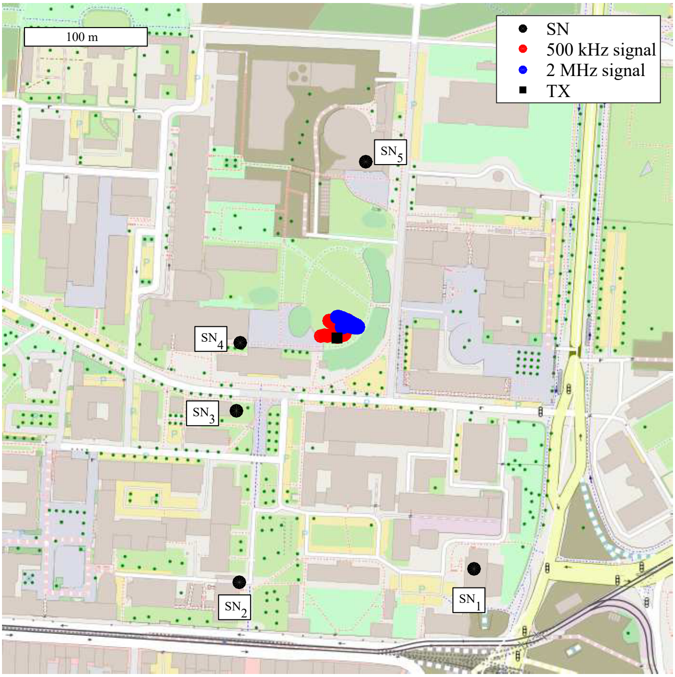


Figure 5.11: Estimated positions of measurement scenario 1

5.3 Scenario 2: Moving Transmitter in a Multipath Channel

Next to showing the basic functionality of the system, its stability as well as its limitations are of importance. A random, lowpass signal with a symbol rate of 2 MHz was transmitted at low power, using a carrier frequency of 431 MHz. The transmitter followed the black track shown in Figure 5.12, within an overall duration of 10 minutes. LoS connections to maximum two sensors at most positions are expected as well as shadowing resulting from surrounding buildings or other obstacles. Next to the map in Figure 5.12 on the right side, the spectra of the received signals at three marked points on the track (green squares) is demonstrated. Each plotted spectrum is colored like its respective sensor marked in the figure.

The quality of the received signals has clearly degraded in comparison to the first scenario. The signal of interest, lying at the center of the received spectrum with a bandwidth of 2 MHz is very little above the noise level and is weaker than some interferers. Additionally, shading effects can be observed.

Looking at the three observed positions, the first one (bottom) shows a relatively good reception at SN_2, SN_3 and SN_4. SN_5 receives a weak signal as well as interference. Hence, it is expected that around this position, maximal three TDoAs can be estimated. In the second point, all five sensors receive the signal moderately well. Shading effects are not dominant and, hence, the estimated TDoAs are expected to be stable and reliable. In the third point, the transmitter lies outside the region surrounded by the sensors and is additionally surrounded by buildings and other objects. Only two sensors received the signal.

Therefore, in this scenario, additional aspects need to be considered in the localization system:

1. Eliminating low-SNR measurements: Including a sensor that does not receive the signal severely degrades the estimation accuracy and destabilizes the tracking process. Therefore, low-SNR sensors must be eliminated by energy detection. This is done by cross-correlating the received signals with the reference signal and eliminating sensors that do not have a correlation peak above γ.

2. Choosing the reference sensor is of high importance in low-SNR and high shadowing scenarios. Here, it is not about trying to minimize the

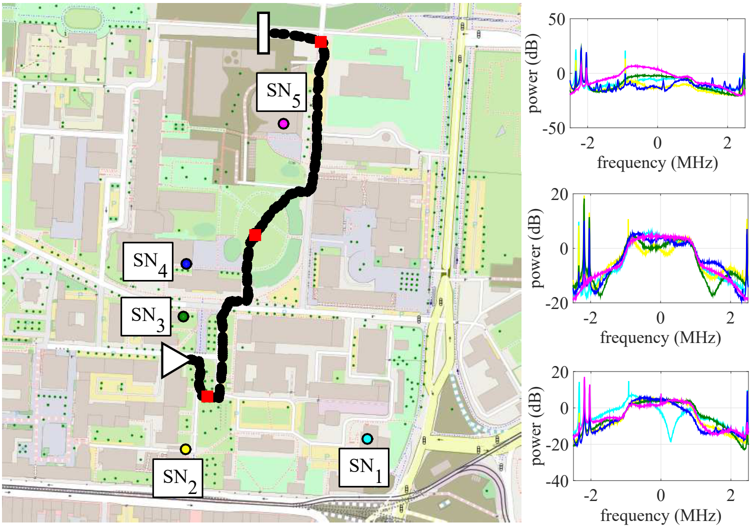


Figure 5.12: Transmitter track of scenario 2 and received signal spectra

GDoP, since the instability of the system originates from its disability to estimate TDoAs due to the low SNRs. Therefore, sensors are to be chosen that show high correlation peaks to as much sensors as possible.

3. In dynamic scenarios, not only is the priority of choosing the reference sensor a different one, but the frequency of re-estimating and re-determining the reference sensor is higher due to the changing environment from one position to the next. Therefore, each 30 seconds, the reference sensor was re-defined based on the quality of the received signals.

Since the number of sensors receiving the signal is maximum four most of the time, residual-based algorithms are not suitable. Chen's algorithm requires a minimum subset size of four sensors, making the number of subsets equal one. Cong's algorithm aims at eliminating NLoS sensors, making the instability of the system much higher since three sensors are the absolute minimum number of required sensors.

On the other hand, the discussed multipath resolving algorithms, i.e., the WRELAX algorithm from Section 4.1.3 or the weighted one-step-localization in Section

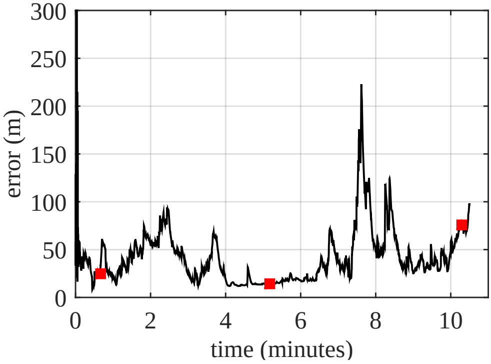


Figure 5.13: Position estimation error of measurement scenario 2

4.2.4, both have shown to be unrobust in low SNR scenarios. Therefore, in this scenario, only basic TDoA algorithms can be applied, i.e., cross-correlation and peak detection, followed by a least squares estimation using the TSE algorithm and a Kalman filter for tracking.

Figure 5.13 shows the estimation error of the track, specifically, the three discussed points of Figure 5.12 are marked by the green squares. The error around the first point stays around 50 meters. The second point shows a stable error of 20 meters and the third point shows the worst performance. The error peak at the beginning of the plot is due to the fact that the Kalman filter needs an initial phase to be able to estimate the system parameters. The peak at minute 7.30 is due to the fact that almost none of the sensors were receiving the transmitted signal since the street is surrounded by buildings and trees around this position.

Figure 5.14 shows the estimated track compared to the true one on the map.

5.4 Summary

In this chapter, the designed TDoA system based on low-cost off-the-shelf hardware was presented. The additional steps between receiving the data and applying the algorithms were explained.

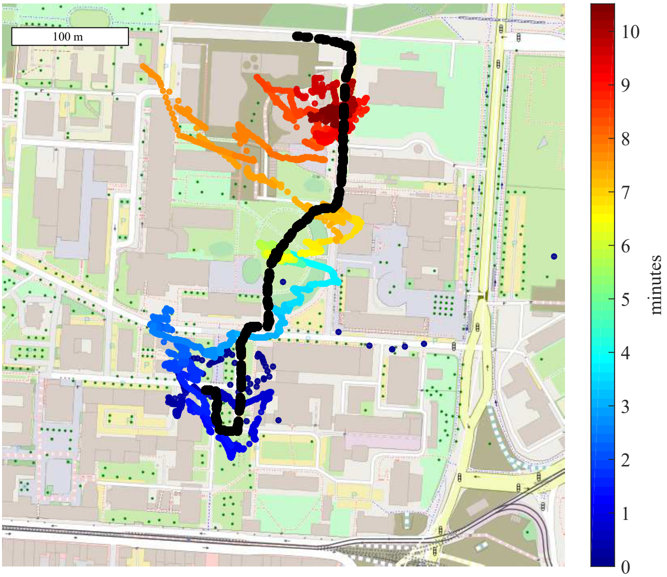


Figure 5.14: Estimated track of measurement scenario 2

The first measurement aimed at showing the basic functionality of the system. It was shown that, due to measurement errors caused by time inaccuracies and inaccuracies caused by erroneous sensor or ground truth positions, the error floor stayed at around 15 meters.

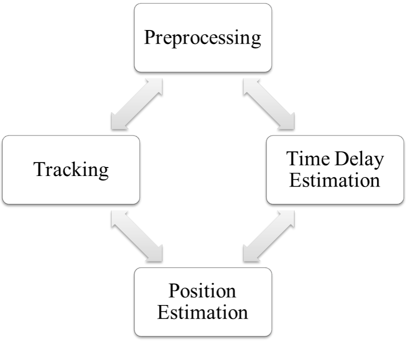
In the second scenario, the limitations of the system were tested by transmitting a weak signal that could not be received by all sensors at once most of the time. Therefore, it was shown how in real world scenarios, additional measures need to be taken to assure a certain stability of the system, i.e., by a well chosen reference sensor, well chosen algorithms as well as energy detection to eliminate senseless measurements.

6 Summary and Conclusion

Passive localization describes the process of seeking the position of a signal source with no cooperation from the transmitter. Such a system should be based on the assumption that the unknown signal of interest is not designed for localization, i.e., by not necessarily having a large bandwidth and by being unknown to the system. These two additional challenges must be considered in the system design.

With no possibility to change the transmitted signal, the localization system must try to gain as much information as possible from the received signals. In this work, methods suitable for passive scenarios were developed and presented, theoretical bounds were derived and an overall system design was depicted and analyzed. The four main blocks of a passive TDoA localization system are preprocessing, delay estimation, position estimation, and tracking. A smart system needs to exchange the information gained by each block to optimize the estimation result. The scope of the thesis followed this principle for the design of the TDoA system.

Starting with the simple AWGN scenario, Chapter 3 presented the known time delay estimation method of cross-correlation and peak detection, focusing on the necessity of interpolation, by preprocessing the signal and/or by post-processing the estimated discrete correlation. Although quadratic interpolation is a popular technique, it always results in an error, primarily due to the parabolic



misfit. However, combined with upsampling techniques, it can approach the CRLB.

Regarding the position estimation in AWGN-scenarios, two popular algorithms were described and compared, whereas the TSE method showed the overall better results. To combine sequential measurements, Kalman filters offer a solution that is robust against outliers and that is suitable for non-stationary scenarios. However, for stationary scenarios, a simple averaging is just as accurate.

In addition to combining sequential TDoA measurements, the utility of the unscented Kalman filter for tracking the position by combining TDoA and RSSD measurements was presented as a new approach and its results attained the CRLB. In case of high TDoA errors due to timing inaccuracies, estimated RSSDs can be incorporated simply by applying the UKF.

A far more challenging scenario for localization are multipath channels, where multiple replica of the same signal arrive with a relative delay at the receiver. Assuming narrowband signals, closely arriving paths result in largely overlapping correlation peaks, shifting the estimated correlation peak from the true TDoA value. Chapter 4 presented methods based on the ML solution to the problem of estimating time delays of a multipath propagated signal using a reference signal. The theoretical bounds were derived and analyzed in comparison to the case of a known signal. Simulation results showed an improved estimation accuracy for the presented algorithms, however not for largely overlapping multipath components. Considering the trade-off in computational complexity and the associated latency, these algorithms were found to be rather impractical for implementation.

Taking the conclusion of Section 4.1 into account, a biased estimate as a result of unresolved multipath delays was the assumed system model in Section 4.2. The solution to this problem are bias estimation and mitigation techniques, aiming at reducing the weight of the biased estimates in the position estimation algorithms or, optimally, identifying and discarding them. For these algorithms to work, sufficient sensors with unbiased TDoA estimates must exist. On the other hand, some scenarios showed better results regarding RMSE by using all available measurements, including the biased ones. This justifies the fact that in some scenarios, including biased TDoA estimates can be preferred, even if it leads to biased position estimates.

An alternative solution to the bias introduced by signals in multipath channels is to learn this bias. This approach was introduced using hidden Markov models. Assuming a certain movement behavior of the transmitter, e.g., staying on the streets, observing the transmitter for a long time can be used to learn the bias environment by applying the Baum-Welch algorithm. Whereas the idea of using hidden Markov models to enhance position estimation accuracy has been presented in the past, using these models to learn the biases of the estimates by applying the Baum-Welch algorithm was new. The results showed that, due to the complexity of the system, learning the model parameters must be supported by feedback to ensure the convergence of the algorithm to the global, and not the local, maximum of the likelihood function.

Since the delay resolving algorithms were found to be impractical, the weighted one-step localization method was presented as a compromise between computational time and estimation accuracy for multipath propagation scenarios. This method estimates the position directly from the signals. While this approach has been presented in the past for AWGN signals, it has not been used before in multipath propagation scenarios. To mitigate errors caused by multipath propagation, the algorithm introduces a signal preprocessing step similar to interference cancellation, and weights the signals according to the outcome of this step. Its results showed an improvement over low-complexity, but rather inaccurate correlation-based methods in multipath scenarios.

Chapter 5 expanded the analysis to real measurements. The sensors designed for the measurement campaign composed of low-cost software defined radios, synchronized using GPS time-stamps, were presented. To yield accurate estimates in real scenarios, additional aspects must be taken into account, among them the choice and the rate of changing the reference sensor as well as choosing the suitable estimation algorithms. Adding these steps, the measurements obtained relatively good results even in harsh environments and low SNR scenarios.

All in all, a TDoA system design was developed and provided with new algorithms. Relations between the four blocks of the system were established and analyzed in order to obtain the best possible estimate under the given conditions.

$\mathcal{A}$ Selected Topics on Estimation Theory

Localization is an estimation problem. Throughout this research, some important definitions of estimation theory are used and are therefore described here.

A.1 Maximum Likelihood Estimation

Suppose we have a statistical model of the form

$$\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{s}; \theta) + \boldsymbol{w}, \tag{A.1}$$

where $\boldsymbol{x}$ is our observation quantity, $\boldsymbol{s}$ is some known quantity, θ is a parameter we want to estimate and $\boldsymbol{w}$ is the random observation error. When the probability density function of $\boldsymbol{x}$ is viewed as a function of the unknown parameter (with $\boldsymbol{s}$ fixed), it is termed the likelihood function [28]. The maximum likelihood (ML) estimator chooses the value of the parameter that maximizes the function

$$\hat{\theta}_{\text{ML}} = \arg\max_{\theta} \left\{ p(\boldsymbol{x}; \theta) \right\}, \tag{A.2}$$

where $p(\boldsymbol{x}; \theta)$ is a family of probability density functions, parameterized by θ and termed the likelihood function. The ML estimator is the most used approach for all practical estimators and is optimal in terms of minimum mean square error.

A.2 Bias

The expected error of the estimate is defined as the estimator bias

$$b[\hat{\theta}] = \mathbb{E}\left\{\hat{\theta}\right\} - \theta. \tag{A.3}$$

An unbiased estimator yields, on average, the true parameter and has zero bias [28].

A.3 Cramér Rao Lower Bound

The variance of any unbiased estimator for a parameter θ must satisfy

$$\mathrm{var}[\hat{\theta}] \geq \frac{1}{-\mathbb{E}\left\{\frac{\partial^2 \ln p(\boldsymbol{x};\theta)}{\partial \theta^2}\right\}}. \tag{A.4}$$

This is called the CRLB [90]. An unbiased estimator that reaches the CRLB is said to be *efficient* in the sense that it efficiently uses the available data. The CRLB is widely used as a measure of estimator quality. The denominator in (A.4) is the so called *Fisher information* $F(\theta) = -\mathbb{E}\left\{\frac{\partial^2 \ln p(x;\theta)}{\partial \theta^2}\right\}$. The Fisher information is additive for independent observations.

Throughout this thesis, the derivation of the CRLB is needed for the following system models:

- For a real signal in white Gaussian noise given as:

  $$x[n] = s[n;\theta] + w[n], \quad n = 0, 1, ..., N-1 \tag{A.5}$$

  and $w[n]$ being the added zero mean white Gaussian noise with a variance σ_w^2, the CRLB becomes [28]:

  $$\mathrm{var}(\hat{\theta}) \geq \frac{\sigma_w^2}{\sum\limits_{n=0}^{N-1} \left(\frac{\partial s[n;\theta]}{\partial \theta}\right)^2} \tag{A.6}$$

- For the equivalent complex signal in complex Gaussian noise with the covariance matrix $\mathbf{\Sigma} = \frac{\sigma_w^2}{2}\boldsymbol{I}_2$ with $\boldsymbol{I}_2$ being the two-dimensional identity matrix, the CRLB can be calculated as:

$$\mathrm{var}(\hat{\theta}) \geq \frac{\sigma_w^2}{\Re\left\{\sum\limits_{n=0}^{N-1}\left(\frac{\partial s[n;\theta]}{\partial\theta}\right)^*\left(\frac{\partial s[n;\theta]}{\partial\theta}\right)\right\}} \tag{A.7}$$

- For a complex Gaussian probability density function of the form

$$p(\boldsymbol{x};\boldsymbol{\theta}) = \frac{1}{\pi^N|\Sigma(\boldsymbol{\theta})|}\exp\left(-(\boldsymbol{x}-\boldsymbol{s}(\boldsymbol{\theta}))^T\mathbf{\Sigma}^{-1}(\boldsymbol{\theta})(\boldsymbol{x}-\boldsymbol{s}(\boldsymbol{\theta}))\right) \tag{A.8}$$

where $\boldsymbol{\theta}$ is a parameter vector $\boldsymbol{\theta} = [\theta_1, ..., \theta_M]$, the calculation of the CRLB for the parameter θ_i is given as

$$\mathrm{var}(\theta_i) \geq \left[\boldsymbol{F}^{-1}\right]_{ii} \tag{A.9}$$

with $\boldsymbol{F}$ being the $M \times M$ Fisher information matrix. The $\{i, j\}$th entry of the matrix can be calculated as:

$$[\boldsymbol{F}(\boldsymbol{\theta})]_{ij} = \mathrm{Tr}\left[\mathbf{\Sigma}^{-1}\frac{\partial\mathbf{\Sigma}}{\partial\theta_i}\mathbf{\Sigma}^{-1}\frac{\partial\mathbf{\Sigma}}{\partial\theta_j}\right] + 2\Re\left\{\frac{\partial\boldsymbol{s}^H(\boldsymbol{\theta})}{\partial\theta_i}\mathbf{\Sigma}^{-1}\frac{\partial\boldsymbol{s}(\boldsymbol{\theta})}{\partial\theta_j}\right\}. \tag{A.10}$$

A.4 Least Squares Estimators

A least squares estimator is defined as

$$\hat{\theta}_{\mathrm{LS}} = \arg\min_{\theta}||\boldsymbol{x} - \boldsymbol{f}(\boldsymbol{s};\theta)||^2, \tag{A.11}$$

where $\boldsymbol{x}$ is the erroneous observation and $f(\mathrm{s};\theta)$ is the hypothetical noise-free observation for a chosen parameter θ. The LS estimator chooses the value that minimizes the sum of the squared errors. For a zero-mean Gaussian error with a diagonal covariance matrix and equal variances, the LS estimator becomes the ML estimator.

A.5 Residuum

The residuum is a measure of estimation quality. Assuming that $\tilde{\theta}$ was estimated using the observation $\boldsymbol{x}$, the residual is defined as

$$R_{\tilde{\theta}} = \left(\boldsymbol{x} - \boldsymbol{f}(\boldsymbol{s}; \tilde{\theta})\right)^2 . \qquad \text{(A.12)}$$

A small residual means that the estimated $\tilde{\theta}$ is close to the measured quantity.

B

Noise Variance of the ML Signal Model in Multipath

Suppose we have the two signals in the discrete Fourier transformation domain

$$Y_1[k] = S[k] + W_1[k], \quad k = 0, 1, ..., K-1 \tag{B.1}$$

$$Y_2[k] = \sum_{p=1}^{P} \alpha_p S[k] \mathrm{e}^{\frac{-\mathrm{j}2\pi\tau_p k}{K}} + W_2[k], \tag{B.2}$$

where W_1, W_2 are white and uncorrelated, following $W_i \sim \mathcal{CN}(0, \sigma^2_{W_i})$. $Y_2[k]$ can be rewritten as

$$Y_2[k] = \sum_{p=1}^{P} \alpha_p Y_1[k] \mathrm{e}^{\frac{-\mathrm{j}2\pi\Delta\tau_p k}{K}} + \tilde{W}[k], \tag{B.3}$$

where

$$\tilde{W}[k] = W_2[k] - \sum_{p=1}^{P} \alpha_p W_1[k] \mathrm{e}^{\frac{-\mathrm{j}2\pi\tau_p k}{K}}. \tag{B.4}$$

$\tilde{W}$ is a white Gaussian process iff it fulfills the following

$$\mathbb{E}\left\{\tilde{W}[k_1]\tilde{W}^*[k_2]\right\} = \begin{cases} \sigma^2_{\tilde{W}}[k_1] & \text{if } k_1 = k_2 \\ 0 & else \end{cases}. \tag{B.5}$$

Evaluating this term for the above model and, taking into account that $W_1[k]$ and $W_2[k]$ are uncorrelated, the remaining terms are

$$\mathbb{E}\left\{\tilde{W}[k_1], \tilde{W}^*[k_2]\right\} = \mathbb{E}\left\{W_2[k_1]W_2^*[k_2]\right\} \\ + \mathbb{E}\left\{W_1[k_1]W_1^*[k_2]\sum_{p=1}^{P}\alpha_p \mathrm{e}^{\frac{-\mathrm{j}2\pi\tau_p k_1}{K}}\sum_{l=1}^{P}\alpha_l^* \mathrm{e}^{\frac{\mathrm{j}2\pi\tau_l k_2}{K}}\right\}.$$

Evaluating each summand,

$$\mathbb{E}\left\{W_2[k_1]W_2^*[k_2]\right\} = \varphi_{22}(k_2 - k_1) \tag{B.6}$$

$$\mathbb{E}\left\{W_1[k_1]W_1^*[k_2]\sum_{p=1}^{P}\alpha_p \mathrm{e}^{\frac{-\mathrm{j}2\pi\tau_p k_1}{K}}\sum_{l=1}^{P}\alpha_l^* \mathrm{e}^{\frac{\mathrm{j}2\pi\tau_l k_2}{K}}\right\} = \\ = \varphi_{11}(k_2 - k_1)\sum_{p=1}^{P}\sum_{l=1}^{P}\alpha_p\alpha_l^* \mathrm{e}^{\frac{-\mathrm{j}2\pi(\tau_p k_1 - \tau_l k_2)}{K}}. \tag{B.7}$$

where $\varphi_{11}(k_2 - k_1), \varphi_{22}(k_2 - k_1)$ are the autocorrelations of W_1 and W_2. Due to their white nature, the autocorrelations of W_1 and W_2 are zero except for $k_1 = k_2$, where it takes the value of their variances. Hence, $\tilde{W}[k]$ is a white process having the variance $\sigma_{\tilde{W}}[k]^2 = \sigma_2^2 + \sigma_1^2 \left|\sum_{p=1}^{P}\alpha_p \mathrm{e}^{\frac{-\mathrm{j}2\pi\tau_p k}{K}}\right|^2$. Therefore, using this signal model, the SNR value of each frequency bin of the discrete Fourier transform of the signals varies as a function of the unknown parameters.

C

Entries of the FIM for the Multipath Signal Model

The entries of the FIM for a vector parameter in a complex Gaussian probability density function is given in (4.40). Using equations (4.43)-(4.49), and defining

$$A(k,i) = \sum_{p=1}^{P} \beta^{(p)} \mathrm{e}^{-\mathrm{j}\Omega_k (\Delta\tau^{(p)} - \Delta\tau^{(i)})} \tag{C.1}$$

$$\Omega_k = \frac{2\pi k}{K}, \tag{C.2}$$

the entries are given as

$$\begin{aligned}[\boldsymbol{F}(\boldsymbol{\theta})]_{(\Delta\tau^{(i)},\Delta\tau^{(j)})} = &\sum_{k=0}^{K-1} \frac{4\sigma_1^4}{\sigma_{\tilde{W}}^4[k]} \Omega_k^2 \Im\Big\{\beta^{(i)} A(k,i)\Big\} \Im\Big\{\beta^{(j)} A(k,j)\Big\} \\ &+ 2\Re\left\{\sum_{k=0}^{K-1} \frac{1}{\sigma_{\tilde{W}}^2[k]} \Omega_k^2 \beta^{(i)} \beta^{(j)} |S[k]|^2 \mathrm{e}^{-\mathrm{j}\Omega_k(\Delta\tau^{(j)} - \Delta\tau^{(i)})}\right\}\end{aligned} \tag{C.3}$$

$$[\boldsymbol{F}(\boldsymbol{\theta})]_{(\beta^{(i)},\beta^{(j)})} = \sum_{k=0}^{K-1} \frac{4\sigma_1^4}{\sigma_{\tilde{W}}^4[k]} \Re\{A(k,i)\} \Re\{A(k,j)\}$$

$$+ 2\Re\left\{\sum_{k=0}^{K-1} \frac{1}{\sigma_{\tilde{W}}^2[k]} |S[k]|^2 \mathrm{e}^{-\mathrm{j}\Omega_k(\Delta\tau^{(j)} - \Delta\tau^{(i)})}\right\} \tag{C.4}$$

$$\begin{aligned}[\boldsymbol{F}(\boldsymbol{\theta})]_{(\Delta\tau^{(i)},\beta^{(j)})} &= \sum_{k=0}^{K-1} \frac{-4\pi\sigma_1^4}{\sigma_{\tilde{W}}^4[k]} \Re\{A(k,i)\}\Im\{A(k,j)\} \\ &\quad + 2\Re\left\{\sum_{k=0}^{K-1} \frac{1}{\sigma_{\tilde{W}}^2[k]} (-\mathrm{j}\Omega_k)\beta^{(j)} |S[k]|^2 \mathrm{e}^{-\mathrm{j}\Omega_j(\Delta\tau^{(j)} - \Delta\tau^{(i)})}\right\}\end{aligned} \tag{C.5}$$

$$\begin{aligned}[\boldsymbol{F}(\boldsymbol{\theta})]_{(\beta^{(i)},\Delta\tau^{(j)})} &= \sum_{k=0}^{K-1} \frac{-4\pi\sigma_1^4}{\sigma_{\tilde{W}}^4[k]} \Re\{A(k,j)\}\Im\{A(k,i)\} \\ &\quad + 2\Re\left\{\sum_{k=0}^{K-1} \frac{1}{\sigma_{\tilde{W}}^2[k]} \Omega_k\beta^{(i)} |S[k]|^2 \mathrm{e}^{-\mathrm{j}\Omega_j(\Delta\tau^{(j)} - \Delta\tau^{i})}\right\}\end{aligned} \tag{C.6}$$

$\mathcal{D}$

TDoA Error in COST-Channels

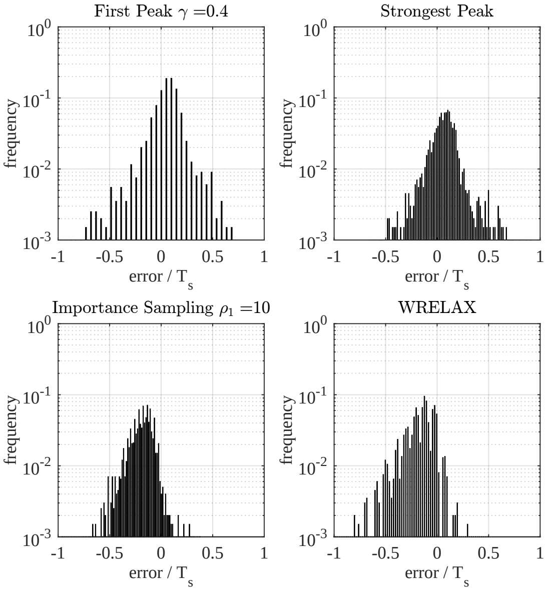


Figure D.1: Histograms for the Cost 207 rural area model and SNR = 5 dB

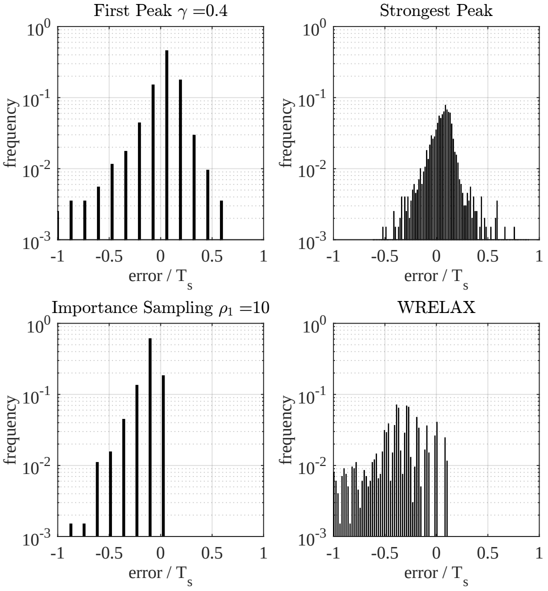


Figure D.2: Histograms for the Cost 207 rural area model and SNR = 25 dB

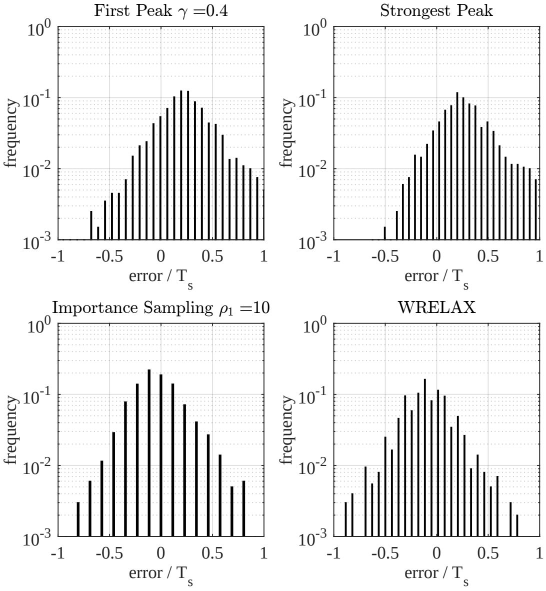


Figure D.3: Histograms for the Cost 207 typical urban model and SNR = 5 dB

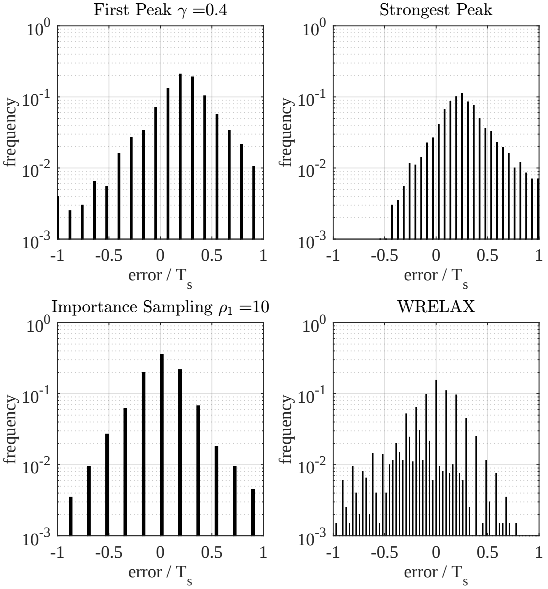


Figure D.4: Histograms for the Cost 207 typical urban model and SNR = 25 dB

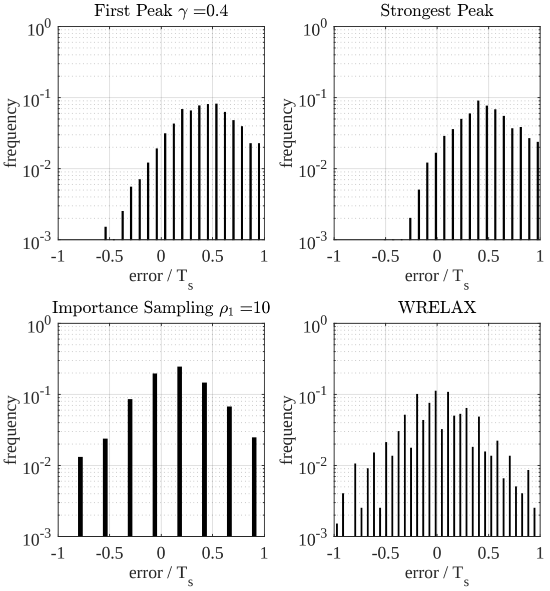


Figure D.5: Histograms for the Cost 207 bad urban model and SNR = 5 dB

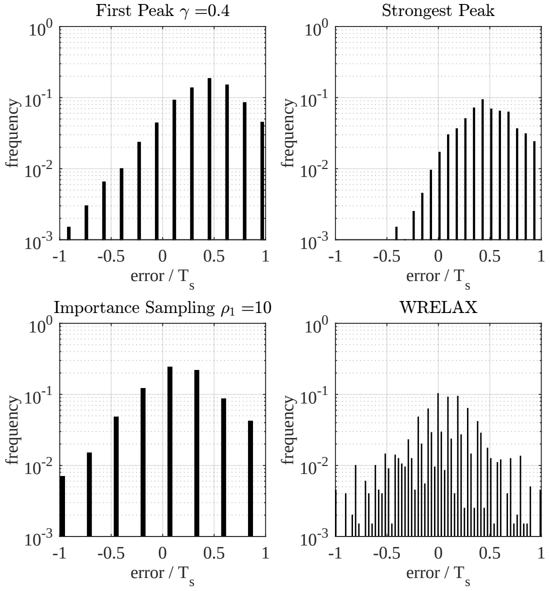


Figure D.6: Histograms for the Cost 207 bad urban model and SNR = 25 dB

$\mathcal{E}$

IS Algorithm Parameters

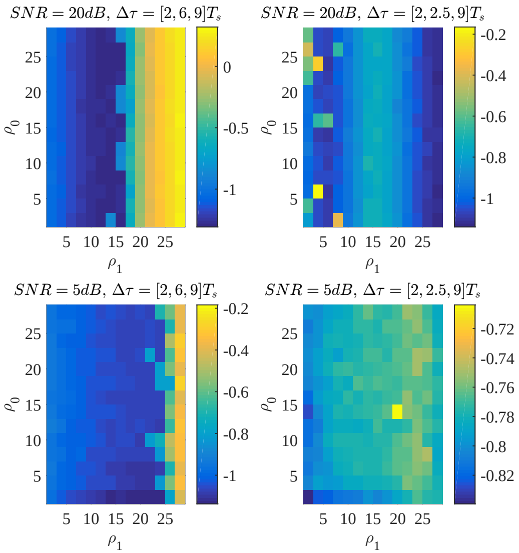


Figure E.1: RMSE (in dB) for estimating $\Delta\tau^{(1)}$ using different algorithm parameters

Acronyms

AoA	Angle of Arrival
AWGN	Additive White Gaussian Noise
BT	Bandwidth Time
CDF	Cumulative Distribution Function
CRLB	Cramér Rao Lower Bound
DFT	Discrete Fourier Transform
ECEF	Earth-Centered Earth-Fixed
EKF	Extended Kalman Filter
EM	Expectation Maximization
FIM	Fisher Information Matrix
FDoA	Frequency Difference of Arrival
FT	Fourier Transform
GCC	Generalized Cross-Correlation
GDoP	Geometric Dilution of Precision
HMM	Hidden Markov Model
IS	Importance Sampling

LoS	Line-of-Sight
LS	Least Squares
MAP	Maximum a Posteriori
MDL	Minimum Description Length
ML	Maximum Likelihood
MMSE	Minimum Mean Square Error
MSE	Mean Square Error
NLoS	Non-Line-of-Sight
RD	Range Difference
RMSE	Root Mean Square Error
RSS	Received Signal Strength
RSSD	Received Signal Strength Difference
SNR	Signal-to-Noise Ratio
TDoA	Time Difference of Arrival
ToA	Time of Arrival
TSE	Taylor Series Estimation
UKF	Unscented Kalman Filter
USRP	Universal Software Radio Peripheral
WGS 84	World Geodetic System 1984
WSS	Wide Sense Stationary
WRELAX	Weighted Fourier Transform and Relaxation
ZZB	Ziv-Zakai Bound

Notations and Symbols

Notations

$(\cdot)^*$	Conjugate complex of the argument
$\hat{(\cdot)}$	Estimate of the argument
$\Re\{.\}$	Real part of a complex variable
$\Im\{.\}$	Imaginary part of a complex variable
$\boldsymbol{I}_d$	Identity matrix with dimension $d \times d$
$\lvert . \rvert$	Scalar: magnitude
	Matrix: determinant
$\lvert\lvert . \rvert\rvert$	Matrix or vector norm
$\mathrm{Tr}(\cdot)$	Trace of a matrix
$(\cdot)^T$	Vector or matrix transpose
$(\cdot)^H$	Vector or matrix hermitian
$\mathrm{Si}(x)$	Sinus cardinalis
$\lfloor \cdot \rfloor$	Round down to largest integer number less or equal

Symbols

$\alpha_i^{(p)}$	Gain of the pth multipath component of sensor i
A	Kalman filter transition matrix

$\beta_i^{(p)}$	Gain ratio between the pth multipath component of sensor i and the reference sensor, assuming only one path at the reference sensor
$\beta_{i,j}^{(k,p)}$	Gain ratio between the kth multipath component of sensor i and the pth multipath component of sensor j
$b_{i,j}$	Bias of $\widehat{\Delta d}_{i,j}$
B	Signal bandwidth
χ_i	Markov state probability
c	speed of light
$C_{i,j}(\tau)$	Continuous cross-correlation between the signals of sensor i and sensor j
$C_{i,j}^D[m]$	Discrete cross-correlation between the signals of sensor i and sensor j
$C(\boldsymbol{\theta})$	Value of the cost function using the parameter vector $\boldsymbol{\theta}$
$\boldsymbol{C}$	Covariance matrix of the estimated range differences
$\boldsymbol{C}_i$	Covariance matrix of the estimated range differences, using sensor i as reference
$\boldsymbol{\Delta}$	State relative shift vector per Kalman filter step
$\Delta d_{i,j}$	Range difference between sensor i and sensor j
$\Delta d_{i,j}(\boldsymbol{x})$	Position dependent range difference resulting from the first path distances
$\Delta\tau_{i,j}$	Time difference of arrival between sensor i and sensor j
$\Delta\tau_{i,j}^{(k,p)}$	Delay difference between the kth multipath component of sensor i and the pth multipath component of sensor j
d_i	Distance between sensor i and the transmitter
$D(\boldsymbol{\theta})$	Cost function using the parameter vector $\boldsymbol{\theta}$
D_m	Cost function at the mth iteration
dim	Position dimension
$\eta_{i,j}$	Additive noise term on the estimated TDoA
$e_{i,j}$	Bias of $\widehat{\Delta\tau}_{i,j}$
$f_i(\boldsymbol{p_j}, \boldsymbol{x})$	TDoA equation for sensor j using sensor i as reference
$\boldsymbol{f}_i(\boldsymbol{p}, \boldsymbol{x})$	TDoA equation system using sensor i as reference
f_s	Sampling frequency
$\boldsymbol{F}(\boldsymbol{\theta})$	Fisher information matrix of parameter vector $\boldsymbol{\theta}$
γ	Threshold for bias identification using Cong's algorithm
g_i	Weight on the ith Sigma-point
K	Number of received samples
$\boldsymbol{K}[n]$	Kalman gain at step number n
λ	Longitude

μ_b	Mean range difference bias
$\boldsymbol{m}_i$	Sigma-points of the unscented Kalman filter
N	Number of sensors
ν_i	Forward Markov state probability
$\boldsymbol{o}$	Hidden Markov model observation
Ω_i	Received signal strength at sensor i
$\Omega_{i,j}$	Received signal strength difference between sensor i and sensor j
Φ	Latitude
Φ_i	Bearing angle of sensor i
$\boldsymbol{\Phi}_A(\boldsymbol{\tau})$	DFT of the delayed and summed multipath signal in the active scenario
$\boldsymbol{\Phi}_P(\boldsymbol{\tau})$	DFT of the delayed and summed multipath signal in the passive scenario
$\boldsymbol{p}_i$	Known coordinates of the position of sensor i
P_i	Number of received multipath components at sensor i
$\boldsymbol{P}$	Kalman/Markov state covariance matrix
ρ_0,ρ_1	Tuning parameters for the importance sampling algorithm
$r_i(t)$	Continuous receive signal at sensor i
$r_i[n]$	Sampled receive signal at sensor i
$R_i[k]$	Discrete Fourier transform of $r_i[n]$
$\sigma^2_{m_i}$	Variance of the parameter m of the ith sensor
$\boldsymbol{\Sigma}_i$	Covariance matrix of $R_i[k]$ in the active scenario
$\tilde{\boldsymbol{\Sigma}}_i$	Covariance matrix of $R_i[k]$ in the passive scenario
$s(t)$	Continuous transmit signal
S_{TX}	Transmit signal power
$\hat{S}_i$	Estimated receive signal power at sensor i
SN_i	Sensor i
τ_i	Delay between transmit and receive time due to the covered distance
$\tau_i^{(p)}$	Delay of the pth multipath component of sensor i
t_0	Transmit time
T_o	Observation window length
T_s	Symbol length
T_{samp}	Sampling interval
TX	Transmitter
$\boldsymbol{\theta}$	Parameter vector
$w_i(t)$	Additive noise term on received signal of sensor i
$\boldsymbol{x}$	Unknown transmitter coordinates

ξ_i	Backward Markov state probability
$\xi_{i,j}$	Additive noise term on the estimated range differences

Bibliography

[1] David Munoz, Frantz Bouchereau Lara, Cesar Vargas, and Rogerio Enriquez-Caldera, *Position Location Techniques and Applications*, Academic Press, 2009.

[2] Reza Zekavat and R. Michael Buehrer, *Handbook of Position Location: Theory, Practice and Advances*, Wiley-IEEE Press, 1st edition, 2011.

[3] J. Kennedy and M.C. Sullivan, "Direction Finding and Smart Antennas Using Software Radio Architectures," *IEEE Communications Magazine*, vol. 33, no. 5, pp. 62–68, May 1995.

[4] M. Vossiek, L. Wiebking, P. Gulden, J. Weighardt, and C. Hoffmann, "Wireless Local Positioning - Concepts, Solutions, Applications," in *Proceedings of Radio and Wireless Conference (RAWCON)*, Aug 2003, pp. 219–224.

[5] Louis A. Stilp, "Time Difference of Arrival Technology for Locating Narrowband Cellular Signals," 1996, vol. 2602, pp. 134–144.

[6] Sichun Wang, R. Inkol, and B.R. Jackson, "Relationship Between the Maximum Likelihood Emitter Location Estimators Based on Received Signal Strength (RSS) and Received Signal Strength Difference (RSSD),"

in *26th Biennial Symposium on Communications (QBSC)*, May 2012, pp. 64–69.

[7] P.C. Chestnut, “Emitter Location Accuracy Using TDOA and Differential Doppler,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-18, no. 2, pp. 214–218, Mar 1982.

[8] S. Stein, “Differential Delay/Doppler ML Estimation With Unknown Signals,” *IEEE Transactions on Signal Processing*, vol. 41, no. 8, pp. 2717–2719, Aug 1993.

[9] C. Knapp and G.Clifford Carter, “The Generalized Correlation Method for Estimation of Time Delay,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 24, no. 4, pp. 320–327, Aug 1976.

[10] Joseph C. Hassab and R. Boucher, “Optimum Estimation of Time Delay by a Generalized Correlator,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 27, no. 4, pp. 373–380, Aug 1979.

[11] G. Jacovitti and G. Scarano, “Discrete Time Techniques for Time Delay Estimation,” *IEEE Transactions on Signal Processing*, vol. 41, no. 2, pp. 525–533, Feb 1993.

[12] R. Boucher and Joseph C. Hassab, “Analysis of Discrete Implementation of Generalized Cross Correlator,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 29, no. 3, pp. 609–611, Jun 1981.

[13] R. Moddemeijer, “On the Determination of the Position of Extrema of Sampled Correlators,” *IEEE Transactions on Signal Processing*, vol. 39, no. 1, pp. 216–219, Jan 1991.

[14] B.M. Sadler and R.J. Kozick, “A Survey of Time Delay Estimation Performance Bounds,” in *Fourth IEEE Workshop on Sensor Array and Multichannel Processing*, July 2006, pp. 282–288.

[15] Harry L. Van Trees, *Detection, Estimation, and Modulation Theory, Part I*, Wiley, 1968.

[16] K.L. Bell, Y. Steinberg, Y. Ephraim, and H.L. Van Trees, “Extended Ziv-Zakai Lower Bound for Vector Parameter Estimation,” *IEEE Transactions on Information Theory*, vol. 43, no. 2, pp. 624–637, Mar 1997.

[17] A. Zeira and P.M. Schultheiss, “Realizable Lower Bounds for Time Delay Estimation. 2. Threshold Phenomena,” *IEEE Transactions on Signal Processing*, vol. 42, no. 5, pp. 1001–1007, May 1994.

[18] A.J. Weiss and E. Weinstein, “Fundamental Limitations in Passive Time Delay Estimation–Part I: Narrow-Band Systems,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 31, no. 2, pp. 472–486, Apr 1983.

[19] E. Weinstein and A.J. Weiss, “Fundamental Limitations in Passive Time-Delay Estimation–Part II: Wide-Band Systems,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 32, no. 5, pp. 1064–1078, Oct 1984.

[20] G.Clifford Carter, “Coherence and Time Delay Estimation,” *Proceedings of the IEEE*, vol. 75, no. 2, pp. 236–255, Feb 1987.

[21] W.H. Foy, “Position-Location Solutions by Taylor-Series Estimation,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-12, no. 2, pp. 187–194, Mar 1976.

[22] D.J. Torrieri, “Statistical Theory of Passive Location Systems,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-20, no. 2, pp. 183–198, Mar 1984.

[23] Benjamin Friedlander, “A Passive Localization Algorithm and its Accuracy Analysis,” *IEEE Journal of Oceanic Engineering*, vol. 12, no. 1, pp. 234–245, January 1987.

[24] J.O. Smith and J.S. Abel, “Closed-Form Least-Squares Source Location Estimation From Range-Difference Measurements,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 35, no. 12, pp. 1661–1669, Dec 1987.

[25] 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*, vol. 35, no. 8, pp. 1223–1225, Aug 1987.

[26] B.T. Fang, “Simple Solutions for Hyperbolic and Related Position Fixes,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 26, no. 5, pp. 748–753, Sep 1990.

[27] Y.T. Chan and K.C. Ho, “A Simple and Efficient Estimator for Hyperbolic Location,” *IEEE Transactions on Signal Processing*, vol. 42, no. 8, pp. 1905–1915, Aug 1994.

[28] Steven M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.

[29] K. Kroschel, *Statistische Informationstechnik: Signal- und Mustererkennung, Parameter- und Signalschtzung*, Springer, 2004.

[30] Angus P. Andrews Mohinder S. Grewal, *Kalman Filtering: Theory and Practice Using MATLAB*, Wiley, 2001.

[31] S. Julier, J. Uhlmann, and H.F. Durrant-Whyte, "A New Method for the Nonlinear Transformation of Means and Covariances in Filters and Estimators," *IEEE Transactions on Automatic Control*, vol. 45, no. 3, pp. 477–482, Mar 2000.

[32] S.J. Julier and J.K. Uhlmann, "Unscented Filtering and Nonlinear Estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, Mar 2004.

[33] N. El Gemayel, H. Jäkel, and F.K. Jondral, "A Hybrid TDOA/RSSD Geolocation System Using the Unscented Kalman Filter," in *IEEE 78th Vehicular Technology Conference (VTC Fall)*, Sep 2013, pp. 1–5.

[34] A. Catovic and Z. Sahinoglu, "The Cramer-Rao bounds of Hybrid TOA/RSS and TDOA/RSS Location Estimation Schemes," *IEEE Communications Letters*, vol. 8, no. 10, pp. 626–628, Oct 2004.

[35] B. Yang and J. Scheuing, "Cramer-Rao Bound and Optimum Sensor Array for Source Localization from Time Differences of Arrival," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Mar 2005, vol. 4, pp. iv/961–iv/964 Vol. 4.

[36] Bin Yang and J. Scheuing, "A Theoretical Analysis of 2D Sensor Arrays for TDOA Based Localization," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2006, vol. 4, pp. IV–IV.

[37] K.C. Ho, "Bias Reduction for an Explicit Solution of Source Localization Using TDOA," *IEEE Transactions on Signal Processing*, vol. 60, no. 5, pp. 2101–2114, May 2012.

[38] F. Jondral, *Nachrichtensysteme: Grundlagen, Verfahren, Anwendungen*, Schlembach, 2011.

[39] M.D. Hahm, Z.I. Mitrovski, and Edward L. Titlebaum, "Deconvolution in the Presence of Doppler with Application to Specular Multipath Parameter Estimation," *IEEE Transactions on Signal Processing*, vol. 45, no. 9, pp. 2203–2219, Sep 1997.

[40] P. Stoica and R.L. Moses, *Spectral Analysis of Signals*, Pearson Prentice Hall, 2005.

[41] Petre Stoica and Arye Nehorai, "MUSIC, Maximum Likelihood, and Cramer-Rao Bound: Further Results and Comparisons," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 38, no. 12, pp. 2140–2150, Dec 1990.

[42] A.-J. Van der Veen, M.C. Vanderveen, and A.J. Paulraj, "Joint Angle and Delay estimation Using Shift-Invariance Properties," *IEEE Signal Processing Letters*, vol. 4, no. 5, pp. 142–145, May 1997.

[43] H. Saarnisaari, "ML Time Delay Estimation in a Multipath Channel," in *IEEE 4th International Symposium on Spread Spectrum Techniques and Applications Proceedings*, Sep 1996, vol. 3, pp. 1007–1011 vol.3.

[44] C. Steffes and S. Rau, "Multipath Detection in TDOA Localization Scenarios," in *Workshop on Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, Sept 2012, pp. 88–92.

[45] J. Scheuing and Bin Yang, "Disambiguation of TDOA Estimates in Multi-Path Multi-Source Environments (DATEMM)," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2006, vol. 4, pp. IV–IV.

[46] A. F. M. Smith and A. E. Gelfand, "Bayesian Statistics without Tears: A Sampling-Resampling Perspective," *The American Statistician*, vol. 46, no. 2, pp. pp. 84–88, 1992.

[47] Samuel H. Brooks, "A Discussion of Random Methods for Seeking Maxima," *Operations Research*, vol. 6, no. 2, pp. pp. 244–251, 1958.

[48] M. J. Appel, R. LaBarre, and D. Radulovic, "On Accelerated Random Search," *SIAM Journal on Optimization*, vol. 14, no. 3, pp. 708–731, 2004.

[49] A. Masmoudi, F. Bellili, S. Affes, and A. Stephenne, "A Non-Data-Aided Maximum Likelihood Time Delay Estimator Using Importance Sampling," *IEEE Transactions on Signal Processing*, vol. 59, no. 10, pp. 4505–4515, Oct 2011.

[50] M. Feder and E. Weinstein, "Parameter Estimation of Superimposed Signals Using the EM Algorithm," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 36, no. 4, pp. 477–489, Apr 1988.

[51] K.C. Sharman, "Maximum Likelihood Parameter Estimation by Simulated Annealing," in *International Conference on Acoustics, Speech, and Signal Processing*, Apr 1988, pp. 2741–2744.

[52] Jian Li and Renbiao Wu, "An Efficient Algorithm for Time Delay Estimation," *IEEE Transactions on Signal Processing*, vol. 46, no. 8, pp. 2231–2235, Aug 1998.

[53] Huigang Wang, S. Kay, and S. Saha, "An Importance Sampling Maximum Likelihood Direction of Arrival Estimator," *IEEE Transactions on Signal Processing*, vol. 56, no. 10, pp. 5082–5092, Oct 2008.

[54] A. Masmoudi, F. Bellili, S. Affes, and A. Stephenne, "A Maximum Likelihood Time Delay Estimator in a Multipath Environment Using Importance Sampling," *IEEE Transactions on Signal Processing*, vol. 61, no. 1, pp. 182–193, Jan 2013.

[55] Martin Pincus, "Letter to the Editor-A Closed Form Solution of Certain Programming Problems," *Operations Research*, vol. 16, no. 3, pp. 690–694, 1968.

[56] J. Rissanen, "Modeling by Shortest Data Description," *Automatica*, vol. 14, no. 5, pp. 465 – 471, 1978.

[57] M. Wax and T. Kailath, "Detection of Signals by Information Theoretic Criteria," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 33, no. 2, pp. 387–392, Apr 1985.

[58] D.C. Rife and R.R. Boorstyn, "Multiple Tone Parameter Estimation From Discrete-Time Observations," *The Bell System Technical Journal*, vol. 55, no. 9, pp. 1389–1410, Nov 1976.

[59] F.C. Schweppe, "On the Angular Resolution of Multiple Targets," *Proceedings of the IEEE*, vol. 52, no. 9, pp. 1044–1045, Sep 1964.

[60] Cost 207, "Digital Land Mobile Radio Communications," Office for Official Publications of the European Communities, 1989.

[61] J. Riba and A. Urruela, "A Non-Line-of-Sight Mitigation Technique Based on ML-Detection," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '04)*, May 2004, vol. 2, pp. ii–153–6 vol.2.

[62] N. El Gemayel, J. Schloemann, R. Buehrer, and F. Jondral, "Improved Indoor Positioning Using the Baum-Welch Algorithm," in *Global Communications Conference Workshops*, Sept 2015, pp. 1–5.

[63] A. Conti, M. Guerra, D. Dardari, N. Decarli, and M.Z. Win, "Network Experimentation for Cooperative Localization," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 2, pp. 467–475, February 2012.

[64] S. Marano, W.M. Gifford, H. Wymeersch, and M.Z. Win, "NLOS Identification and Mitigation for Localization Based on UWB Experimental Data," *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 7, pp. 1026–1035, September 2010.

[65] I. Guvenc and Chia-Chin Chong, "A Survey on TOA Based Wireless Localization and NLOS Mitigation Techniques," *IEEE Communications Surveys Tutorials*, vol. 11, no. 3, pp. 107–124, rd 2009.

[66] Z. Sahinoglu S. Gezici, "UWB Geolocation Techniques for IEEE 802.15.4a Personal Area Networks," *MERL Technical report, Cambrige, USA*, 2004.

[67] K.W.K. Lui, H.C. So, and W.-K. Ma, "Maximum A Posteriori Approach to Time-of-Arrival-Based Localization in Non-Line-of-Sight Environment," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 3, pp. 1517–1523, Mar 2010.

[68] S. Venkatesh and R.Michael Buehrer, "NLOS Mitigation Using Linear Programming in Ultrawideband Location-Aware Networks," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 5, pp. 3182–3198, Sept 2007.

[69] R.M. Vaghefi, J. Schloemann, and R.M. Buehrer, "NLOS Mitigation in TOA-Based Localization Using Semidefinite Programming," in *10th Workshop on Positioning Navigation and Communication (WPNC)*, Mar 2013, pp. 1–6.

[70] Sung-Shik Woo, Heung-Ryeol You, and Jong-Seog Koh, "The NLOS Mitigation Technique for Position Location Using IS-95 CDMA Networks," in *52nd IEEE Vehicular Technology Conference*, 2000, vol. 6, pp. 2556–2560 vol.6.

[71] M.I. Silventoinen and T. Rantalainen, "Mobile Station Emergency Locating in GSM," in *IEEE International Conference on Personal Wireless Communications*, Feb 1996, pp. 232–238.

[72] Pi-Chun Chen, “A Non-Line-of-Sight Error Mitigation Algorithm in Location Estimation,” in *IEEE Wireless Communications and Networking Conference (WCNC)*, 1999, pp. 316–320 vol.1.

[73] N. El Gemayel, S. Meier, and F.K. Jondral, “On the Applicability of the Residual Weighting Algorithm for TDOA,” in *4th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, Oct 2012, pp. 143–147.

[74] Li Cong and Weihua Zhuang, “Nonline-of-Sight Error Mitigation in Mobile Location,” *IEEE Transactions on Wireless Communications*, vol. 4, no. 2, pp. 560–573, Mar 2005.

[75] Li Cong and Weihua Zhuang, “Non-Line-of-Sight Error Mitigation in TDOA Mobile Location,” in *IEEE Global Telecommunications Conference (GLOBECOM)*, 2001, vol. 1, pp. 680–684 vol.1.

[76] L. Rabiner and B.H. Juang, “An Introduction to Hidden Markov Models,” *IEEE ASSP Magazine*, vol. 3, no. 1, pp. 4–16, Jan 1986.

[77] George Soules Norman Weiss Leonard E. Baum, Ted Petrie, “A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains,” *The Annals of Mathematical Statistics*, vol. 41, no. 1, pp. 164–171, 1970.

[78] L. Rabiner, “A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition,” *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, Feb 1989.

[79] N. El Gemayel, H. Jäkel, and F.K. Jondral, “A Weighted Least Squares Algorithm for Passive Localization in Multipath Scenarios,” in *IEEE 82nd Vehicular Technology Conference (VTC Fall)*, 2015.

[80] A.J. Weiss, “Direct Position Determination of Narrowband Radio Frequency Transmitters,” *IEEE Signal Processing Letters*, vol. 11, no. 5, pp. 513–516, May 2004.

[81] N. Vankayalapati, S. Kay, and Quan Ding, “TDOA Based Direct Positioning Maximum Likelihood Estimator and the Cramer-Rao Bound,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 50, no. 3, pp. 1616–1635, Jul 2014.

[82] Yihong Qi, Hisashi Kobayashi, and H. Suda, “Analysis of Wireless Geolocation in a Non-Line-of-Sight Environment,” *IEEE Transactions on Wireless Communications*, vol. 5, no. 3, pp. 672–681, Mar 2006.

[83] N. El Gemayel, S. Koslowski, F.K. Jondral, and J. Tschan, “A Low Cost TDOA Localization System: Setup, Challenges and Results,” in *10th Workshop on Positioning Navigation and Communication (WPNC)*, Mar 2013, pp. 1–4.

[84] N. El Gemayel, H. Jäkel, and F.K. Jondral, “Error Analysis of a Low Cost TDoA Sensor Network,” in *IEEE/ION Position, Location and Navigation Symposium (PLANS)*, May 2014, pp. 1040–1045.

[85] Michael Sebastian Mühlhaus, *Automatische Modulationsartenerkennung in MIMO-Systemen*, Ph.D. thesis, Karlsruhe Institute of Technology, 2014.

[86] Ettus Research, “USRP2: The Next Generation,” `http://www.ettus.com`, Online. Accessed: 15.01.2016.

[87] Ettus Research, “WBX 50-2200 MHz Product Information,” `http://www.ettus.com/product/details/WBX`, Online. Accessed: 15.01.2016.

[88] Trimble Navigation LTD, “Trimble Mini-T GPS Disciplined Clock Board,” `http://www.trimble.com/timing/mini-t.aspx`, Online. Accessed: 15.01.2016.

[89] “Garmin eTrex Owner’s Manual,” `http://static.garmincdn.com/pumac/eTrex_10-20-30_OM_EN.pdf`, Online. Accessed: 15.01.2016.

[90] Harald Cramér, *Mathematical Methods of Statistics*, vol. 9, Princeton university press, 1999.

Sponsorship

Some parts in this thesis are based on the research visit to Wireless@VT Group in Virginia Tech, Virginia. I would like to thank the Karlsruhe House of Young Scientists (KHYS) as well as the Communications Engineering Lab (CEL) for supporting this visit.

Index