Schuh Maximilian, BSc

Accurate UWB-based Localization using Online Calibrated and Corrected Single-Sided Two-Way Ranging

MASTER'S THESIS

to achieve the university degree of

Diplom-Ingenieur

Master's degree programme: Computer Science

submitted to

Graz University of Technology

Supervisor

Dipl.-Ing. Bernhard Großwindhager
Ass. Prof. Dr.techn. Carlo Alberto Boano

Institute of Technical Informatics
Head: Univ.-Prof. Dipl-Inform. Dr.sc.ETH Kay Uwe Römer

Graz, May 2019

AFFIDAVIT

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master‘s thesis.

Date

Signature

Kurzfassung

Positionsdaten sind ein wichtiger Bestandteil vieler von uns täglich genutzten elektronischen Geräte. Während Satellitennavigationssysteme wie Galileo, GLONASS oder GPS Lokalisierung im Freien ermöglichen, gibt es für die Lokalisierung in Gebäuden noch keine präzise, universelle Lösung. Speziell das Internet der Dinge (IoT) und die damit einhergehende wachsende Anzahl vernetzter mobiler Geräte benötigt präzise und genaue Lokalisierung. Mögliche Anwendungsgebiete finden sich unter anderem in der Lagerlogistik, in der Lokalisierung von Robotern und Drohnen, oder auch im Gesundheits- und Pflegewesen. Die Standardisierung von Ultrabreitband (UWB) innerhalb des IEEE 802.15.4 Standards erlaubt präzise, funkbasierte Distanzmessungen, die die Entwicklung von präzisen Lokalisierungssystemen ermöglicht.

Während *time-of-arrival* (ToA) beziehungsweise *time-difference-of-arrival* (TDoA) einen hohen Synchronisierungsaufwand haben, sind *two-way ranging* (TWR) Techniken weniger aufwändig und dennoch präzise. Insbesondere das so genannte *double-sided two-way ranging* (DS-TWR) erlaubt zentimetergenaue Distanzmessungen. Diese Genauigkeit wird durch den Austausch von mindestens drei Nachrichten zwischen zwei Geräten erreicht, wodurch die Skalierbarkeit und Effizienz dieser Technik limitiert ist. Die Verwendung von single-sided two-way ranging (SS-TWR) hingegen reduziert die Anzahl an Nachrichten, allerdings auf Kosten eine geringeren Präzision.

In dieser Arbeit wird die Genauigkeit und Präzision von SS-TWR mit herkömmlichen UWB Empfängern untersucht. Dazu wurden die größten Fehlerquellen für SS-TWR experimentell analysiert. Insbesondere die Antennenverzögerung, die Signalstärke und die Taktdifferenz unter dem Einfluss der Antwortzeit und der Taktkalibrierung werden betrachtet. Außerdem präsentiert diese Arbeit Kalibrier- und Korrekturtechniken, die zentimetergenaue SS-TWR Distanzmessungen ermöglichen. Darauf basierend wird ein Echtzeit Lokalisierungssystem vorgestellt, welches den geringen Aufwand von SS-TWR mit der hohen Präzision der vorgestellten Methoden kombiniert. Das gesamte System wurde auf einer Mikrokontroller-basierten Plattform implementiert. Zur Positionsberechnung wurden zwei effiziente Algorithmen implementiert, der MinMax Algorithmus und eine linearisierte Lösung der kleinsten Fehlerquadrate. Diese Algorithmen sind effizient genug, um direkt auf dem Mikroprozessor ausgeführt zu werden. Die Auswertung zeigt, dass das implementierte System eine 3D-Lokalisierung mit einer durchschnittlichen Genauigkeit von 30 cm erreicht. 90 % der Positionsfehler sind unter 50 cm, bei einer Wiederholrate von bis zu 50 Hz.

Abstract

Location awareness has become a crucial requirement for a lot of electronic devices we are using on a daily basis. While global satellite navigation systems such as Galileo, GLONASS, or GPS enable outdoor localization with an accuracy of several metres, there is still no universal solution enabling accurate indoor localization. Especially the expanding market following the rise of the Internet of Things (IoT) demands precise and accurate indoor localization on mobile devices. Applications can be found for example in the field of warehouse management, drone and robot tracking, as well as assisted living and personalized healthcare. In this context, the standardization of ultra-wideband (UWB) technology within the IEEE 802.15.4 standard enables accurate ranging and fostered the development of accurate localization systems.

While *time-of-arrival* (ToA) or *time-difference-of-arrival* (TDoA) schemes require a high synchronization effort, the *two-way ranging* (TWR) scheme is much simpler and nonetheless quite accurate. Especially the so called *double-sided two-way ranging* (DS-TWR) scheme obtains cm-level accuracy and precision. However, this precision comes at the cost of a three-way handshake between a pair of devices, which hinders the scalability and efficiency of this technique. Removing the third message in the handshake and hence performing a single-sided two-way ranging (SS-TWR) would increase both the efficiency and scalability of a localization system. However, SS-TWR is known to obtain less precise rangings.

This thesis first analyses experimentally the accuracy and precision of the SS-TWR scheme on off-the-shelf UWB radios. To this end, the impact of the major influences of SS-TWR have been analysed. In particular, the focus has been laid on the *antenna delay*, the *received signal strength* and the *clock speed offset* including the *response delay*. As a result of this investigation, we show that calibration and correction techniques allow a precision of 4 cm for SS-TWR. We then leverage our findings and design a SS-TWR-based real-time location system combining the efficiency of SS-TWR with the high precision ranging capabilities. The whole system has been implemented on an embedded platform. In particular, two computationally-lightweight algorithms – a bounding-box and a linearized least squares lateration algorithm – have been employed to estimate the positions directly on the embedded device. An experimental evaluation shows that the implemented system can achieve a mean 3D-localization accuracy of 30 cm with 90 % of the position errors below 50 cm and a position update rate of up to 50 Hz.

Acknowledgments

This thesis has been conducted at the Institute of Technical Informatics at Graz University of Technology.

First, I want to thank my supervisors Bernhard Großwindhager and Carlo Alberto Boano, for their help, advices, and support during my work on this thesis.

Further, I want to thank my family and friends, especially my parents and my sister, for their continuous support, motivation and patience. Without their help I would not have gotten this far.

Graz, May 2019 Maximilian Schuh

Contents

List of Tables

List of Figures

Abbreviations

BPM-BPSK	Burst Position Modulation-Binary Phase-Shift Keying
CCA	Clear Channel Assessment
CDF	Cumulative Distribution Function
CRI	Carrier Recovery Integrator
CSMA	Carrier Sense Multiple Access
DS-TWR	Double Sided TWR
DLL	Delay-Locked Loop
EWMA	Exponentially Weighted Moving Average
FEC	Forward Error Correction
FPU	Floating Point Unit
IoT	Internet of Things
LLS	Linear Least Squares
MAC	Medium Access Control
MAE	Mean Absolute Error
MCU	Microcontroller Unit
NLLS	Non-Linear Least Squares
NLOS	Non-Line-of-Sight
PDF	Probability Density Function
PHR	PHY header (physical layer header)
PLL	Phase-Locked Loop
PPDU	PHY protocol data unit
PRF	Pulse Repetition Frequency
RSME	Root Mean Squar Error
RTLS	Real-Time Location System
RTT	Receiver Time Tracking
SFD	Start of Frame Delimiter
SHR	Synchronization Header
SS-TWR	Single-Sided TWR
TDMA	Time-Division Multiple Access
TWR	Two-Way Ranging
UWB	Ultra-Wideband

1 Introduction

Accurate position data is a crucial part of numerous applications in our everyday life. Since the rise of global satellite navigation (e.g., Galileo or GPS), localization is a straightforward task, finding many applications in our everyday life. However, these solutions cannot be used in indoor environments because of unreliable signal quality and the insufficient accuracy. Therefore, there is an urgent need for an indoor localization solution. In this field so called Real-Time Location Systems (RTLS) emerged. The aim is to provide a comparable user experience on a lower scale.

Especially the emerging field of the Internet of Things (IoT), where many, independent devices combine to a network, opens up the market for multiple localization applications. In *smart homes* gathering sensor data might only be useful with precise localization information. For example measuring the room temperature for heating control is only beneficial, if the controller knows in which room the temperature data has been gathered. Thereby, it is possible to optimize the controlling to save energy, money, and to improve the user experience. But not only the private market has an urge for precise indoor localization. The possible use cases for such a technology spans from applications in the field of robotics [39] over healthcare [32], squad localization for firefighters [49] to warehouse management [18], and many more.

Recently the rise of ultra-wideband (UWB) technology created new opportunities in regard to the achievable localization resolution for radio devices. In contrast to common narrowband technology, ultra-wideband makes use of a high bandwidth. Data does not need to be transmitted in long packages spread over time, but in short pulses spread over a higher bandwidth. The evolution towards very narrow pulses allows an increasing timing resolution. The reception and transmission of a frame can now be tracked precisely. Based on the development of UWB new timing-based approaches for localization have been made available. Especially the rise of two-way ranging (TWR) schemes for distance estimation allows a quick and easy setup of precise localization systems achieving sub-decimetre accuracy.

1.1 Problem Statement

UWB ranging methods can be distinguished in two major groups. There are synchronous and asynchronous ranging methods. Synchronous methods such as time-of-arrival (ToA) or time-difference-of-arrival (TDoA) rely on a precise clock synchroniza-

tion of the system. Asynchronous methods such as two-way ranging (TWR) or received signal strength (RSS) do not need any synchronization among the devices.

Synchronous methods ensure a high scalability as they are able to perform a localization of a tag with a single message. Therefore, multi-tag localization can easily be achieved. However, synchronous methods tend to have a high setup complexity. Achieving a precise synchronization is a complex topic [56] and the organization of the network in available synchronous systems, such as Ubisense [57], OpenRTLS [43, 59] or Sewio [52], relies on additional infrastructure such as an ethernet backbone.

Asynchronous methods such as TWR do not need such a complex infrastructure. TWR relies on an exchange of multiple messages to determine a range. For example the so called double-sided two-way (DS-TWR) ranging needs the exchange of at least three messages. To locate a tag, it needs to perform a DS-TWR handshake with multiple anchors. The result is a high number of messages and a reduced efficiency of the system. The simpler single-sided two-way ranging scheme needs less messages, however, it is prone to different error sources that decrease the ranging performance.

Therefore, the core problem of each RTLS is to find a trade-off between precision, scalability and setup complexity. This thesis presents a SS-TWR based RTLS with a low setup complexity that still achieves a reasonable scalability and high precision.

1.2 Contribution

This thesis presents a SS-TWR based RTLS. In order to achieve this goal several problems need to be tackled beforehand.

Analysis of TWR Error Sources. This thesis analyses the main errors contributing to a decrease of SS-TWR performance. Namely, these errors are the antenna delay, the received signal strength and the clock speed offset between transmitter and receiver. Several experiments have been made to show the impact.

Improving SS-TWR. The second contribution is the introduction of compensation and calibration techniques. Based on the results from the analysis the thesis presents possible solutions to reduce the errors. Therefore, a method for the calibration of the antenna delay is described. Furthermore, a model to compensate the influence of the signal strength is presented. In order to reduce the error induced by the clock-speed offset the thesis presents two approaches. First, it is shown how to use offset estimations to improve the ranging results. Second, an online clock calibration method is introduced, calibrating the offset during runtime in order to mitigate the impact of the error sources.

Development and Implementation of a SS-TWR based RTLS prototype. The third contribution is the development of a SS-TWR based RTLS prototype. The thesis

shows a possibility to set up a highly reliable localization system with low complexity supporting the operation of multiple tags. The system has been implemented bare-metal on an embedded system featuring an ARM Cortex A3 microcontroller and the Decawave DW1000 UWB transceiver. The implementation is self sustainable meaning there is no need for further infrastructure, such as Ethernet links or a centralized location servers, and therefore allowing an easy employment in future projects.

Evaluation. The last contribution is the evaluation of the implemented prototype. The impact to the localization performance of the proposed correction and calibration methods will be evaluated. Further, the thesis evaluates the performance of the proposed prototype and compares it in terms of performance, setup complexity, reliability and refresh rate to related work from commercial and academical background.

1.3 Thesis Outline

Chapter 2 will give a brief introduction to UWB technology. Two common localization algorithms that will be used later on will be presented and analysed. Further, the chapter describes the hardware platform that was used within this thesis.

Chapter 3 will investigate the different ranging approaches for UWB. The concepts of double-sided two-way ranging and single-sided two-way ranging will be discussed in detail. Special focus will be taken on the the concept of SS-TWR and its most important error sources. Calibration and correction techniques will be discussed that improve the ranging performance of SS-TWR.

Chapter 4 will take the previous findings and design a SS-TWR based real-time location system. The system shall use the calibration and correction techniques, found in the previous chapter, in order to allow high precision localization. The chapter will give an overview of the most important parts of the implementation and presents some comparable RTLS with commercial and academical background.

In Chapter 5 the proposed system will be evaluated. In order to simulate the behaviour within a larger system a simulation model has been created. Thereby an evaluation of the localization performance of the chosen algorithms can be done for larger systems and different scenarios.

Chapter 6 will summarize and discuss the findings and give a glance on possible future work.

2

Background

This chapter gives a short introduction to the UWB technology (Section 2.1). Furthermore, Section 2.2 describes two localization algorithms. Both algorithms will be used later on during the implementation to retrieve position data from the performed rangings. Finally, Section 2.3 introduces the used hardware for this project.

2.1 Ultra-Wideband in the IEEE 802.15.4 Standard

In 2002 the Federal Communications Commission (FCC) permitted the operation of so called ulta-wideband (UWB) radios. It defines UWB as a radio signal with at least 500 MHz bandwidth or a fractional bandwith larger than 0.2 [19]. In the year of 2003 the first version of the IEEE 802.15.4 standard has been released. The standard contains specifications for physical layer (PHY) and media access control of "low-data-rate, low-power, and low-complexity short-range radio frequency (RF) transmissions" [29]. The intended use cases are low-rate wireless personal area networks (LR-WPAN) and wireless sensor networks. In 2007 the IEEE 802.15.4a standard [28] was introduced as an amendment to the IEEE 802.15.4-2006 version. It includes two new physical layers with one being the UWB PHY. This has been the first occurrence of a standardized ultra-wideband technology for data transmission.

The aim of the new physical layer was to provide high-precision ranging within wireless sensor networks. This feature has not been a focus of the previous versions of the standard. Existing localization applications, for example based on received signal strength (RSS) measurements, at best only achieve an accuracy of about 1 to 2 meters [31, 39]. The UWB PHY, on the other side, provides ranging information with an accuracy below one metre as will be seen in Chapter 3.

Furthermore, the number of wireless devices using the unlicensed 2.4GHz ISM band is increasing steadily [6, 24]. Especially due to the rise of sensor networks and the trend towards ubiquitous, networked computing (Internet of Things), these frequencies become more and more crowded. By using UWB, one can avoid interferences with these devices. Indeed, UWB can be used within three frequency bands [29] that provide 16 different physical channels (see Table 2.1).

	channel	center frequency [MHz]	bandwidth [MHz]
sub-GHz band	0	499.2	499.2
low band	1	3494.4	499.2
	2	3993.6	499.2
	3	4492.8	499.2
	4	3993.6	1331.2
high band	5	6489.6	499.2
	6	6988.8	499.2
	7	6489.6	1081.6
	8	7488.0	499.2
	9	7987.2	499.2
	10	8486.4	499.2
	11	7987.2	1331.2
	12	8985.6	499.2
	13	9484.8	499.2
	14	9984.0	499.2
	15	9484.8	1354.97

Table 2.1: UWB PHY band allocation according to IEEE 802.15.4a [28].

2.1.1 Frame Structure

The IEEE 802.15.4 standard proposes a certain frame format. An illustration thereof can be seen in Figure 2.1. The package consists out of three parts: the synchronisation header (SHR), the physical layer header (PHR) and a data field containing the payload. Next each part is described explicitly.

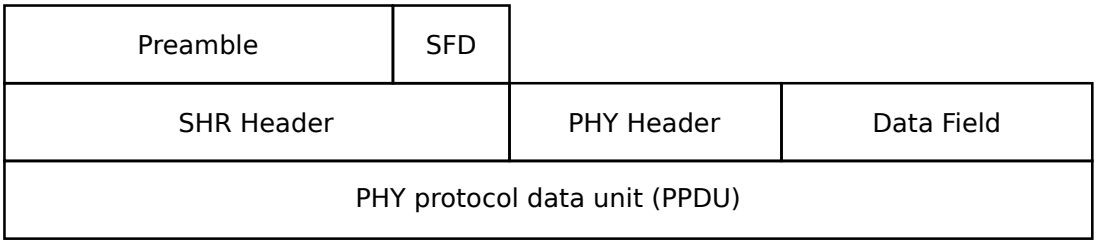


Figure 2.1: Structure of a IEEE 802.15.4 frame.

SHR Header. The synchronization header consists of a preamble and a start-of-frame delimiter (SFD). The preamble is a repetition (16, 64, 1024 or 4096) of a predetermined symbol. A symbol consists of a length 31 (optional 127) ternary code. The combination of a channel and the preamble forms a so called complex channel. The preamble thereby serves the purpose of identifying the begin of a new package as well as synchronizing the transmitter and receiver device. The reason for longer preamble codes is to support non-coherent receivers by improving the signal-to-noise ratio via processing gain [50].

It is possible to run multiple complex channels with differing preambles on the same physical channel. Further, for some channels there are desired interferences with other overlapping channels in order to allow inter-channel communication by using the same preamble [29].

The SFD indicates the end of the preamble and the start of the PHY header. Furthermore, the reception of the SFD marks the time of reception of the package. The resulting timestamp will be of high importance for the ranging process. The SFD itself is a fixed 8 or 64 preamble symbol code.

PHY Header. The physical header contains the necessary information for the further reception. Its size is 19 bits. It contains the data rate for the following data field, the length of the package, a bit indicating that the package is used for ranging and the preamble duration. Sending the preamble duration is important to improve ranging by tracking the duration of the preamble [29] (see receiver time tracking in Section 3.4.3). Further, the PHY contains a hamming block for the correction of single bit errors and detection of two bit errors.

Data Field. The data field contains the payload of the package. In the standard its length is bound to 127 bytes by the length value defined within the PHY header. However, vendors such as Decawave also implement a non-standard PHY header that allows up to 1023 bytes payload [14]. They also mention that package errors may increase by using a larger payload.

2.1.2 Modulation

A IEEE 802.15.4 package is modulated in two ways. The first part the SHR header (preamble + SFD) consists of single pulses. Information is coded in a ternary alphabet (-1,0,+1) that are mapped to positive, negative or no pulse.

The remaining PHY header and the data payload is modulated using a so called BPM-BPSK scheme. BPM stands for burst position modulation whereas BPSK refers to binary phase-shift keying. Each burst thereby represents two bits depending on the burst position and the burst polarity (see Figure 2.2). So called 'guard intervals' are introduced between the bursts in order to "provide resistance to intersymbol interference" [10]. Further, the exact burst position within its burst interval is chosen pseudo-random, in order to prevent multi-user interferences.

2.1.3 Ranging Support

The crucial property that enables high-precision ranging within the UWB standard is the high time resolution due to the high bandwidth provided by the transceivers. Thus, it is possible to determine a precise time of arrival of each frame. In praxis the timestamp is taken after the reception of the preamble and at the reception of the SFD (see Section 2.1.1). Providing this high timing resolution makes UWB an excellent choice

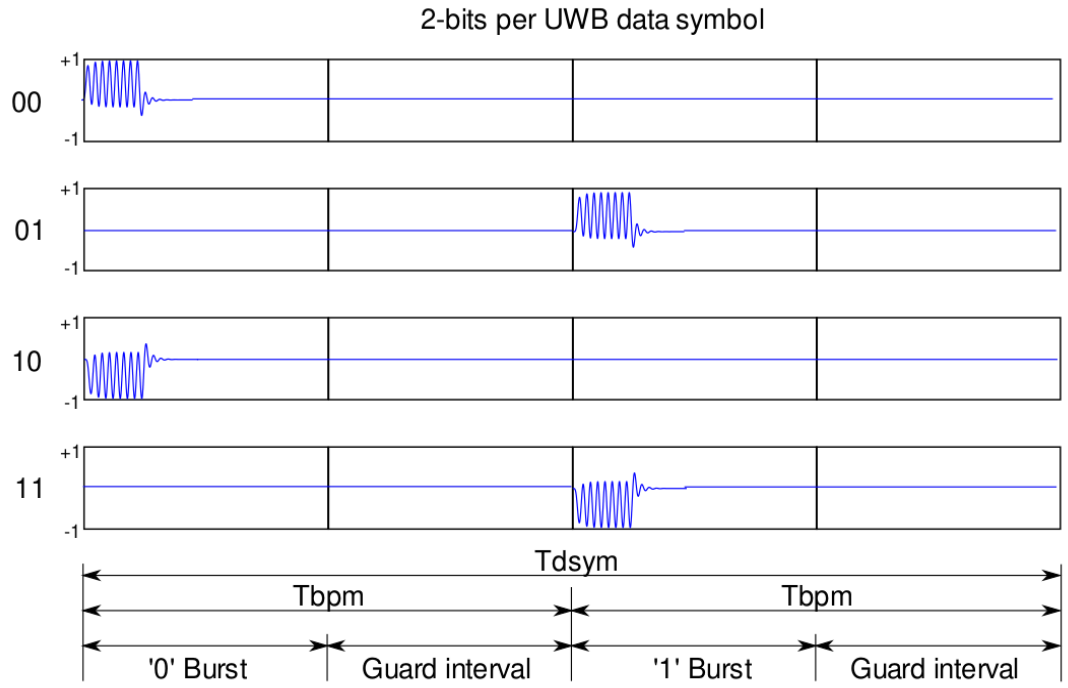


Figure 2.2: BPM-BPSK modulation of UWB data symbols (taken from [10])

for timing based ranging methods such as Time-of-Arrival, Time-Difference of Arrival, or Two-Way Ranging (see Section 3.1).

2.1.4 Media Access Strategies

Media access in wireless networks is a fairly well researched topic. For common narrowband radio technologies such as Wi-Fi, Bluetooth, or Zigbee there already exist several different channel access strategies. However, existing narrowband solutions may fail because of the unique properties of UWB.

In the context of narrowband radio, carrier sense multiple access (CSMA) via energy detection is often used to check for a free channel. Due to the nature of UWB, this mechanism cannot be used because the signal vanishes in the background noise. If a receiver missed the preamble the device thereby cannot distinguish between a free and an occupied channel. However, the possibility of carrier sensing is crucial for CSMA. Thus, the IEEE 802.15.4 standard suggests a clean channel assessment (CCA) method for UWB. Compatible transmitters should split the data field of the frame into smaller chunks and insert preamble symbols in between. Receivers thereby can listen for these preamble symbols and identify an ongoing transmission even if they missed the leading preamble. Unfortunately, the DW1000 does not support this feature by itself. However, Decawave recently presented a pseudo CCA example, implementing this feature at software level [17]. It should be noted, that including preamble symbols within the

data part increases the transmission time of the frames. Furthermore, listening for preamble symbols, in contrast to the energy detection of narrowband, is very energy consuming. On the other hand, the more preamble symbols are inserted the higher the possibility of a successful CCA is going to be [44]. A tradeoff between increasing frame duration, energy efficiency and the success rate of the CCA needs to be taken [44].

For the time-division multiple access (TDMA) approach participating devices allocate a time slot at which they can send their messages without the risk of interference by other devices. Such approaches often need a centralized coordinator having the authority to schedule the transmission times for other devices. These schedules then will be spread across the network so every participant knows its reserved slot. UWB based TWR-RTLS systems following the TDMA appoach are for example SurePoint [33] or Decawaves' own localization approach [15]. SurePoint works with a centralized coordinator that manages the network by using a flooding mechanism called *Glossy* [20, 33]. Decawave implemented a decentralized version, where devices manage their surrounding collaboratively, but keep synced by a common superframe.
TDMA systems allow a high utilization of the channel [47], however, they come at the cost of a certain amount of network organization. Further, timing constraints for transmissions are quite strict, limiting the number of participating devices as well as the number of performed rangings and thereby reduce the achievable update rate.

ALOHA is a simple so called random access approach introduced 1970 by Abramson [1]. Participating devices just transmit messages at free will. This very basic approach does neither avoid, identify nor resolve collisions. Transmitted messages may arrive or not. As Ridolfi [47] pointed out, using ALOHA in a low traffic environment can be a viable option. If the channel occupation is below 18 %, the possibility of a successful transmission is 97 %.
The simplicity of ALOHA allows the easy setup of the network. There is no need for network infrastructure such as coordinators and devices can communicate independently. The downside is a low channel occupation if the possibility of collisions should be minimized.

2.2 Localization Algorithms

Based on the distance estimation between a mobile device (tag) and static base stations (anchors) one can estimate the position of the tag. Especially in the field of embedded systems the selection of a proper localization algorithm is bound by some limitations. One needs to balance factors such as computational effort, energy efficiency, and the resulting accuracy. Further, the used hardware platform will yield limitations. Does the microprocessor contain an FPU for floating point operations? Is the microprocessor fast enough to perform the computation in reasonable time?

The following section is going to investigate two common localization solutions. The two selected algorithms are known to work well in embedded systems, because of their simplicity and low computational requirements. Special care will be taken to survey

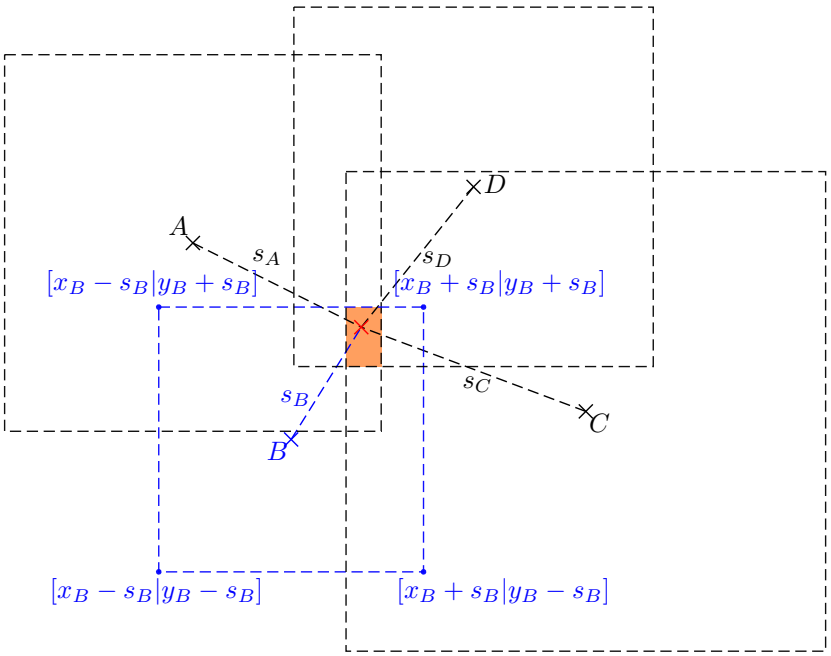


Figure 2.3: Graphical representation of the MinMax Algorithm.

their scaling behaviour in terms of the needed floating point operations.

2.2.1 MinMax Algorithm/Bounding Box

The MinMax algorithm, that is also known as bounding box algorithm, is a simple, geometric solution to the localization problem [2, 3, 35, 60, 61]. The main idea is to set up a square shaped bounding box around each anchor node with an edge size two times the measured distance. The intersection of bounding boxes form an intersection box in which the searched position should be found. The upper and lower perimeters of this intersection box can be found by taking the maximum of all lower bounding box perimeters and the minimum of all upper perimeters, respectively. The position can be estimated at the center of the intersection box. Figure 2.3 shows an example of a MinMax localization algorithm in action.
The main benefit of the algorithm is its low complexity and relatively good performance, especially when the tag is surrounded by anchors (i.e. within the convex hull) [60, 2]. As building the bounding boxes depends on the number of anchors ($O(n)$), and the intersection box is build with of a constant number of lookups ($O(n)$)„ the complexity of the algorithm lies within $O(n)$, for n being the number of used anchors. The algorithm is mostly used for two-dimensional localization but can easily be expanded to three dimensions by using cubes instead of squares as bounding boxes. The complexity remains linear with the number of anchors in this case.

2.2.2 Linear Least Squares

Lateration describes the calculation of a new position based on the distance estimation to already known points. Linear least squares (LLS) can be seen as the most straightforward approach towards lateration and has established as a quasi-standard reference for localization algorithms [2, 3, 35, 41]. The concept of LLS is to find a common intersection of circles around the anchors. The radius of the particular circle is given by the estimated distance between the tag and the anchor. Each ranging results in an equation. By solving the resulting system of equations a tag position can be determined.

However, as will be seen in Chapter 3, ranging is prone to noisy measurements. Therefore, the system cannot be solved unambiguously. Thus, the position that minimizes the sum of the squared residuals of the measured distances - the so called least squares solution - will be searched.

As this problem is non-linear, there is either the need for an optimization algorithm (e.g., Gauss-Newton) or one has to linearise the problem. Based on that, lateration algorithms are distinguished in non-linear least squares (NLLS) and linear least squares (LLS) algorithms. As NLLS solutions tend to have a high computational effort, they are non-applicable in the context of an embedded system. Therefore, the following part will focus on the LLS approach following [41].

Let $n \in \mathbb{N} \geq 4$ be the number of anchors and $0 < i < n$ be its index. Let $P_1, P_2, ...P_n$ be the coordinates of the anchors with $P_i = (x_i, y_i, z_i) \in \mathbb{R}^3$. Further, let Q be the coordinates of our tag with $Q = (x_Q, y_Q, z_Q) \in \mathbb{R}^3$. During the ranging process the distances $r_i = |Q - P_i|$ had been retrieved.
For each anchor an equation will be set up:

$$r_i^2 = (x_Q - x_i)^2 + (y_Q - y_i)^2 + (z_Q - z_i)^2 \tag{2.1}$$

with r_i being the radius of the sphere surrounding P_i. By rearranging (2.1) one gets:

$$\begin{aligned} r_i^2 &= (x_Q^2 - 2x_Q x_i + x_i^2) + (y_Q^2 - 2y_Q y_i + y_i^2) + (z_Q^2 - 2z_Q z_i + z_i^2) \\ &= (x_Q^2 + y_Q^2 + z_Q^2) - 2x_Q x_i - 2y_Q y_i - 2z_Q z_i + (x_i^2 + y_i^2 + z_i^2) \end{aligned} \tag{2.2}$$

that forms a nonlinear system of equation with the unknowns x_Q, y_Q, z_Q. Some sources [2, 3, 35] suggest to linearise by subtracting the last equation from the first $n-1$ other equations. However, by picking a noisy measurement this approach would worsen all other measurements. Following Meissl et al. [41] one can also linearise by introducing an additional unknown $u_Q = x_q^2 + y_q^2 + z_q^2$ which seems to be a more robust way in this case.

$$r_i = \underbrace{\frac{1}{r_i}}_{a_i} \underbrace{(x_Q^2 + y_Q^2 + z_Q^2)}_{u_Q} - \underbrace{\frac{2x_i}{r_i}}_{b_i} x_Q - \underbrace{\frac{2y_i}{r_i}}_{c_i} y_Q - \underbrace{\frac{2z_i}{r_i}}_{d_i} z_Q + \underbrace{\frac{1}{r_i}(x_i^2 + y_i^2 + z_i^2)}_{e_i} \tag{2.3}$$

Thereby, it has become a linear system of equations with four unknowns x_Q, y_Q, z_Q, u_Q of the form:

$$r_i = a_i u_Q + b_i x_Q + c_i y_Q + d_i z_Q + e_i \tag{2.4}$$

As the ranging measurements are noisy the system will not have a unique solution. Therefore, the least squares solution will be calculated where the sum of the squared residuals gets minimized. For the radius r_i of the spheres around the anchors we now set the measured distances s_i. To accommodate for the noisy measurements we add v_i to the left side, which models the residuals. By inserting in (2.4) one gets:

$$s_i + v_i = a_i u_Q + b_i x_Q + c_i y_Q + d_i z_Q + e_i \tag{2.5}$$

with

$$\begin{aligned} a_i &= \frac{1}{s_i} \\ b_i &= -\frac{2x_i}{s_i} \\ c_i &= -\frac{2y_i}{s_i} \\ d_i &= -\frac{2z_i}{s_i} \\ e_i &= \frac{1}{s_i}(x_i^2 + y_i^2 + z_i^2) \end{aligned} \tag{2.6}$$

This equation can be rewritten as matrix operation to obtain a linear system of equations:

$$\underbrace{s_i - e_i}_{\Delta s_i} + v_i = \begin{bmatrix} a_i & b_i & c_i & d_i \end{bmatrix} \begin{bmatrix} u_Q \\ x_Q \\ y_Q \\ z_Q \end{bmatrix} \tag{2.7}$$

resulting in a system of equations

$$\begin{bmatrix} \Delta s_1 \\ \Delta s_2 \\ \vdots \\ \Delta s_n \end{bmatrix} + \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} = \begin{bmatrix} a_1 & b_1 & c_1 & d_1 \\ a_2 & b_2 & c_2 & d_2 \\ \vdots & \vdots & \vdots & \vdots \\ a_n & b_n & c_n & d_n \end{bmatrix} \begin{bmatrix} u_Q \\ x_Q \\ y_Q \\ z_Q \end{bmatrix} \tag{2.8}$$

abbreviated as

$$\Delta s + v = Ax \tag{2.9}$$

For a linear least squares solution the quadratic sum of the deviations $v^T v$ has to be minimized with respect to the parameter x. The objective function $\Phi(x)$ is defined as

$$\begin{aligned} \Phi(x) &= v^T v \\ &= (Ax - \Delta s)^T (Ax - \Delta s) \\ &= (x^T A^T Ax) - (x^T A^T \Delta s) - (\Delta s^T Ax) + (\Delta s^T \Delta s) \\ &= (x^T A^T Ax) - 2(x^T A^T \Delta s) - (\Delta s^T \Delta s) \end{aligned} \tag{2.10}$$

To find the minimum of $\Phi(x)$ with respect to x, the gradient of equation 2.10 has to be set to zero $\nabla_x \Phi(x) \stackrel{!}{=} 0$. It is given by

$$\nabla_x \Phi(x) = 2A^T Ax - 2A^T \Delta s \tag{2.11}$$

therefore

$$2A^T A\widetilde{x} - 2A^T \Delta s \stackrel{!}{=} 0 \tag{2.12}$$

must hold and the optimal parameter $\widetilde{x}$ - which is the calculated position - can be computed by

$$\widetilde{x} = (A^T A)^{-1} A^T \Delta s \tag{2.13}$$

Hypothesis: Solving the LLS problem can be achieved in $O(n)$ for n being the number of anchors.
Proof: The matrix A will be a $4 \times n$ matrix, whose setup complexity is in $O(n)$. The matrix multiplication $A^T A$ results in a 4×4 matrix. Therefore, the calculation can be done with $16 * 2n$ operations which is in $O(n)$. Calculating $A^T \Delta s$ takes $4 * 2n$ operations - as A^T is a matrix with four rows - which is in $O(n)$. As the result of $A^T A$ is a symmetric, positive definite 4×4 matrix the system of linear equations can be solved with a Cholesky factorization [51]. The effort for that is constant as the system of linear equations is of constant size.
The highest order term is scaling linear $O(n)$. Therefore, the whole algorithm scales linear in regard to the number of used anchors n. However, the good scaling performance shall not lead to the conclusion that LLS is a fast method for localization. Compared to MinMax its effort is notably higher as will be seen during the evaluation in Section 5.5. It shall show that the effort of LLS does not explode for a large number of anchors. Therefore, its effort is sustainable by an MCU with limited resources.

2.3 Employed Hardware

The used nodes consist of a 'Nucleo-64' development board produced by STM and a DW1000 development board that has been developed at the TU Graz by Bernd Baumann as part of his master project [5]. The nucleo boards have a common Arduino compatible pin connector. Via this connector the DW1000 board can be stacked on top of the Nucleo board (see Figure 2.4).

2.3.1 Decawave DW1000

In November 2013, Decawave announced the launch of their first UWB chip - the DW1000 [9]. The device has been the first consumer grade IEEE 802.15.4-compatible UWB chip available on the market. The communication between the microcontroller and the chip is done via a SPI interface. The DW1000 can communicate over the SPI with up to 20 MHz.

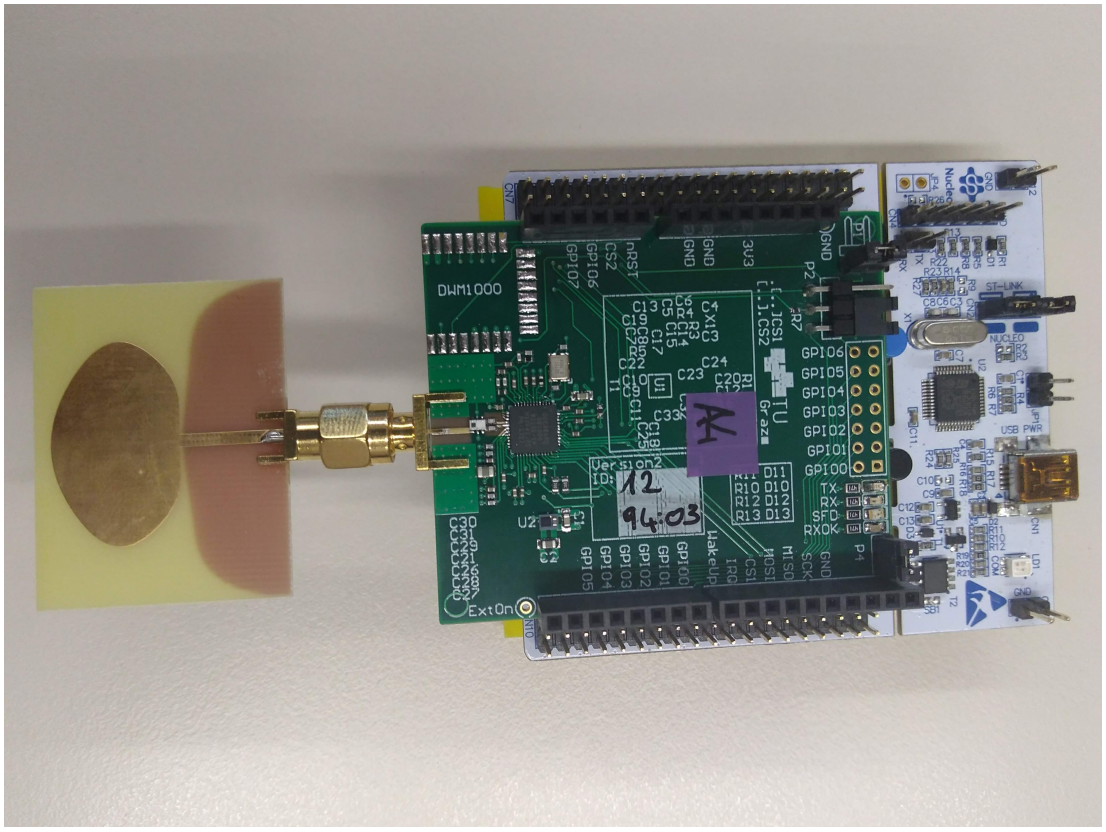


Figure 2.4: Image of a UWB node used in this thesis. The DW1000 board (green) is connected to the Nucleo-64 board (white) via an Arduino-compatible connector.

The DW1000 has a so called coherent receiver, meaning that it performs further signal analysis in order to improve the signal quality. Therefore, longer ranges are possible [13]. Furthermore, the device is able to analyse the received signal and "resolve the channel in detail and determine the arrival time of the first(most direct) path, even when attenuated" [14] by identifying signal echos and using their energy to improve the signal.

The DW1000 includes some power management features for improving energy efficiency. Besides the standard power state, the device also supports a state called DEEPSLEEP. Within this state the device powers down most of its parts in order to save energy. A state called SLEEP works in a similar way, however, there is a counter that wakes the chip periodically, e.g., for listening to incoming messages.

The device has been designed with localization in mind. Therefore, it provides reception and transmission timestamps with an accuracy of about 15 ps. This results in a high timing resolution suitable for accurate two-way ranging and other timing-based ranging methods (see Section 3.1). Further, the chip offers the possibility to schedule future

transmissions with an accuracy of 8 ns, which allows to embed a transmission timestamp in the message itself. This feature drastically reduces the ranging effort by reducing the number of needed transmissions.

2.3.2 STM32L152RE

STM produces several different Nucleo-64 boards that contain different mircoprocessors, however, they share the same layout. The used Nucleo-64 board has a 32 bit ARM Cortex M3 processor clocked with 32 MHz top speed. As memory, it features a 512 kB flash memory as well as 80 kB SRAM. Further, the STM32L152RE has a floating point unit (FPU) so floating point operations can be performed. The MCU allows serial communication via UART (up to 115 200 bit/s) and SPI (up to 16 MHz clock speed). Further, the microprocessor has a lot of basic GPIO functionality including the handling of external interrupts.

3 Distance Estimation and Calibration

Sufficiently precise and accurate distance estimations are key to the performance of a localization system. There are several different approaches towards ranging with radio frequency devices. As described in Chapter 2.1, in contrast to other RF-technologies, UWB has already been designed with a focus on ranging. Therefore, it increases the ranging performance in terms of accuracy and precision. However, the ranging performance is still prone to multiple influences. In order to further improve, one needs to deal with these influences.

This chapter gives an overview of different ranging methods that can be used with UWB (Section 3.1). Further, it describes the field of single- (Section 3.2) and double-sided (Section 3.3) two-way ranging in detail. Finally, it investigates the accuracy and precision of single-sided two-way ranging and show how to optimize the ranging performance (Section 3.4). Thereby, a step-by-step calibration and correction technique will be developed (see Section 3.5) incorporating the previous findings.

3.1 UWB Ranging Techniques

There are several different approaches towards ranging with UWB. They apply for different use cases and require a variable amount of effort. This section briefly wants to present some common approaches of UWB ranging before looking at two-way ranging in more detail in the following sections.

Received Signal Strangth (RSS) RSS is a common ranging procedure already well known for ranging with narrowband technologies such as ZigBee [31, 26], Bluetooth [4] or WiFi [7]. To estimate a distance to the transmitter, the receiver tracks the RSS of the messages. If the transmission power of the transmitter is known, the receiver can thereby calculate the signal loss. Due to the link between distance and signal loss, the receiver can estimate a distance.
Using RSS in combination with UWB is possible [25, 22], however, the high timing resolution of UWB enables the usage of more promising ranging techniques.

Time of Arrival (ToA) Time of arrival makes use of the high timing resolution UWB devices offer. If transmitter and receiver are highly synchronized, the distance can be

estimated by the absolute time difference between transmission and reception of a package. The challenge of this approach is to achieve a sufficiently precise synchronization among all devices [56].

Time Difference of Arrival (TDoA) For time difference of arrival only the anchors need to be synchronized among each other. A mobile tag sends a ranging message. From the time difference of the message reception the anchors can conclude a position of the tag.

Two-Way Ranging (TWR) Two-Way Ranging works with a handshake exchange between tag and anchor. By measuring the round-trip time, one can predict the time-of-flight of the packages. Therefore, neither anchors nor the the tag needs to be synchronized by a common clock if they can just measure the reception and transmission timestamps in a sufficiently accurate way (see Section 3.2 and Section 3.3). This vastly simplifies the setup of TWR-based localization systems. There are two approaches towards TWR. The simpler one is called single-sided TWR (see Section 3.2). It consists only of a simple two-message exchange between both ranging devices. The more complex double-sided TWR (see Section 3.3) needs an additional message, however, improves ranging performance.
As the TWR scheme benefits from the high timing resolution of UWB transceivers as well as its low requirements regarding synchronization, this thesis wants to take a look at this scheme in more detail in the next sections.

Angle of Arrival (AoA) In contrast to the previous described techniques this is not a ranging based approach. The aim of the angle of arrival approach is not to retrieve a distance estimation but to estimate the angle to the device. To measure the angle of arrival one can use directional antennas [46] or an array of antennas estimating the angle via the timing difference between the received signal [57]. Using the angle instead of the distance leads to triangulation instead of trilateration for the position determination.

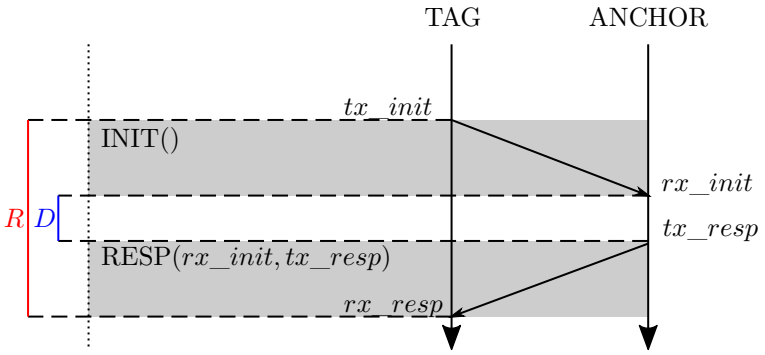


Figure 3.1: Protocol for single-sided two-way ranging (SS-TWR).

3.2 Single-Sided Two-Way Ranging (SS-TWR)

Single-sided two-way ranging (SS-TWR) is the simplest method for two-way ranging as it only requires the exchange of two messages.

3.2.1 Protocol

The ranging protocol consists of two messages (see Figure 3.1). From each message, one needs to know the time of transmission and the time of reception. The first message (INIT) is sent by the tag. It initializes a ranging attempt. The second message is the anchors response (RESP). Besides the message type it also includes two timestamps. The first timestamp corresponds to the time of arrival of the INIT-message at the anchor. The second timestamp corresponds to the time of transmission of the RESP-message itself (see Section 2.3.1). Thereby, the tag has all needed values to perform the ranging calculation.
The time-of-flight T is calculated by subtracting the overall duration of the ranging process R by the delay time of the anchor D (see Figure 3.1). In the remaining time period the two messages must have been sent, therefore, the time-of-flight for a single message has to be the half. Therefore,

$$T = \frac{1}{2}(R - D) \tag{3.1}$$

with

$$R := rx\_resp - tx\_init$$
$$D := tx\_resp - rx\_init$$

gives the time-of-flight for SS-TWR.

3.2.2 SS-TWR Clock Speed Offset Error

A main source of error is known to be the varying clock speed between tag and anchor [11, 36, 30]. In order to measure the impact, Nierynck et al. [42] introduced a simple model. Let T' be an estimation of the time-of-flight T with:

$$\begin{aligned} T' &= \frac{1}{2}(R' - D') \\ R' &:= R(1 + e_A) \\ D' &:= D(1 + e_T) \end{aligned} \tag{3.2}$$

where e_T and e_A are the clock speed offset of the devices given in ppm. From (3.1) and (3.2)

$$\begin{aligned} E := T' - T &= \frac{1}{2}(R' - D') - \frac{1}{2}(R - D) \\ &= \frac{1}{2}(R' - D' - R + D) \\ &= \frac{1}{2}(R(1 + e_A) - D(1 + e_T) - R + D) \\ &= \frac{1}{2}(R + Re_A - D - De_T - R + D) \\ &= \frac{1}{2}(e_T R - e_A D) \end{aligned} \tag{3.3}$$

can be derived as an estimation of the systematic error. One needs to keep in mind that R depends on the delay time D. That implies that, in order to minimize the overall error, it is essential to minimize the anchors' delay time D.

3.3 Double-Sided Two-Way Ranging (DS-TWR)

The purpose of double-sided two-way ranging (DS-TWR) is to mitigate the error caused by clock speed offsets. Instead of a single ranging attempt, two consecutive rangings are performed.
Under the assumption that the delay times D_1 and D_2 are close to equal, the error can be reduced vastly as shown in Section 3.3.2. Ranging algorithms based on that approach are called symmetric.
Since equal delay times are hard to achieve there is a different approach that does not depend on this assumption. This so called asymmetric approach will be discussed in Section 3.3.3.

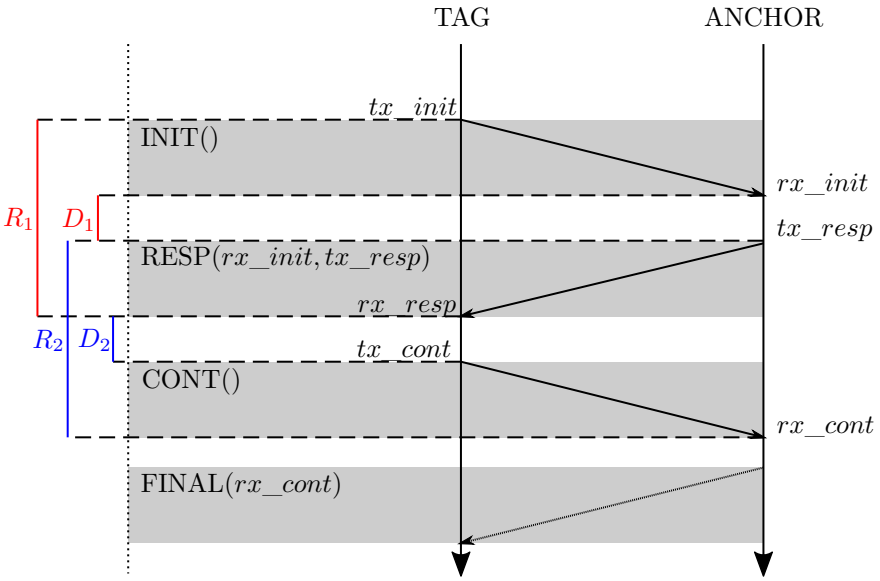


Figure 3.2: Protocol for Double-Sided Two-Way-Ranging (DS-TWR).

3.3.1 Protocol

As shown in Figure 3.2 the protocol consists of at least three messages. The first part is identical to the single-sided ranging (see Section 3.2). However, in DS-TWR the RESP message has the double function of finalizing the first ranging round and initiating another. By sending an additional CONT message, the tag finishes this second measurement. If the ranging calculation can be done by the anchor, the DS-TWR can be finished here. On the other hand, if the application demands that the ranging needs to be done by the tag, one more message (FINAL) is needed in order to transmit the timestamps of the second ranging to the tag.

3.3.2 Symmetric DS-TWR

The straightforward approach for calculating the range is the symmetric DS-TWR. From (3.1) it is known how to calculate the time-of-flight for a single measurement:

$$T_1 = \frac{1}{2}(R_1 - D_1) \tag{3.4}$$

$$T_2 = \frac{1}{2}(R_2 - D_2) \tag{3.5}$$

Since T_1 equals T_2 both functions can be added up to get:

$$\begin{aligned} 2T &= \frac{1}{2}(R_1 - D_1) + \frac{1}{2}(R_2 - D_2) \\ &= \frac{1}{2}(R_1 - D_1 + R_2 - D_2) \end{aligned}$$

Therefore,

$$T = \frac{1}{4}(R_1 - D_1 + R_2 - D_2) \tag{3.6}$$

is the time-of-flight between tag and anchor.

It can be seen that this method vastly reduces the error caused by the clock offset. Let e_T and e_A be the error induced by the clock speed offset of the tag (T) and the anchor (A) given in ppm. Respectively $k_T := 1 + e_T$ and $k_A := 1 + e_A$ can be seen as the scaling factor to the timing measurements. Thereby, the ranging error of R_1 and R_2 (3.1) following [42] can be modelled as:

$$\begin{aligned} R'_1 &:= (1 + e_T)R_1 := k_T R_1 \\ R'_2 &:= (1 + e_A)R_2 := k_A R_2 \\ D'_1 &:= (1 + e_A)D_1 := k_A D_1 \\ D'_2 &:= (1 + e_T)D_2 := k_T D_2 \end{aligned} \tag{3.7}$$

$$\begin{aligned} T' &= \frac{1}{4}(R'_1 - D'_1 + R'_2 - D'_2) \\ &= \frac{1}{4}(k_T R_1 - k_A D_1 + k_A R_2 - k_T D_2) \\ &= \frac{1}{4}[k_T(R_1 - D_2) + k_A(R2 - D1)] \\ &= \frac{1}{4}[(1 + e_T)(R_1 - D_2) + (1 + e_A)(R_2 - D_1)] \\ &= \frac{1}{4}[(R_1 - D_2 + R_2 - D_1) + e_T(R_1 - D_2) + e_A(R_2 - D_1)] \end{aligned} \tag{3.8}$$

Therefore, the overall error is:

$$\begin{aligned} E &:= T' - T \\ &= \underbrace{\frac{1}{4}[(R_1 - D_2 + R_2 - D_1)}_{=T} + e_T(R_1 - D_2) + e_A(R_2 - D_1)] - T \\ &= \frac{1}{4}[e_T(R_1 - D_2) + e_A(R_2 - D_1)] \end{aligned} \tag{3.9}$$

Substituting $R_1 = 2T + D_1$ and $R_2 = 2T + D_2$ leads to:

$$\begin{aligned} &= \frac{1}{4}[e_T(2T + D_1 - D_2) + e_A(2T + D_2 - D_1)] \\ &= \frac{1}{4}[2Te_T + e_T(D_1 - D_2) + 2Te_A + e_A(D_2 - D_1)] \\ &= \frac{1}{4}[2T(e_T + e_A) + (e_T - e_A)(D_1 - D_2)] \\ &= \frac{1}{2}T(e_T + e_A) + \frac{1}{4}(e_T - e_A)(D_1 - D_2) \end{aligned} \tag{3.10}$$

One now can see, that in case D_1 equals D_2, or at least if both values are close to equal, the second part of the error equation will be zero. Therefore, the clock speed offset error only depends on the time-of-flight T which is of negligible size when remarking that the time-of-flight is in the range of nanoseconds, whereas the delay times can be up to several microseconds.

3.3.3 Asymmetric DS-TWR

Another approach is the so called asymmetric DS-TWR proposed by Neirynck et. al [42]. Its advantage compared to the symmetric variant is the fact that the delay times D_1 and D_2 do not have to be equal. Instead of summing up both single-sided rangings they now will be multiplied.

$$\begin{aligned} R_1R_2 &= (2T + D_1)(2T + D2) \\ &= (2T)^2 + 2TD_2 + 2TD_1 + D_1D_2 \\ &= 2T(2T + D_2 + D_1) + D_1D_2 \\ R_1R_2 - D_1D_2 &= 2T(\underbrace{2T + D_2}_{\text{(Equation 3.5)}\Rightarrow = R_2} + D_1) \\ &= 2T(R_2 + D_1) \end{aligned} \tag{3.11}$$

$$\Rightarrow T = \frac{R_1R_2 - D_1D_2}{2(R_2 + D_1)} \tag{3.12}$$

As in Section 3.3.2 one can now take a look at the clock error.

$$\begin{aligned} T' &= \frac{R'_1R'_2 - D'_1D'_2}{2(R'_2 + D'_1)} \\ &= \frac{k_TR_1k_AR_2 - k_AD_1k_TD_2}{2(k_AR_2 + k_AD_1)} \\ &= \frac{k_Tk_A}{k_A}\frac{R_1R_2 - D_1D_2}{2(R_2 + D1)} \\ &= k_T\frac{R_1R_2 - D_1D_2}{2(R_2 + D1)} \\ &= k_TT \end{aligned} \tag{3.13}$$

Because k_A gets cancelled in the nominator and the denominator the remaining error just depends on k_T. Therefore, the overall error

$$E := T' - T = k_TT - T \tag{3.14}$$

is independent of the delay times.

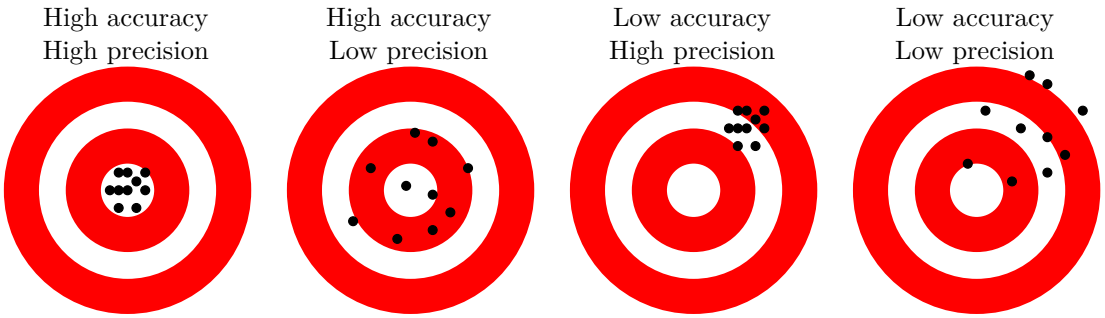


Figure 3.3: Schematic representation of accuracy and precision.

3.4 Sources of Error and Calibration

There are two measurement units to describe the performance of a ranging method. The accuracy describes how far from the true distance the measurements mean lies. The precision on the other hand gives an overview of the measurements deviation (see Figure 3.3). Sources of error may contribute to a lower accuracy as well as a decrease in precision. Following the most relevant sources of error are described that influence accuracy and precision of the distance estimation. All following evaluations have been taken on channel 2 with a preamble length of 128 symbols, a PRF of 64 MHz and a 6.8 Mbit/s data rate. The impact of different configurations has not been evaluated for this thesis.

3.4.1 Antenna Delay

The accuracy of the receive and transmit timestamps is key to the ranging accuracy. However, it is unknown how long a signal is going to propagate from the chip to the antenna. Decawave states that propagation times vary form chip to chip [12]. Furthermore, they identify the components between the antenna and the DW1000 as potential influences of the delay. Also the temperature may vary the delay. Decawave observed an accuracy change of 2.15 mm per degree Celsius [12]. Kempke et al. [33] also observed an influence of the transmission power.

In order to show the impact of the antenna delay, formula (3.1) for SS-TWR will be expended. Let $i \in [1:4]$ be the index of the timestamps. m_i corresponds to the measured timestamps within the UWB device. t_i describe the true timestamps when the message left or arrived at the antenna (see Figure 3.4). ΔA and ΔB describe the antenna delays for device A and B, respectively. Reception and transmission delays

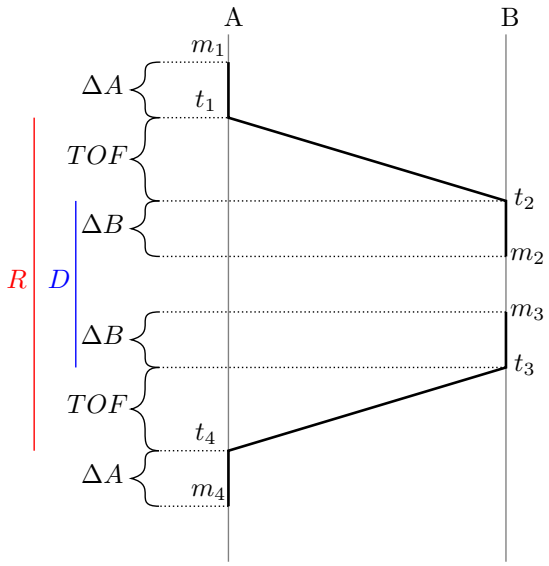


Figure 3.4: Impact of the antenna delays ΔA and ΔB. m_i (for $i \in [1:4]$) correspond to the measured timestamps, while t_i correspond to the true timestamps.

are assumed to be equal.

$$\begin{aligned} TOF &= \frac{1}{2}(R - D) \\ &= \frac{1}{2}((t_4 - t_1) - (t_3 - t_2)) \\ &= \frac{1}{2}(t_4 - t_1 - t_3 + t_2) \\ &= \frac{1}{2}((m_4 - \Delta A) - (m_1 + \Delta A) - (m_3 + \Delta B) + (m_2 - \Delta B)) \\ &= \frac{1}{2}(m_4 - m_1 - 2\Delta A - m_3 + m_2 - 2\Delta B) \\ &= \frac{1}{2}(m_4 - m_1 - m_3 + m_2) - \Delta A - \Delta B \end{aligned} \tag{3.15}$$

It thereby can be seen that the antenna delay results in a constant offset and wrong antenna delay estimations, which lead to a decrease in accuracy.

The DW1000 has the possibility to account for a known delay to the reception and transmission timestamps [14]. Therefore, the delays have to be determined before performing the ranging. In their application note, Decawave suggests a setup for evaluating the delays [12]: They suggest setting up three uncalibrated devices as a isosceles triangle with known length. Then rangings between each pair of devices will be performed. The result is a euclidean distance matrix containing the estimated distance between each node. From the distance matrix the measured time-of-flight is derived. By comparing the measured time-of-flight with the theoretical time-of-flight - that can be computed by the known distances - an optimization problem can be set up, where best fitting delays for all three devices are searched. After applying these values to the nodes, these

three nodes can be considered to be calibrated. All further nodes can now be calibrated by performing rangings with known distance to one of the calibrated nodes and adapt their delays accordingly.

If all devices are of the same type, one can assume that their antenna delay is close to equal. Calibrating each device separately can thereby be neglected. Decawave refers to an accuracy decrease below 30 cm due to board and component differences [12]. A proper individual calibration of all devices is out of the scope of this thesis. Therefore, the antenna delay has been set to the default value proposed by [5] for all following measurements.

3.4.2 Received Signal Strength

According to [11] the received signal strength affects the ranging performance. Presumably the leading edge detection algorithm of the DW1000 is responsible for this error [40]. The purpose of this algorithm is to identify the first path of a transmission and correct the receive timestamp accordingly to prevent wrong timestamps due to multipath effects. Similar to the antenna delay, the received signal strength causes a loss in accuracy. However, in contrast to the antenna delay, the received signal strength varies. The actual signal strength and thereby the offset is influenced by the antenna gain, transmit power, and other sources of loss or gain [11]. Further, there are more dynamic influences towards the received signal strength. For example an obstacle such as a wall might influence the current transmission and the signal strength. However, indicator for the signal strength loss is the measured distance itself as signal strength decreases to the square of the distance. Therefore, this influence can also be corrected via the estimated ranging distance.

In order to evaluate the impact of the received signal strength, an experiment has been made: The experiment has been executed in a long, straight hallway whose walls consist of concrete. The two ranging devices have been mounted on a tripod in 1.5 metres height and placed in the middle of the hallway. The measurements have been taken from one metre up to 50 metre in one metre steps. For each evaluation point, 1000 distinct measurements have been made. All rangings have been performed on Channel 2 with a preamble length of 128 symbols and a 6.8 Mbit/s data rate.

In Figure 3.5 one can see the results of the experiment. One sees that especially below 10 metre there is a drop of accuracy. Rangings within this interval therefore might be corrected with an according linear approximation. High peaks such as the measurements between 14 and 23 metres presumably have been caused by interferences from echos of the received signal [23].
In order to correct the error caused by the varying signal strength a linear fitting can be made for the critical interval between 0 m to 10 m. This fittings correspond to the orange line and red line in Figure 3.5 respectively. Via the formula

$$r_{corr} = \begin{cases} r - (-0.28 + 0.028r) & < 10m \\ r & >= 10m \end{cases} \tag{3.16}$$

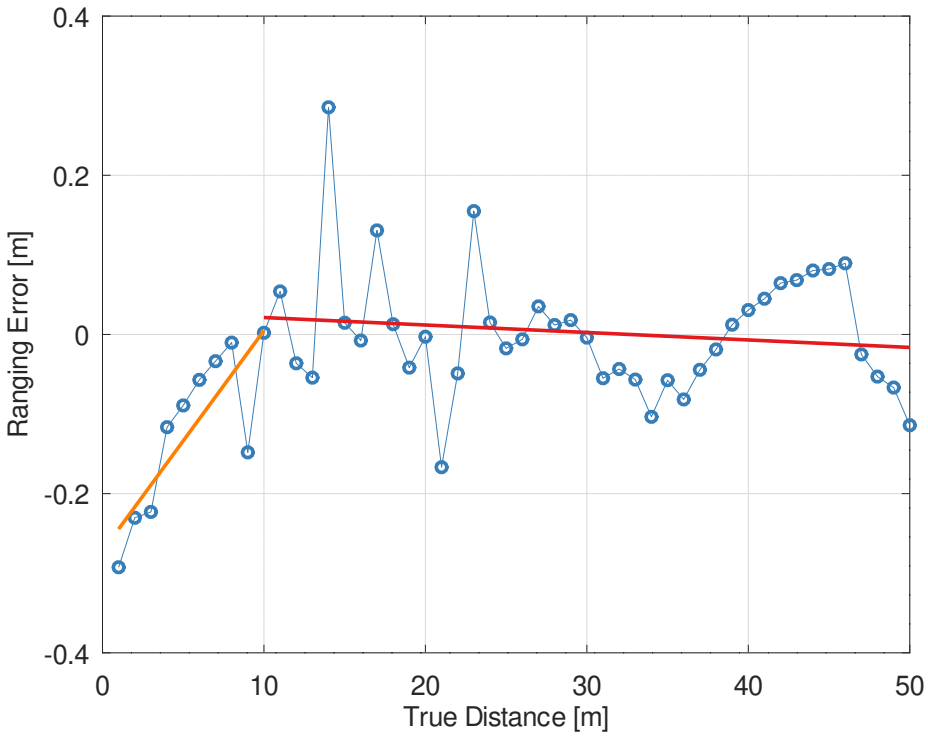


Figure 3.5: Signal strength based ranging error. Orange line is a linear approximation of the 0-10 metre interval. Red line is a linear approximation of the 10-50 metre interval.

a corrected distance r_{corr} can be calculated in dependence of a ranging measurement r. Martel et al. [40] propose a similar technique. They have also observed an offset below 10 m and corrected these measurements with a linear interpolation. However, they proposed different parameters for the linear interpolation ($r_{corr}(r) = 0.969r + 0.0979$ for $r < 10m$). These different observations might be rooted in the usage of different board components with different properties.
Decawave [11] suggests that also a low signal strength will result in a loss of accuracy, however, this effect could not be seen during this experiment. Presumably this effect will be seen for larger distances than measured here. However, it is questionable if distances larger than 50 metres are of high relevance in an indoor positioning scenario. Please note that these results heavily depend on the used device type and are not universally valid. Due to different antenna gains and other factors this measurements might not fit for other devices.

3.4.3 Clock Speed Offset

In order to properly measure time on a device, clock accuracy is key. If the clock is too fast, the measured interval will be larger compared to the real-time that passed. If the clock is too slow, the measured time will be smaller. The accuracy of TWR

highly depends on accurate timing measurements. Both the speed offset among the two ranging devices as well as their speed offset compared to the real time have an impact.
In presence of a clock speed offset among the devices the timing information received by the anchor will be interpreted wrong by the tag. A reported respond delay will be interpreted longer or shorter and thereby increases or decreases the calculated time-of-flight and directly influences the accuracy of the ranging. The clock speed offset among two devices will normally be measured in parts-per-million (ppm).

In a centralized synchronous network all devices are synchronized by a common master-clock [8]. Thereby, ideally there is no speed offset among the devices and a speed offset towards the real time can be mitigated by calibration of the master-clock. However, the setup for such a network and the needed synchronization is difficult [56].
Asynchronous networks thereby are much more common. However, sensor nodes typically lack of high accuracy crystals. Reasons therefore can be seen in pricing restrictions and economical considerations [62]. Furthermore, energy efficiency often demands the use of simpler, more efficient oscillators [5]. Therefore, clock accuracy is often influenced by operating conditions such as temperature, load capacitance or drive level [63]. In 2007 Zhen et al. [63] claimed a typical crystal stability of ± 40 ppm for a common sensor node. However, in 2013 D'Amico et al. [8] reported typical clock-frequency offsets of about 5 ppm to 20 ppm. The crystal oscillator used on our nodes states a 10 ppm stability [5].

The DW1000 generates its own system clock of 19.2 MHz with an external reference oscillator. In order to reduce the influence of the clock speed offset due to imprecise oscillators there is the possibility to perform a clock-trim. By setting the according five bit FS_XTALT register it is possible to manipulate the system clock speed in roughly 1.5 ppm steps. For a factory calibration the DW1000 can produce a continuous wave which can be observed with a spectrum analyser. By adapting the the trim value one can manipulate the frequency until it matches the expected frequency [14].

In order to lessen the error due to clock speed offset the IEEE 802.15.4-2015 [29] standard proposes a solution for an offset estimation: the crystal characterization. The Decawave DW1000 UWB transceiver calls this approach 'Receiver Time Tracking' (RTT) [14]. Further, the DW1000 implements another estimation procedure called 'Carrier Recovery Integrator' (CRI) [14] . Both methods give an estimation of the relative clock speed offset between the transmitter and the receiver.
These values are given in parts per million (ppm). Let δ be an estimation of the clock speed offset (either RTT or CRI) given in ppm. Further, R is the measured ranging interval and D is the response delay (see Figure 3.1). A corrected measurement of the time-of-flight T_{corr} can be calculated by adapting formula (3.1)

$$T_{corr} = \frac{1}{2}(R - (D * (1 - \delta))) \tag{3.17}$$

In the following the characteristics of both clock speed offset estimators will be investigated.

Receiver Time Tracking (RTT). In order to make a clock speed offset estimation Zhen et al. [62] presented a solution. At the beginning of the frame the receiver starts a counter. The counter increments at the local speed of the receivers clock. At the end of the package the counter will be stopped. By the number of received symbols one can calculate the theoretical duration of the package and compare it to the measured duration by the counter. Presumably the DW1000 Receiver Time Tracking works with a similar technique.
The crystal characterisation within the IEEE 802.15.4 standard defines two values: The 'receiver time tracking offset' (RX_TTCKO) and the 'ranging tracking interval' (RX_TTCKI). The RX_TTCKI represents the interval over which the offset had been tracked. The RX_TTCKO is a 19 bit signed integer that represents the offset within the interval. An offset estimation in ppm can be calculated with:

$$RTT[ppm] = \frac{\text{RX\_TTCKO}}{\text{RX\_TTCKI}} \tag{3.18}$$

For example if the transceivers clock is running 10 ppm faster than the receivers clock, 'ranging tracking offset' would be 10 if the 'ranging tracking interval' is one million. In opposite, if the receiver is faster by 10 ppm, the 'ranging tracking offset' would have the value -10 with a 'ranging tracking interval' of one million.

Carrier Recovery Integrator (CRI). Dealing with clock speed offsets among communicating devices is a common topic [63]. The DW1000 is a so called coherent receiver being able of tracking carrier timing and phase shifts [14]. The phase shift can be measured by a phase detector within a phase-locked loop (PLL) or a delay-locked loop (DLL) [63, 53]. A phase detector compares the received carrier frequency with an internal reference frequency. The integral of the thereby detected offsets can be taken as an estimator of the clock speed offset. The DW1000 provides these information within the 'Carrier Recovery Integrator' register (DRX_CAR_INT) after a message has been received. DRX_CAR_INT is a 21 bit fixed point real value. The upper four bits represent the integer part of the number, the lower 17 bit the fractional part. One can calculate the absolute offset in Hz following [14]:

$$CRI[Hz] = \frac{C_{int} * 2^{-17}}{2\left(\frac{1024}{998.4*10^6}\right)} \tag{3.19}$$

with C_{int} being the value of the DRX_CAR_INT register. By scaling this value with the carrier frequency F_c one can retrieve the relative speed offset in ppm [14].

$$CRI[ppm] = \frac{CRI[Hz]}{F_c} * -10^6 \tag{3.20}$$

3.4.3.1 Impact of the Response Delay

The response delay is the time interval between the reception of the INIT message and the transmission of the RESP message. Its lower bound is given by the hardware's ability

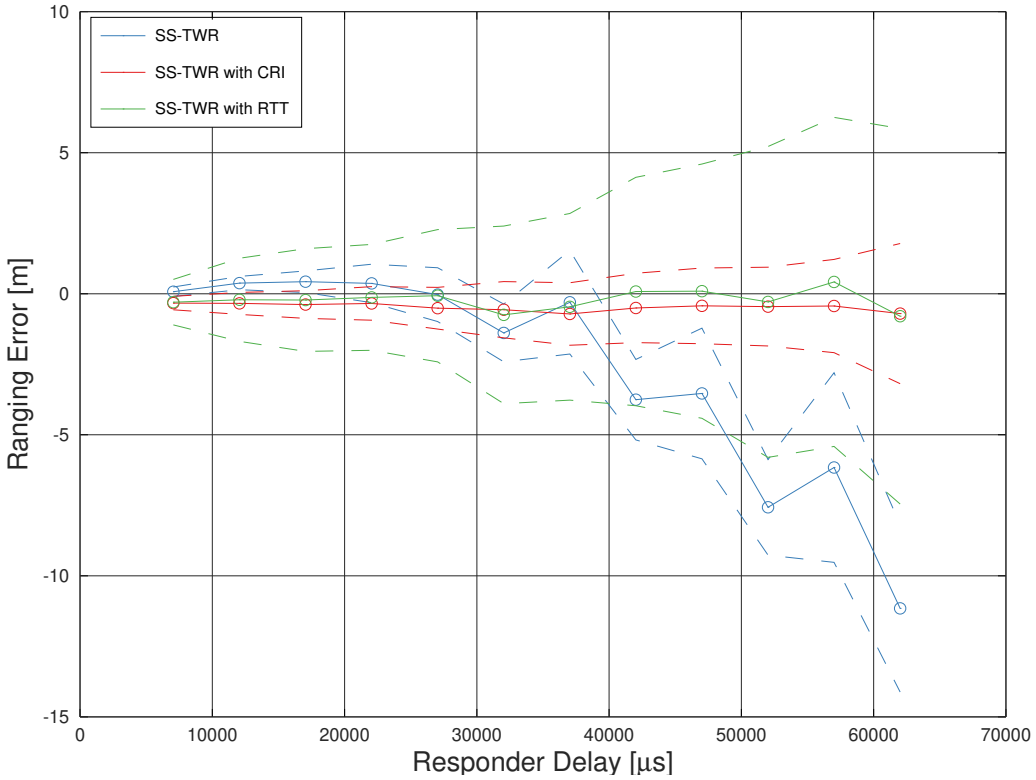
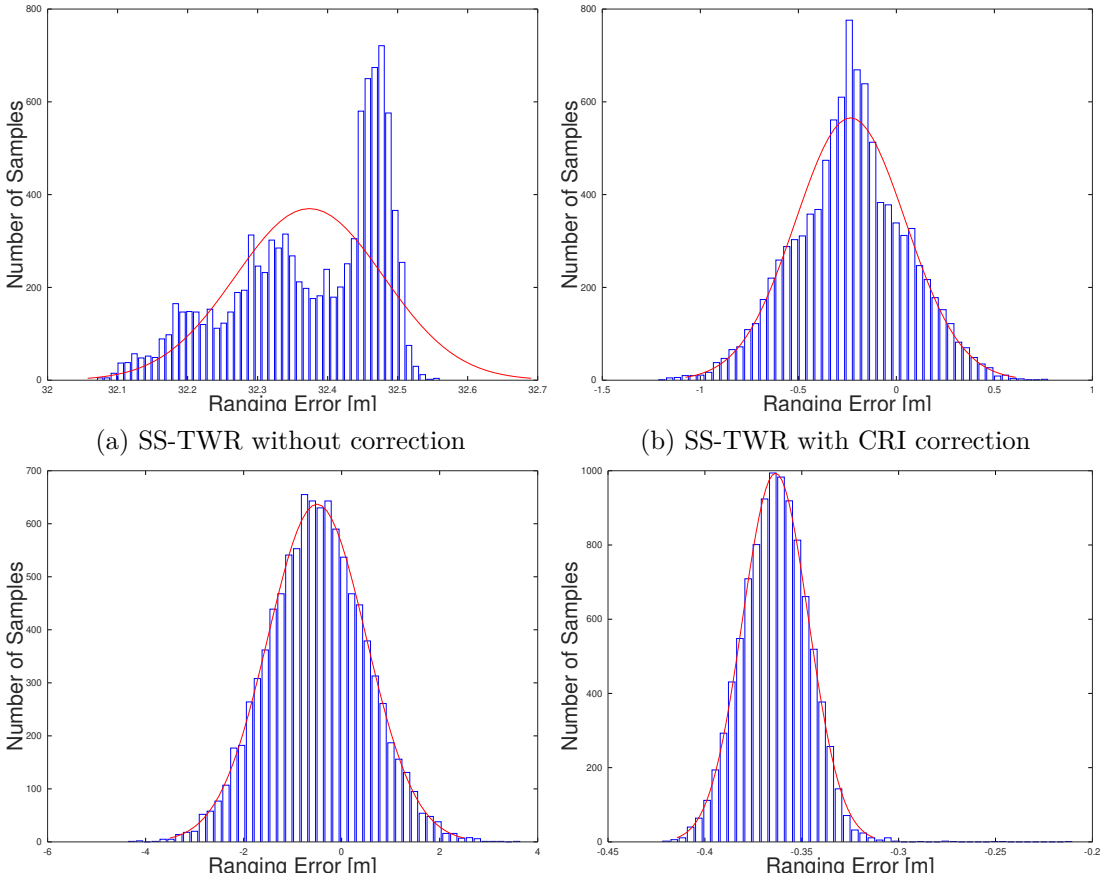


Figure 3.6: SS-TWR ranging error with increasing response delay. Dotted lines show the standard-deviation of the corresponding measurement. The solid lines represent the mean values.

to receive, process and respond to the INIT message. Especially the serial communication between the DW1000 and the microprocessor bounds the delay time in our setup. The response delay has an amplifying effect to the error induced by the clock speed offset (see Section 3.2.2). Therefore, it especially influences SS-TWR. To investigate this influence some measurements have been taken with increasing response delay. In this experiment both the receiver and the transmitter have been calibrated in regard to the clock-trim during the start-up. Therefore, clock speed offset effects should be as minimal as possible. However, a little error still remains. For each evaluation point 1000 estimates have been taken. The mean and the standard deviation of each measurement has been examined and is presented in Figure 3.6. As shown, the accuracy of the uncorrected SS-TWR drops with an increasing delay above 30 ms. This effect is due to the remaining clock speed offset. The accuracy of the corrected variants stays constant. Therefore, it can be seen that a correction of SS-TWR measurements is crucial for increasing delays. Furthermore, the distribution of RTT and CRI corrected measurements can be observed. It seems that both correction values induce some noise to the measurement. The RTT even more than the CRI. This effect will be further examined in Section 3.4.3.2. Furthermore, the deviations increases linear for longer delays.



(a) SS-TWR without correction

(b) SS-TWR with CRI correction

(c) SS-TWR with RTT correction

(d) DS-TWR

Figure 3.7: Ranging distribution of the four variants of two-way ranging. Be aware of the different scales in the graphs.

3.4.3.2 Impact of the Correction Values

Now a closer look at the measurement distribution of our rangings is taken. For the experiment two boards with a clock speed offset of 12 ppm were taken. The response delay has been set (7 ms) to show the impact of the correction values. Both boards have been placed on a 1.5 metre tripod and faced each other over a distance of 10 m in direct line of sight. 10.000 rangings have been performed in four different ways (see Figure 3.7 and Table 3.1): Besides the raw SS-TWR measurements, the CRI and RTT corrected SS-TWR measurements, as well as DS-TWR measurements have been observed for comparison. The results can be summarized as follows:

- due to the large clock speed offset and the higher response delay, SS-TWR without correction results in an unacceptable high loss of accuracy;
- both SS-TWR correction methods vastly improve accuracy as they reduce the error caused by the clock speed offset;
- both SS-TWR correction methods induce noise and thereby decrease precision;

	mean offset [m]	deviation [m]
SS-TWR	32.37	0.10
SS-TWR (RTT)	-0.50	1.00
SS-TWR (CRI)	-0.23	0.28
DS-TWR	-0.36	0.017

Table 3.1: Accuracy and precision values with four different ranging techniques.

- DS-TWR does not further improve accuracy, but greatly improves precision.

3.4.3.3 Impact of the Clock-Trim

In order to investigate the clock speed offset influence on SS-TWR, another experiment has been made. The response delay has been reduced to about 1500 µs and stays constant throughout the experiment. 1500 µs is the shortest possible delay in our system. The anchor has been factory calibrated in terms of the clock-trim. The clock-trim feature of the tag now has been deliberaty used to manipulate its clock speed offset. For each measurement the clock trim value has been increased by one within the ±10 ppm interval. Furthermore, two measurements have been taken with a high and a low clock trim value to show the trend for higher clock speed offsets. Beyond these two trim values the ranging was limited as a increasing number of reception errors has been observed. The Carrier Recovery Integrator (CRI) has been used as an estimator of the current clock speed offset, since the previous experiment indicated its superiority over the Receiver Time Tracking in terms of precision. Over a distance of 5 metres, 1000 rangings have been performed for each trim value. Figure 3.8 shows the outcome of this experiment. The results lead to two conclusions:

1. First, error induced by the clock speed offset of a device is indeed linearly depending from the clock speed offset, as stated in Section 3.2.2.
2. Secondly, both accuracy correction methods (RTT/CRI) are fairly resilient against this source of error.

3.4.4 Clock Speed Offset Mitigation Techniques

Previously the impact of the clock speed offset to SS-TWR has been shown. As can be seen in Section 3.4.3.2, one can choose between high precision (SS-TWR without correction) or high accuracy (SS-TWR with RTT or CRI correction). However, high precision as well as high accuracy is desirable in the context of our localization scenario. This section wants to present a way to achieve both.

Offline clock Calibration. The first, straightforward approach consists in performing an offline factory calibration to all used devices before the ranging. Using a spectrum analyser one could calibrate each device individually and thereby reduce the clock speed offset [13]. However, this approach requires quite a lot of effort and does not adapt to

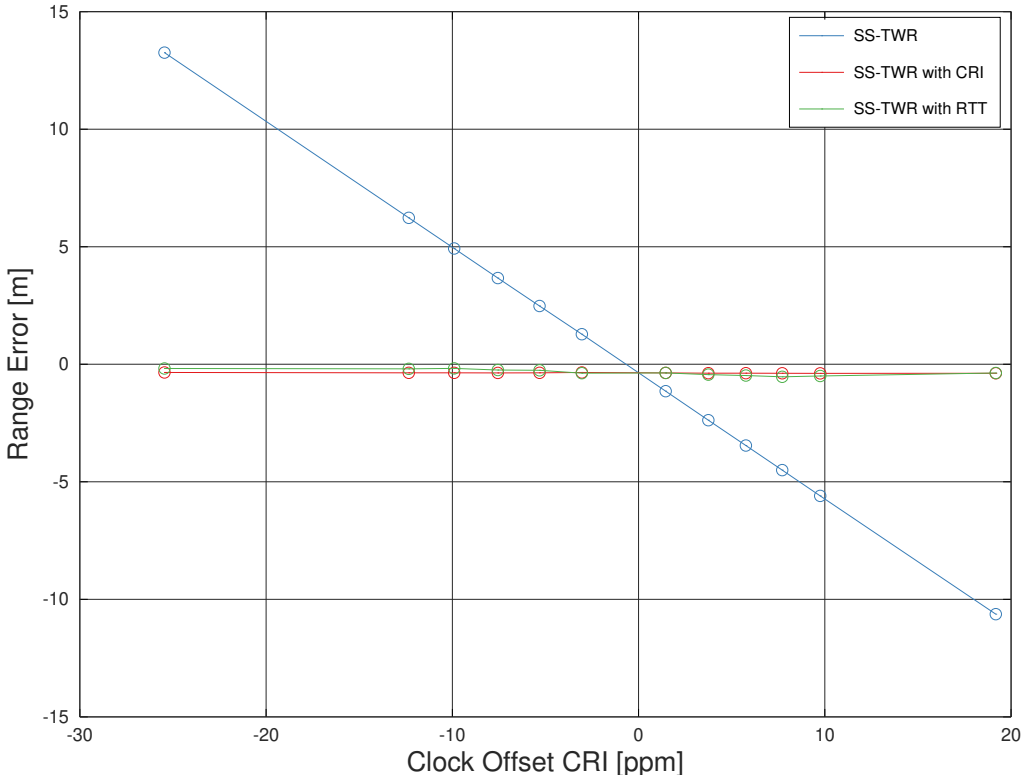


Figure 3.8: Ranging Error with varying clock speed offset.

a changing environment, for example in the case of a change of temperature or other external influences.

Online clock Calibration. The second, more promising approach is an online clock calibration. Such an approach can be done as follows: A new, uncalibrated device will listen to the ongoing communication. When receiving messages from an calibrated device the uncalibrated device will estimate the relative clock speed offset. This can either be done via the CRI or via the RTT. While calibration via the more precise CRI is only supported for coherent receivers such as the DW1000, the RTT value can be used for all IEEE 802.15.4 compatible UWB receivers as this estimation value is mandatory. Given that estimation the device may now perform a clock-trim. If the device is aware of the relation between the own clock-trim value and the relative offset a precise correction value can be computed within a single step. Otherwise it is also possible to converge to the optimal trim value in multiple steps by repeating this procedure several times. An implementation of such an online clock calibration can be seen in Section 4.2.3. Beyond its reduced effort the benefit of the online clock calibration lays in the possibility of the device to adapt to environmental influences at runtime.
For example imagine an RTLS deployed within a building. The system is calibrated and works properly. If someone opens a window and thereby decreases the temperature the affected devices will run at a different speed, thereby increasing their clock speed

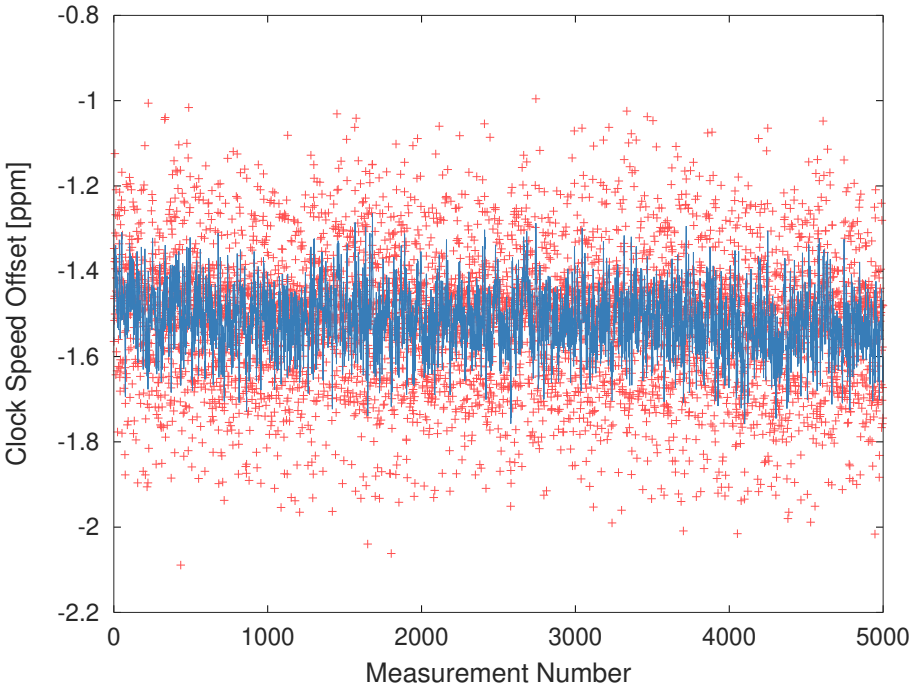
offset in relation to the other (unaffected) devices. The consequence will be an increasing ranging offset and thereby a decreasing localization performance. Having an online calibration mechanism will prevent such a scenario. The affected device will recognize its offset during the reception of the next message of a reference device. Thereby, it has the possibility to adapt to the new environment accordingly.

Applying correction values In order to achieve high accuracy as well as a high precision the noise induced by the correction values (CRI or RTT) must be reduced. To achieve that a simple moving average is sufficient. It smooths the noisy measurements and thereby reduces the overall deviation. Figure 3.9a shows a measurement of 5000 CRI values and their smoothing through an exponential weighted moving average (EWMA) following Hunter [27]. It is a simple method to implement a moving average from a sequence of measurements. Let $\{m_0, ..., m_n\}$ be a sequence of measured values. Let $i \in \{0, ..., n\}$ be the index of the measurement and $\lambda \in [0 : 1]$ the smoothness factor. The average a_i at the time i is defined recursive by

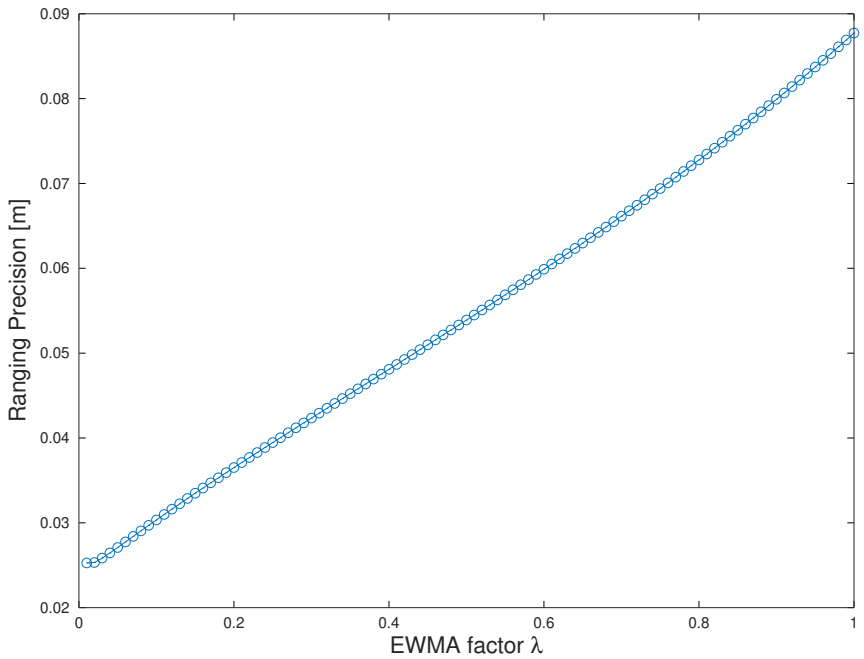
$$a_i := \begin{cases} m_0 & i = 0 \\ m_i\lambda + m_{i-1}(1-\lambda) & \forall i \in [1 : n] \end{cases} \tag{3.21}$$

λ determines the smoothness of the average values a_i. For high values of λ the recent measurements are of higher importance. For lower values the average becomes smoother. The past measurements have a higher impact whereby the deviation is decreased. On the downside this means the system gets less reactive. For example in case of a sudden change of temperature, the system will adapt to this change with a slight delay.

One can see that applying an EWMA to the correction values will reduce the noise and thereby increase the ranging precision. The remaining problem is to choose a appropriate value for λ. Choosing a value is a tradeoff between being up-to-date and ensuring a good adaptability on the one hand, and reducing the noise on the other hand. The aim has been to achieve a comparable precision as the uncorrected SS-TWR. For the chosen response delay of 1500 µs this means a precision of 4 cm. From a measurement of 5000 rangings with the according CRI values the impact of different smoothing factors has been simulated (see Figure 3.9b). As result it can be seen that setting λ to 1/3 seems to be the desired tradeoff.



(a) CRI values (red crosses) with a smoothing (blue line) through an exponentially weighted moving average (EWMA). Smoothing factor λ set to 1/3.



(b) Simulation of the impact of an increasing EWMA factor λ on the ranging precision.

Figure 3.9: EWMA-CRI correction for SS-TWR.

	raw [m]	CRI [m]	RTT [m]	EWMA-CRI ($\lambda = 0.33$) [m]
Scenario 1	-4.27	-0.35	-0.47	-0.36
Scenario 2	0.64	-0.34	-0.29	-0.36
Scenario 3	-0.48	-0.34	-0.36	-0.36

Table 3.2: Accuracy offset of the rangings in metre. 'raw' refers to the offset of the ranging without correction. 'CRI', 'RTT' and 'EWMA-CRI' refers to the rangings with applied correction values.

	raw [m]	CRI [m]	RTT [m]	EWMA-CRI ($\lambda = 0.33$) [m]
Scenario 1	0.074	0.083	0.261	0.040
Scenario 2	0.019	0.088	0.214	0.042
Scenario 3	0.041	0.088	0.215	0.044

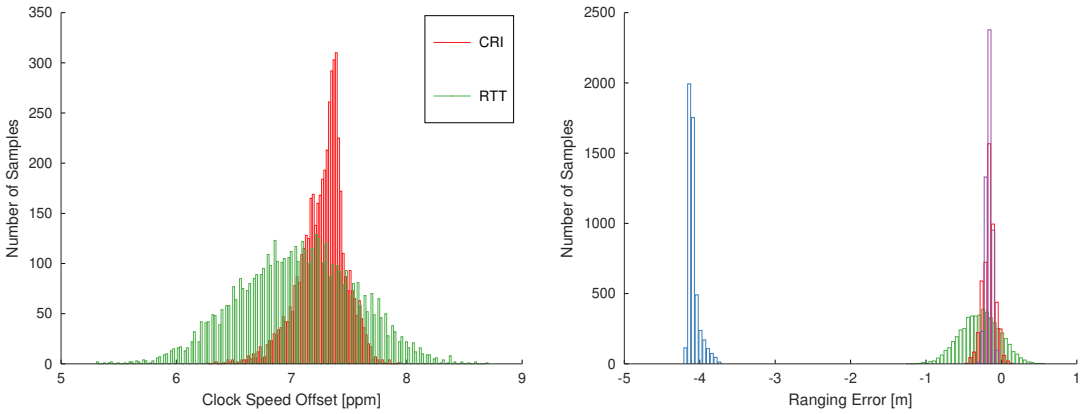
Table 3.3: Precision in terms of the standard deviation in metre.

Evaluation In order to evaluate the performance of the different techniques some rangings that have been taken. 5000 rangings have been taken in three different scenarios. The response delay has been set to 1500 µs at a ranging distance of 7 m. The results can be seen in Figure 3.10. Table 3.2 and 3.3 list the according improvements of the accuracy and the precision.

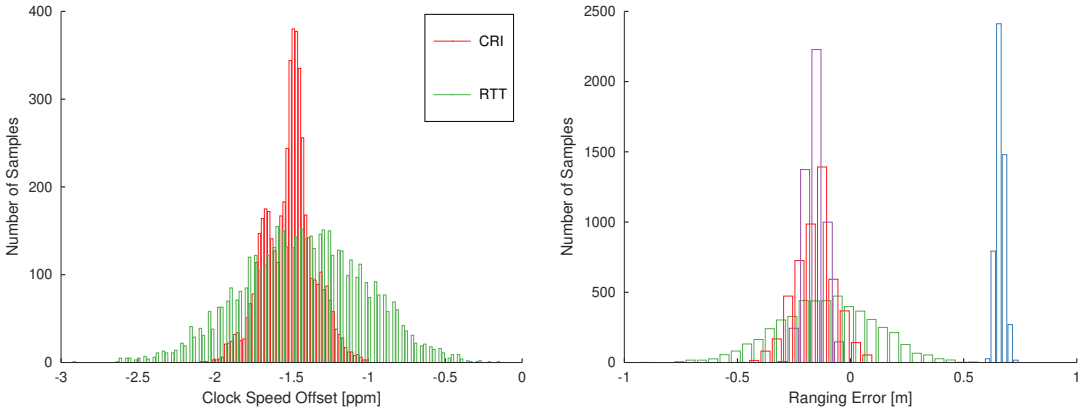
The first scenario has been the ranging between two arbitrary not calibrated devices (see Figure 3.10a). This scenario represents the worst possible case in terms of device calibration. It can be seen that the high clock speed offset of about 7 ppm induced a four metre ranging offset of the uncorrected SS-TWR (blue). The three correction methods RTT (green), CRI (red) and the EWMA-CRI (cyan) all eliminate this offset. Furthermore, the precision increase of the EWMA-CRI can be observed.

In the second scenario (Figure 3.10b) both devices have been calibrated with the boards default values as described in [5]. The proposed clock-trim value is identical for all devices and does not take different device characteristics into account. One can see that the offset of the uncorrected SS-TWR has already decreased vastly. It should be noted that the measured offset varies for different boards, however, this default value gives a good first approximation for most devices.

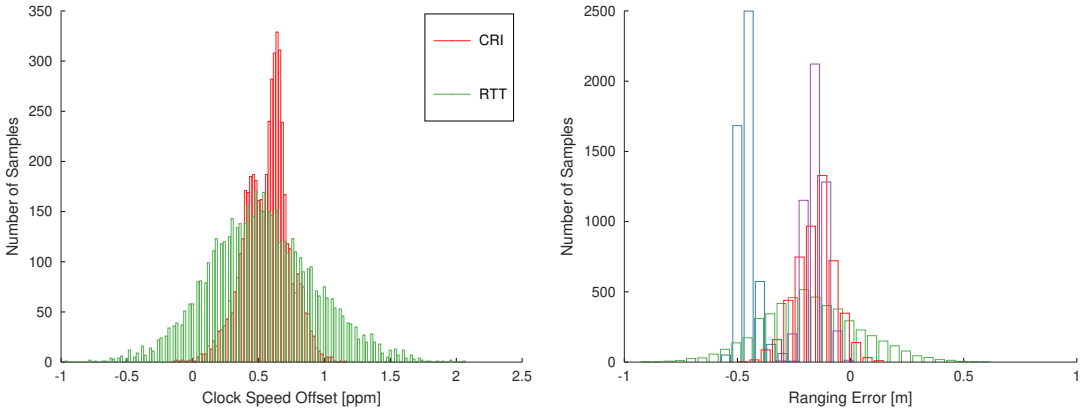
For the third scenario (Figure 3.10c) both - tag and anchor - performed a online calibration with a common reference device as described previously. One can observe that the clock speed offset thereby can be reduced to about 0.5 ppm. Consequential, with this approach the ranging offset of the uncorrected SS-TWR can also be reduced to a minimum. This property is of high importance for weaker, non-coherent UWB devices that for example lack the precise CRI estimation. It can be seen that with performing an online clock calibration these devices can still perform high performance ranging by using uncorrected SS-TWR.



(a) **Scenario 1:** not calibrated nodes with high clock speed offset.



(b) **Scenario 2:** offline clock calibrated with default value according to [5] not taking into account different board characteristics.



(c) **Scenario 3:** online clock calibration via a common reference device.

Figure 3.10: Distribution of the CRI and RTT values (left side) and the according rangings (right side). CRI correction(red), RTT correction(green), without correction(blue) and with EWMA-CRI corretion ($\lambda = 1/3$)(cyan)

3.4.5 Further Influences

This chapter investigates the most important influencing factors on TWR. However, there are more adjustments that could be tested. Only the behaviour on channel 2 with fixed preamble length (128), fixed PRF (64 MHz) and fixed data rate (6.8 Mbit/s) has been investigated. As Malajner et al. [38] indicate, all these variables could be tuned for further improvement of the ranging performance.

3.5 Improving SS-TWR

Previous the major errors have been seen. Now this section sums up the results and shows a step-by-step solution on how to improve SS-TWR. Four steps need to be taken:

STEP 1 antenna calibration (offline) - before the startup of the system the antenna delay should be calibrated. Without calibration offsets up to 30 cm may occur (see Section 3.4.1).
STEP 2 clock calibration (online) - before ranging devices should perform an online clock calibration. This will vastly reduce the clock speed offset error (see Section 3.4.4).
STEP 3 apply correction values - Correction values such as RTT, CRI or EWMA-CRI should be applied according to Equation (3.17). This will correct the remaining clock speed offset error especially for larger response delays as has been seen in Section 3.4.3.1.
STEP 4 apply receive signal strength correction (see Section 3.4.2). Reducing the error due to the received signal strength will eliminate its offset. Otherwise ranging offsets up to 30 cm may occur.

An example of the improvement through calibration and correction can be seen in Figure 3.11 as a cumulative distribution function (CDF), over the absolute ranging error. The ranging measurements have been taken at a distance of three metre. The responder delay has been set to 9 ms. Five different scenarios have been evaluated.

First, two boards have been calibrated with the default values according to [5] (see blue line in Figure 3.11). This scenario corresponds to a ranging before **STEP 2**. The mean clock speed offset among the devices has been 3.42 ppm according to the CRI values. One can see the resulting high mean offset of 6.16 m.

In the next step an online clock-calibration has been performed (see red line in Figure 3.11 and Figure 3.12). This scenario corresponds to a ranging after **STEP 2**. Figure 3.12 shows the probability density functions after the clock-calibration. The mean clock speed offset has been reduced to 0.71 ppm through the online clock calibration. It can be seen that the ranging offset has been vastly reduced to 1.48 m while the deviation remains low.

Next the correction values have been applied (**STEP 3**). The green line in Figure 3.11 and Figure 3.12 shows the impact of the CRI correction. One can see that the CRI

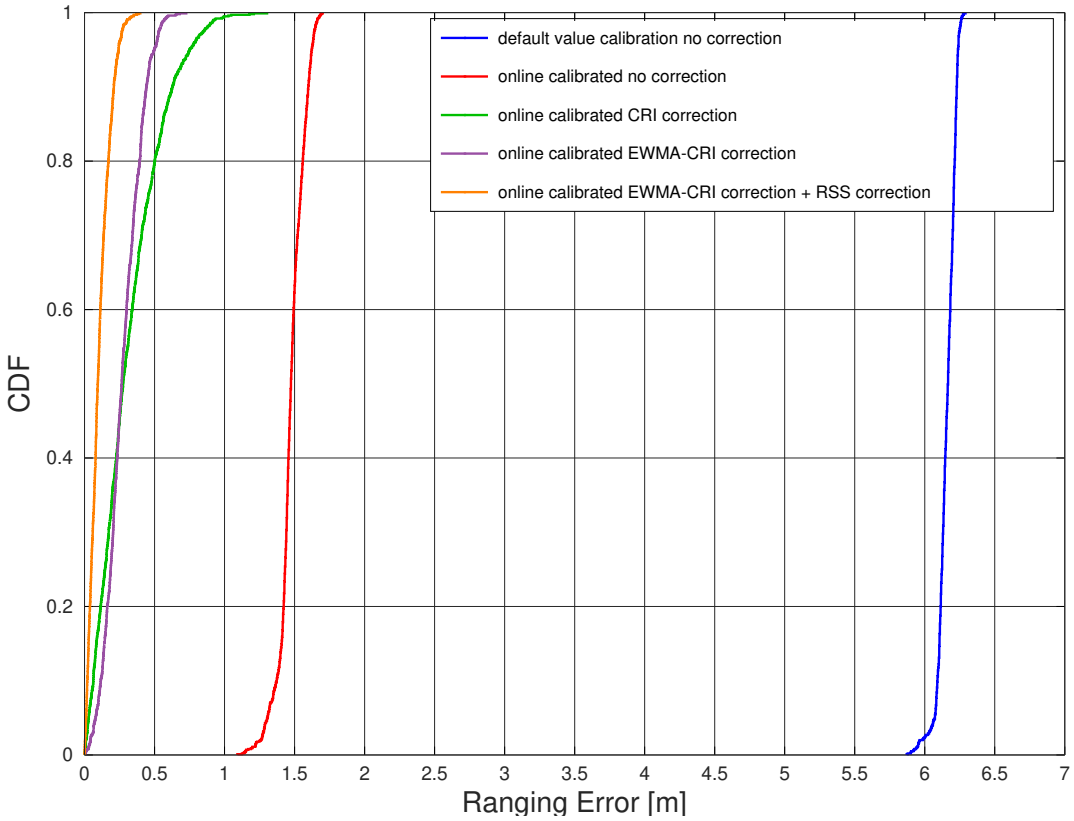


Figure 3.11: Cumulative distribution function (CDF) of the ranging improvement through calibration and correction methods.

correction eliminates the remaining clock speed offset and thereby improves the accuracy. The offset has been reduced to 0.31 m. However, the standard deviation has been increased to 0.22 m as it has also been observed in Section 3.4.3.2. In order to decrease the deviation applying the EWMA filter is applied to the CRI-values. Applying en EMWA-CRI correction value (cyan line in Figure 3.11) eliminates the remaining clock speed offset while keeping the deviation low at 0.12 m.

As a last step the error due to the received signal strength has been corrected according to equation (3.16). This corresponds to **STEP 4** of the proposed step-by-step solution. The orange line in Figure 3.11 and Figure 3.12 shows the resulting improvement. This reduces the remaining offset from 0.28 m to 0.11 m. The remaining error of 0.11 m could be further reduced via an individual antenna delay calibration that has not been performed for this thesis.

Example: Lets take a look at a concrete example. The used values have been taken from a random ranging that has been conducted for Figure 3.11 and Figure 3.12. Lets assume the antenna has been calibrated accordingly during an offline calibration

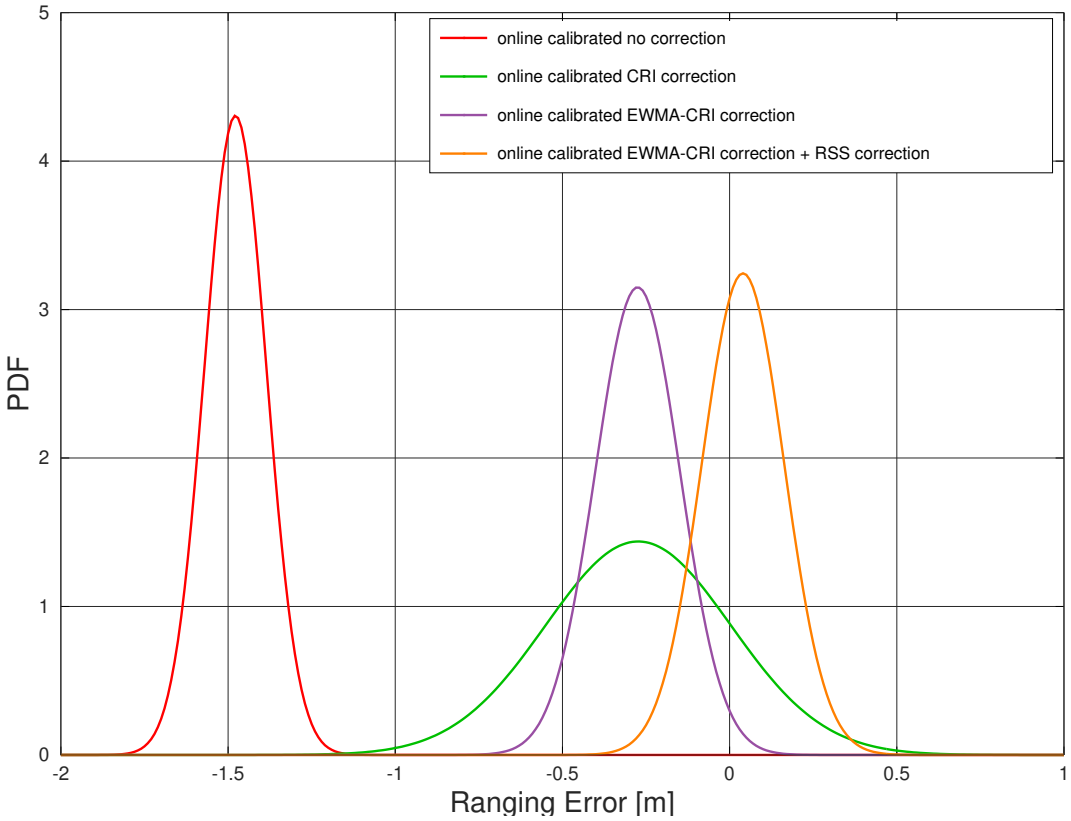


Figure 3.12: Probability density function (PDF) of the ranging improvement after online clock calibration through correction methods.

(**STEP 1**). Further lets assume the clock has been online calibrated according to **STEP 2**. The true ranging distance is 3 m. From our single-sided two-way ranging we have obtained the values for R and D (see Figure 3.1). From Equation (3.1) we know how to calculate the time-of-flight T:

$$T = \frac{1}{2}(R - D)$$

with $R - D = 11.738\,\text{ns}$ we obtain a time-of-flight of 5.8688 ns. Multiplied with the speed of light this would result in a range estimation of 1.76 m. However, we can use the provided correction values to improve the ranging. The EWMA-CRI is reported to be 0.587 64 ppm in this example. Therefore, the corrected time-of-flight (**STEP 3**) following Equation (3.17) can be calculated via:

$$T_{corr} = \frac{1}{2}(R - (D * (1 - 0.58764 * 10^{-6})))$$

So our EWMA-CRI corrected time-of-flight now happens to be 9.1867 ns. By multiplying with the speed of light we obtain a ranging distance of 2.7533 m. We now apply the

received signal strength correction (**STEP 4**) according to Equation (3.16) and get

$$r_{corr} = 2.7533 - (-0.28 + 0.028 * 2.7533) = 2.9562$$

Therefore, the ranging in this example has an offset of just 4.4 cm.

3.6 Lessons Learned

Three major influences to the accuracy and the precision of SS-TWR have been identified.

- antenna delay
- received signal strength
- clock speed offset

Furthermore, the response delay and the clock-trim have been identified as crucial factors influencing the error induced by the clock speed offset. While the antenna delay is a calibration problem that should be solved beforehand with an offline calibration, the other influences are more dynamic and need a continuous observation during ranging. The effect of the received signal strength can easily be corrected via the measured ranging distance or by measuring the signal strength directly. The most complex part is to keep the influence of the clock speed offset under control. While the response delay and its impact can be kept small by an efficient software, the impact of the clock-speed offset is determined by hardware and needs to be handled adequately. It an be seen that the offered correction methods improve the ranging accuracy in exchange of a lower precision. In order to reduce the noise of the clock speed offset estimations, filtering is a viable way to go as has been shown in Section 3.4.4.

Taking into account the antenna delay, the received signal strength and the clock speed offset, the ranging performance in terms of accuracy and precision can be improved vastly. Improving the precision of SS-TWR to achieve a standard deviation of 4 cm is possible (see Table 3.3). Thereby, the performance of SS-TWR is comparable to the performance of DS-TWR.

This proves that high performance ranging is possible even with the reduced effort of SS-TWR. Further, it can be assumed that the evolution of UWB devices is just at its beginning. This raises the hope for even faster, more precise transceivers in the future. While faster devices will improve the ranging performance through shorter response delays, more precise clocks will further reduce the clock speed offset.

4

Real-Time Location System

The previous chapter showed how UWB two-way ranging is done and what ranging results can be expected. This chapter now investigates how to build a localization system upon that. A TWR localization system consists of multiple devices (see Figure 4.1). Anchors at fixed position and with the ability to track time precisely perform two-way rangings with a tag. The tag is able to collect these ranging information and hence can calculate its own position. This position can then be presented at a user interface or wherever the position data is needed.

This chapter is structured as follows. Section 4.1 defines the requirements for a real-time location system (RTLS). Section 4.2 will then show a prototype of a RTLS. Section 4.3 gives an insight of the system implementation. A brief overview of other commercial and academical RTLS will be given in Section 4.4.

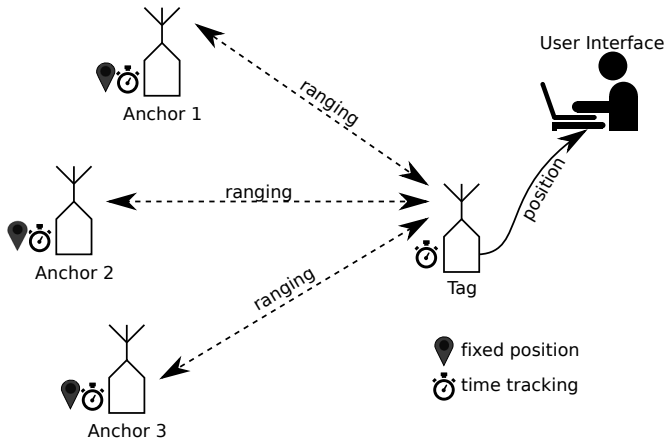


Figure 4.1: Overview of a UWB-TWR RTLS. Anchors with fixed (and known) position help a mobile tag to localize itself by measuring the round trip time of a TWR handshake.

4.1 Requirements

There are numerous different UWB based RTLS solutions currently available both commercial and academical (see Section 4.4). They tackle different problems and have

different intended use cases. Their requirements thereby differ vastly. A localization system localizing a single drone within a building might require a high refresh rate and high accuracy whereas the localization of customers within a supermarket might put the focus on localizing as many devices as possible and thereby neglect accuracy and refresh rate to a certain extend. However, some challenges can be generalized and account for almost all real-time location systems.

Ranging and Localization Performance. The localization performance is the most obvious criteria for a successful RTLS. Especially in the context of indoor navigation a high precision is of fundamental importance. In a dense indoor scenario even centimetres will make the difference between hitting or avoiding an obstacle. To achieve a high localization performance highly accurate distance estimations are key.

Multi-Tag Support. The ability to allow an increasing number of devices to be located is a desirable property for most RTLS. Often not only the position of a single device is important but also the position in relation to other devices. The coordination of multiple devices thereby is an important aim.

Refresh Rate. Especially for mobile devices the refresh rate of a localization is key. For example a drone depending on a quick localization for navigation will hit obstacles if the position cannot be estimated quick enough. Parameters such as the velocity can only be derived from multiple dense measurements. Therefore, a high refresh rate is highly beneficial for mobile devices.

Setup Complexity. The setup complexity is a crucial topic for distributed systems. A complex setup hinders the wide distribution of the system and thereby decreases the availability and performance of a system. In the context of a RTLS, high complexity might be rooted in the need for additional infrastructure such as an ethernet backbone. A low complexity ensures the ability to distribute the system even in complex areas with limited infrastructure.

Resilience. Resilience is an important feature for cooperating devices. It describes how well the failure of system components can be compensated by the overall system. Especially in the context of cheap, mobile, battery powered embedded devices resilience is important as the reliability of single device can be problematic. Single devices might get interrupted by environmental influences, need to power down because of energy restrictions or simply move out of range. In order to increase the resilience of a RTLS the localization shall not depend on single devices (e.g. a central coordinator), a centralized infrastructure should be avoided, and the ability to compensate failures should be embraced.

4.2 Real-Time Location System Prototype

This section presents a prototype for a RTLS. Devices can take up two different roles, namely anchor and tag. Anchors are devices with known position. They serve the

purpose of being ranging partners to other devices with unknown position so called tags. The distinction between tag and anchor is not strict. Thereby tags can become anchors after successfully localizing themselves.

The main aim of the system is to allow tags to perform rangings to arbitrarily many anchors within a single ranging attempt. Furthermore, the system shall allow the operation of multiple tags within the same environment. In order to keep the setup simple, the system should be self-organizing and decentralized. Therefore, the overhead for new devices joining or leaving the network is kept small.

The following section is structured as follows: Section 4.2.1 will define the used UWB channel settings. Section 4.2.2 to 4.2.7 give an overview of the prototype system. Section 4.2.8 and 4.2.9 take a look at further important considerations and decisions one has to take to achieve an accurate localization. Section 4.2.10 summarises the needed messages and declares their structure.

4.2.1 Channel Settings

The system consists of a main channel and multiple side channels. In this context, a channel corresponds to the definition of complex channel of the IEEE 802.15.4 standard as described in Section 2.1. This means a channel is determined by the physical channel and a preamble code. The main channel is chosen to be channel 2 with preamble code 9. Besides the main channel there are three further side channels with the preamble symbols 10, 11 and 12. The only use of the side channel is the exchange of ranging responses during the ranging (see Section 4.2.4).
Besides the channel, the PRF is set to 64 MHz with a preamble length of 128 and a data rate of 6.8 Mbit/s. These settings have been chosen to match the settings taken in Chapter 3 in order to keep the ranging performance consistent. However, the functionality of the prototype is not bound to these settings and could also be set to different channels, pulse rates, and preambles.

4.2.2 Network Discovery

The network exploration is done via the exchange of beacon BECN messages (see Section 4.2.10). Anchors broadcast these beacons periodically. The interval between two beacons can be scaled dynamical depending on the current occupation rate of the main channel (see Section 4.2.7). The beacon message contains the following information:

- *next* - interval until next beacon message
- *load* - current load estimation of the main channel (see Section 4.2.7)
- *position* - current position of the device

The beacon messages will be used to set up a device table. Each device within the network maintains its own device table containing information about all active devices in the surrounding. The table entries contain the following information:

- device id

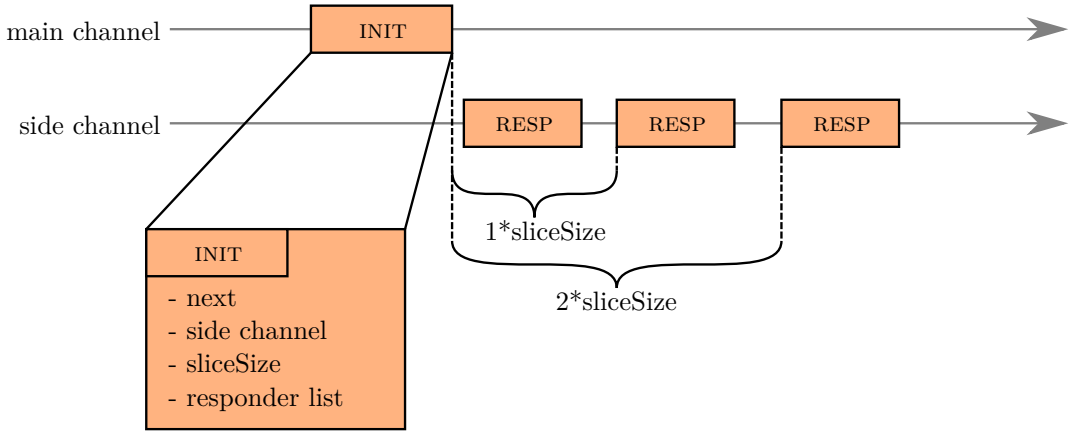


Figure 4.2: Example of a single ranging attempt. The declared slice size and the order within the *responder list* within the INIT message determine the timing of the RESP messages.

- device position (if known)
- last update (local time)
- next update (local time)
- clock speed offset (CRI/RTT)
- device load

To ensure the freshness of the table, it is sufficient to erase devices that are past due with a certain threshold. For example anchors that have not been heard for 20 s (twice the maximum beacon interval) can be deleted.

4.2.3 Clock Speed Calibration

To support the usage of non-offline calibrated devices the network enables devices to calibrate with established reference devices. Calibrated reference devices transmit a special type of beacon message - the clock calibration message (CCRN). These messages behave the same as normal beacon messages containing the same information, however, differ in the type (see Section 4.2.10). On reception of a CCRN message uncalibrated devices can determine their clock speed offset in relation to the calibrated device either by the use of the CRI or RTT (see Section 3.4.3). By adapting their own clock-trim value accordingly, a calibration can be performed.

4.2.4 Dynamic Ranging

The performed rangings are single-sided two-way rangings consisting of two types of messages (see Figure 4.2) as SS-TWR proved to produce precise rangings (see Section 3.6). The ranging will be started by the tag with a broadcast of an INIT message on the main channel. The message contains a list of desired anchors (responder list) as well

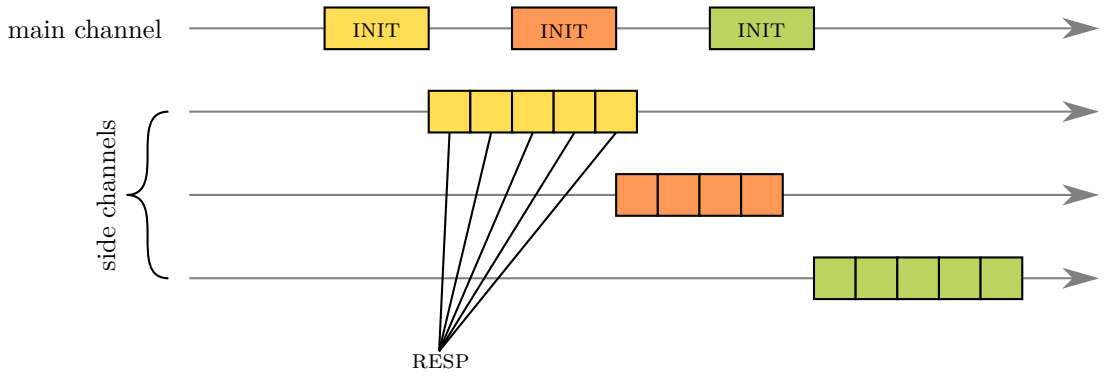


Figure 4.3: Ranging with usage of side channels. INIT messages will be transmitted on the main channel. According responses will be transmitted on a different complex channel specified by the preamble code declared in the INIT message. Multiple rangings can be performed in parallel, if sufficiently many different anchors are available.

as a slice size, indicating when the responses are expected. The placement within the list as well as the slice size, indicates each anchor when his response is to be expected. The expected response delay can simply be calculated by

$$delay[\mu s] = s[\mu s] * i \tag{4.1}$$

with s being the slice size and i being the index within the responder list. Within their reserved interval the anchors answer with their RESP message, containing only the measured delay between reception of the INIT and transmission of the RESP message (see Section 4.2.10). In order to avoid conflicts with other devices tags can specify a preamble code for the responses and thereby declare a side channel. Hence, they will be sent on a different complex channel and will not occupy the main channel. Several different ranings can thereby be performed even in parallel using different complex channels. Of course this implies that sufficiently many anchors must be available. After the responses have been transmitted the anchors switch back to the main channel waiting for new ranging attempts.
If a tag wants to perform rangings periodically, it can declare its interval in the INIT message within the 'next' variable. Thereby, listening anchors and tags can take this information into account and schedule their own activities accordingly. To avoid collisions with other devices the tag shall check its device table (see Section 4.2.2) before sending the INIT.

4.2.5 Network Management - Entering/Exiting the Network

The proposed system is a very loose network with very little limitations and restrictions. Therefore, adding new devices, removing devices or suspending devices need little to no management. For entering the network it is sufficient for new devices to listen to the ongoing traffic. Within 10 s (maximum interval for beacon messages), all surrounding potential ranging partners will be detected.

If a device wishes to exit the network, it can either go offline and wait until the other devices clear the entry off their device tables or actively indicate its exit by sending a BECN message with the *next* field set to zero. This will indicate the recipients that the device can be erased from the device table.

4.2.6 Media Access Control

In the field of narrowband communication simultaneous transmission often pose a huge risk. If two devices transmit at the same time, it is very likely that the messages get corrupted. Due to the nature of UWB communication, transmissions are very robust against multi-user interferences. The time-hopping mechanism of the BPM-BPSK modulation as well the usage of large guard intervals (see Section 2.1.2) reduce the probability of collisions. Further, the forward error correction (FEC) feature of the DW1000 increases its robustness [10].

However, even if simultaneous UWB transmissions are less effected by multi-user interference, the problem of colliding transmissions still needs to be tackled. If two or more transmission are ongoing simultaneously only one will be received. The other transmissions vanish unnoticed and neither receiver nor transmitter have a chance to detect this conflict. In the worst case, both messages will destroy each other. Furthermore, the device needs to accommodate the fact that all devices need some time to process received messages and therefore, be deaf for a short amount of time after each message. Therefore, media access control is still crucial for UWB systems.

In the prototype the crucial part is the ranging. Neither the INIT nor the responses should get lost. On the other hand, a loss of single beacon messages is acceptable as they are sent periodically. By listening to the ongoing traffic, a lot of future traffic can be predicted. However, these observations cannot give a complete image of the whole system. Therefore, collisions still may occur e.g., by a hidden terminal. Even if the loss of ranging messages cannot be prevented in all cases, the tag gets an instant feedback and can accommodate accordingly. Following feedbacks may occur:

- tag receives all requested responses ⇒ everything fine
- tag receives no responses ⇒ supposable conflict of INIT message.
- tag receives few responses ⇒ multiple causes:
    1. some anchors might be out of range
    2. some anchors might be occupied by other devices
    3. slice size might be chosen too small, therefore, responses overlap each other (especially if every second response is missing)

The tag can thereby react accordingly. If all expected responses arrived no further action is needed. The tag can proceed by calculating its position. If no responses arrived, it is very likely that there has been an transmission right before the INIT message. If at least some responses arrived, the tag has multiple options. If sufficient responses arrived for localization, a position should be calculated. If there have been

too few responses, or a higher accuracy is desired for localization, the tag may retry to range with the missing anchors.

4.2.7 Load Balancing

As seen in Section 2.1.4, keeping the channel occupation below 18 % ensures a high chance of conflict free transmissions. Here, load balancing is in the responsibility of the anchors. The aim is to provide as much time as possible for ranging attempts. The exchange of beacons is declared less important. Further, it can be assumed that the transmission of a single message takes between 1 ms to 3 ms with the proposed channel settings. In order to measure the current load of the channel, the anchors keep track of how many beacon messages or ranging attempts have been ongoing since the last transmission of its own beacon message. Just before transmitting the next beacon, the channel occupation L can be calculated by

$$L = \frac{n * t_{avg}[ms]}{t_{int}[ms]} \tag{4.2}$$

with n being the number of overall transmissions, t_{avg} being the average length of a transmission (e.g., 3 ms) and t_{int} being the length of the measurement interval. If the load is below 15 %, the beacon interval will be decreased by 10 ms. If the load is above 20 %, the beacon interval will be increased by 100 ms. In order to prevent a spamming of beacons in case of low channel occupation, there is a lower bound of 100 ms for the interval length. On the other hand, in order to ensure the freshness of the network the upper bound is set to 10 000 ms.

4.2.8 Ranging Considerations

A tag has some degree of freedom on how it attempts to range within the system. Following some considerations that should be taken into account when initiating a ranging are discussed.

There are two options on how to arrange rangings to multiple anchors. The first option is the standard mechanism, i.e., the tag ranges with multiple anchors in a single ranging attempt. It should be mentioned that precision will decrease for higher response delays (see Section 3.4.3.1). Delays above 20 ms should be avoided. Rangings with more than ten anchors should thereby be split into multiple separate rangings.
As second option the protocol also allows the individual ranging with each anchor. Thereby, the tag sends an INIT message only containing one anchor. The idea behind this approach is to benefit from the short response delays when there is only one anchor. Therefore, the gathered rangings are more accurate (see Section 3.4.3.1), which might improve the localization performance. However, this approach comes with a significant penalty. First, the overall ranging procedure takes much longer, resulting in a lower refresh rate of the tag position. Further, obviously this method stresses the main channel at a much higher rate, increasing the load and might even prevent the ranging of other devices. This method therefore should only be used within low utilized environments

where multi-user support is not the highest priority.

In order to minimize the slice size, it might be advisable to leave the first slot of the *responder list* empty. Anchors have a minimum achievable response delay, bounded by the SPI speed and the speed of the microprocessor. The minimal response delay in our implementation is around 1.5 ms. The first anchor cannot provide the response fast enough to fit in the first slice, if it is chosen too small. On the other hand, larger slice sizes might be unnecessary for the later anchors, as they can prepare their messages beforehand anyway. Therefore, leaving the first slot empty might decrease the ranging delay for the later anchors, as well as decrease the overall time of a ranging attempt.

4.2.9 Anchor Selection

Special care should be taken when selecting appropriate anchors. If a position can be estimated (e.g., from former localization attempts) anchors can be chosen by their relative position. Being within the convex hull of the anchors benefits most localization algorithms (see Chapter 2.2). Also the distance can be taken as criteria. As can be seen in Section 3.4.2, the received signal strength poses a high threat to the accuracy of the ranging measurement. By selecting anchors that are assumed to be further away (in our case at least 10 m) this influence can be mitigated. Anchors with a distance greater than 50 m should also be avoided, as the ranging accuracy will decrease also for large distances as discussed in Section 3.4.2. Further, when choosing anchors one should also take care of their clock speed offset. High CRI and RTT values may indicate not calibrated devices thus leading to inaccurate rangings (see Section 3.4.3). Choosing devices with low clock speed offset will improve localization results. Last but not least, one should also take into account the anchors' load as using such anchors increase the risk of incomplete rangings as the probability of conflicts rises.

4.2.10 Messages

A message consists of a standardized header and a payload. The header includes the sender (FROM) and the recipient (TO) id as well as a sequence number (SEQ) and a type (TYPE) (see Table 4.1). The payload can be data up to 127 bytes.

	Header						**Payload**
Byte	0	1	2	3	4	5	6 - 127
Name	TO		FROM		SEQ	TYPE	

Table 4.1: Frame Structure.

TO Contains the 16-bit recipient address. Messages with the recipient id $0xFFFF$ are considered to be a broadcast message.

FROM Contains the 16-bit transmitter address.

SEQ A 8-bit sequence number.

Type Identifies the type of the Frame (see Table 4.2).

Name	Type	Value
beacon	BECN	0xA1
clock-calibration	CCRN	0xA2
range init	INIT	0xB1
range resp	RESP	0xB2

Table 4.2: Frame Types.

Beacon. The BECN messages (see Table 4.3) is a broadcast message serving multiple purposes. First, it signals other devices the availability as a potential anchor. Second, it publishes the own position and status information. Therefore, other devices can set up a device table containing potential anchors. The NEXT section indicates the time interval after which the next beacon message of the device is to be expected. LOAD corresponds to the result of the load balancing calculation (see Equation 4.2) given in percent. The position coordinates are given as IEEE 754 single-precision floating point values.

Byte	0 - 5
Name	Header

TO Broadcast
FROM DeviceID
TYPE 0xA1

Byte	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Name	NEXT		LOAD		X-COORD				Y-COORD				Z-COORD			

Table 4.3: Structure of a BECN message.

Clock-calibration. The Clock-Calibration Message CCRN (see Table 4.3) is - apart of the type - identical to the BECN message. However, it must only be used by factory clock-calibrated devices. Other, uncalibrated devices can thereby compare the clock skew between their receiver and the calibrated transmitter by calculating the Channel Response Integrator or the Receiver Time Tracking (see Section 3.4.3) and calibrate its clock.

Byte	0 - 5
Name	Header

TO Broadcast
FROM DeviceID
TYPE 0xA2

Byte	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Name	NEXT		LOAD		X-COORD				Y-COORD				Z-COORD			

Table 4.4: Structure of a CCRN message.

Range init. The INIT message (see Table 4.5) initializes the rangings. NEXT is a 16 bit unsigned integer indicating when the next ranging of the device is to be expected

in ms. Can be set to 0 for single one time ranging. RESP PSYM refers to the preamble symbol under which the responses are expected. It thereby defines the side channel for this ranging (see Section 4.2.4). The SLICE SIZE is a 16 bit unsigned integer, indicating the duration of the time interval in µs, that each anchor has for sending its reply. The slice size must be chosen large enough so that the tag has enough time to process the incoming responses before receiving the next one. On the other hand, the slice size should not be chosen too large in order to keep delay times small and the main channel free. The *responder list* contains the 16 bit device IDs of the desired anchors. The number of anchors in the list is not limited, however one should keep the increasing response delay in mind.

Byte	0 - 5
Name	Header

TO Broadcast
FROM Tag ID
TYPE 0xB1

Byte	6	7	8	9	10	11	12	13	14	···
Name	NEXT		RESP PSYM	SLICE SICE		RESP 1		RESP 2		···

Table 4.5: Structure of a INIT message.

Range resp. The RESP message (see Table 4.6) finalizes the ranging between the tag and the anchor. It contains the measured delay time in µs between the reception of the INIT message and the transmission of the RESP message. This value corresponds to the value D - the difference of the *rx_init* and the *tx_resp* timestamp - in Figure 3.1.

Byte	0 - 5
Name	Header

TO Tag ID
FROM Anchor ID
TYPE 0xB2

Byte	6	7	8	9	10
Name	DELAY				

Table 4.6: Structure of a RESP message.

```
- build          contains the compiled *.hex *.elf files
- doc            contains a doxygen documentation
- src            source directory containing all needed source code
  - board        board specific files provided by STM
  - com          serial device driver (SPI,I2C,UART)
  - dw1000       dw1000 device driver
  - loc          RTLS implementation
  - math         mathematical functions
  - sensors      sensor drivers
  - utils        miscellaneous
  main.c
  system_it.c    interrupt file, contains low-level callback functions
  global_dev.h   global device handles
- target         MCU driver provided by STM
  - STM32L1xx
  - STM32L4xx
startup_xxx.s    startup assambler file
Makefile
```

Figure 4.4: Overview of the file structure.

4.3 Implementation

Aim of the implementation has been to create a bare metal software solution for the nodes, allowing the usage of different Nucleo boards. The bare metal approach reduces the memory requirements by only implementing the necessary functionality as well as allowing full, direct control of the underlying hardware. The file structure of the project is briefly shown in Figure 4.4. An overview of the implemented software stack can be seen in Figure 4.5.

4.3.1 Toolchain and Build System

The aim of the build system was to keep it simple and platform independent. It consists of a modified version of an already existing makefile [37]. Modifications have been made to support different Nucleo boards and to simplify the programming process. By declaring the device type in the makefile it is possible to pass the type of the current build target to the compiler, which treats it like a common preprocessor macro during compilation. Thereby, within the C source code it is easily possible to retrieve the current build target by using normal preprocessor directives. This simplifies the creation of specific code for each build target. The adaption of the required build paths is also managed automatically via this macro. Further, STM provides an assembler file that needs to be compiled together with the C code. It sets the memory layout and other low-level basic functions. The startup file is platform dependant. Therefore, the Makefile needs to ensure that the correct file for each build target is loaded.
For compilation the open source ARM GCC has been used. The programming of the Nucleo boards has been done with the open source tool openocd and ST-Link and can be utilized via build targets in the makefile.

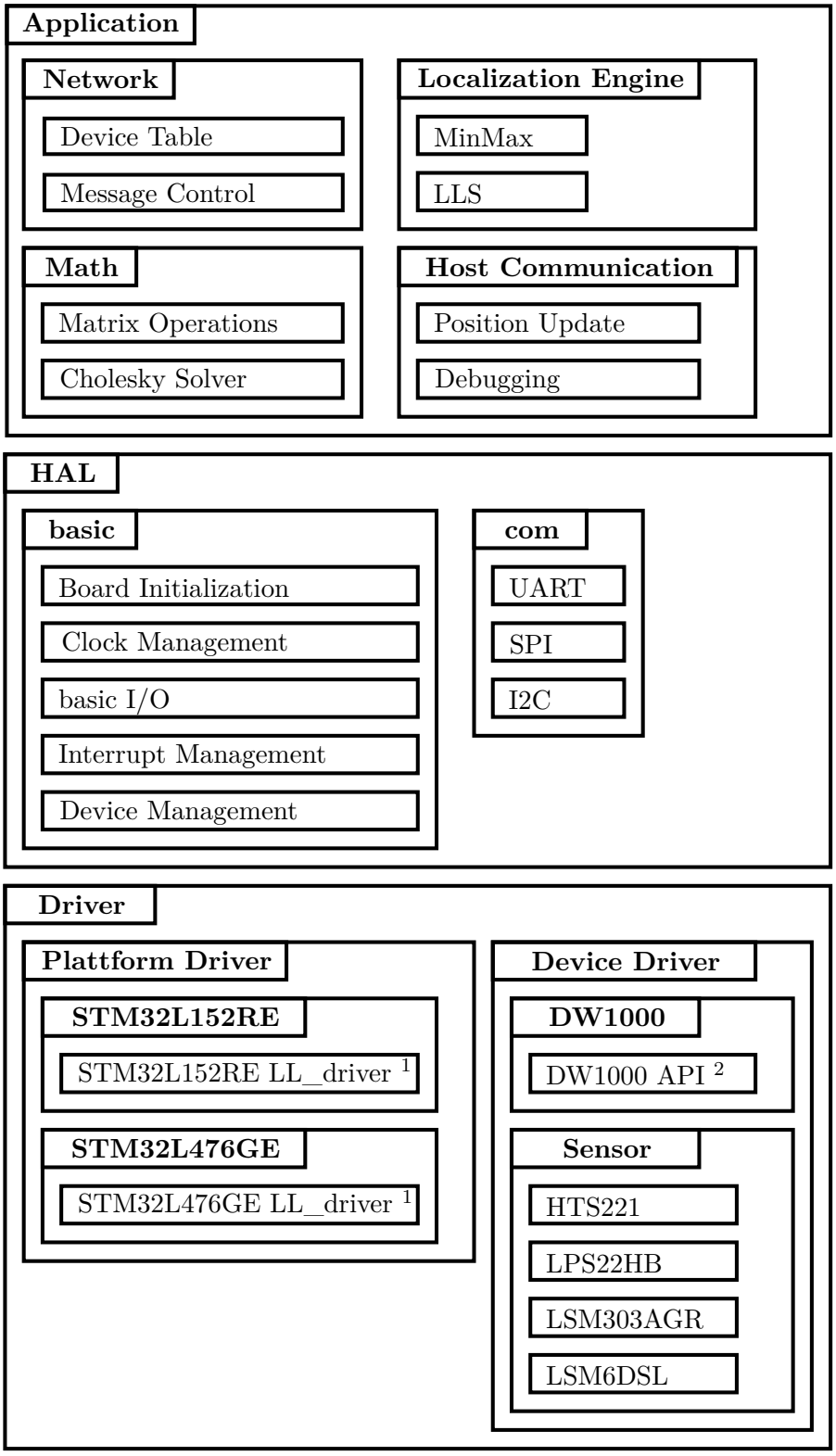


Figure 4.5: Software Stack of the UWB Node. Parts marked with [1] refer to the API provided by STM [55] Parts marked with [2] refer to the API provided by Decawave [16].

4.3.2 Drivers

The drivers allow access to the hardware functionality. The Nucleo boards provide drivers from the vendor, allowing basic register access and access to the core functionality [55]. The functionality of these drivers has been extended in order to simplify the usage of the board functionality for the higher abstraction layers.

DW1000 Driver Integration. Decawave provides an API for their DW1000 module, however, these functions needed to be integrated in the system. Therefore, the implemented driver has to specify all used pin connections and create a link to the implemented SPI driver and the DW1000 API. Furthermore, some additional functionality has been added for example to allow access to the RTT values, or to hard reset or wakeup the DW1000 via the pin connections. Special care has been taken to avoid the slow serial communication via SPI. Therefore, the DW1000 operates fully interrupt-based to achieve low reaction times.

Serial Driver. The serial devices are a key component of the node. This implementation features support for SPI, I2C and UART communication. Peripheral devices such as the DW1000 or Sensors are connected to the MCU via serial connections. Further, the communication to the host computer has been done via a serial UART connection. For the proper function of the node an efficient implementation is needed. Therefore, all drivers have been implemented interrupt based. This ensures that the MCU is not hindered by the comparable slow serial communication. The implementation can be found in the COM/ directory.

4.3.3 Hardware Abstraction Layer

The hardware abstraction layer (HAL) shall hide the complexity of device management and allow easy access to the needed functionality. The HAL manages basic tasks such as the clock management, interrupt handling and access to basic input and output operations.

Hardware Initialization. During the node start-up several steps have to be taken. The *init()* function implemented in the MAIN.C file. It is the first function that will be called and is responsible for the initialization of all needed hardware. It first has to perform the clock-management of the MCU itself. Therefore, the *SystemClock_Config()* function in the UTILS/CLOCK.H file is called. This function sets the frequency of the MCU and configures the system clock. Further, the different clocks for the components (e.g., serial devices, GPIO ...) have to be activated. Afterwards, the initialization functions for the different components will be invoked.

Device Management. The *init()* function triggers the initialization of the used components. All implemented components contain a handler that allows access to the current state of the component. For example the handler of the SPI contains the reception and transmission buffer as well as information about the speed and the used

```
#define SALP_MAXFRAMESIZE   128
typedef struct salp_handle{
  dw1000_handle*  dw1000;
  uint16_t        id;
  float           pos[3];
  salp_devTable   devTab;
  uint8_t         bool_calibrated;
  uint8_t         bool_rxReceived;
  uint8_t         bool_txCompleted;
  uint8_t         rx_frame[SALP_MAXFRAMESIZE];
  uint64_t        rx_frame_ts;
  size_t          rx_frame_size;
  double          rx_frame_cri_ppm;
  double          rx_frame_rtt_ppm;
  uint32_t        next_beacon_tick;
  uint8_t         seqn;
} salp_handle;
```

Figure 4.6: Protocol handle: C-struct containing all relevant data for the protocol execution. (File: LOC/SALP.H).

hardware pins. All handles are stored in the GLOBAL_DEV.H file, to allow easy, systemwide access to the components.

4.3.4 Real-Time Location System

The following section gives an overview of the implementation of the RTLS prototype described in Chapter 4.

Message Reception. The reception of a message consists of several parts. Their implementation can be found in the file LOC/SALP.H. If the DW1000 receives a message the device will invoke an interrupt. This interrupt will invoke the execution of the *salp_rxGoodFrameCallback()* callback function. In order to reduce the number of needed SPI transmissions the callback function first just downloads the message header from the DW1000. By checking the recipient address of the header it can be seen if the message is relevant for the device. In case the message is addressed to the device the message will be downloaded to the *rx_frame* buffer within the protocol handle (see Figure 4.6). After downloading the message, the callback function gets the RTT and CRI value of the message as well as the receiver timestamp. All information will be stored in the protocol handle until further processing.

Network Management. The network management is achieved via two very similar functions within the LOC/SALP.H. *salp_run_passive()* and *salp_run_active()*.

```
status salp_run_passive(const uint32_t timeout_ms);
status salp_run_active(const uint32_t timeout_ms);
```

```
typedef struct salp_devTable_entry{
  uint8_t bool_vld;
  uint16_t devID;
  float pos[3];
  uint32_t  update_tick;
  uint32_t  next_tick;
  double cri;
  double rtt;
} salp_devTable_entry;
```

Figure 4.7: Structure of a device table entry (File: LOC/SALP_DEVTABLE.H).

As parameter the functions take an integer value defining how long (value in ms) the functions should run. Both functions listen for ongoing traffic and update the own device table accordingly. They track the position of the other devices as well as the clock speed offset and the update values. Furthermore, during execution of these functions the device will listen for CCRN-messages and perform online clock calibration (see Section 4.3.4). However, the active variant also broadcasts BECN-messages. This variant should be taken for anchors. The passive variant is suited for new devices or tags that just want to listen and observe the network.

Device Table. The device table fulfils a key role in the RTLS system. It saves the collected information about the surrounding infrastructure. It is implemented as an array of entry elements. The structure of an entry can be seen in Figure 4.7. The application can check the table in order to select appropriate anchors for the ranging. Further, the device table can be checked before the transmission of a message to avoid some collisions. The 'next_tick' field tells when the next message from a device is expected. The 'update_tick' field indicates when the last message has been received. This is useful to verify the freshness of the data. Furthermore, an entry contains information that will be required during the localization. It contains the exponentially weighted CRI value that is needed for the clock speed offset correction and the reported position of the devices.

Online Clock Calibration. During the execution of the *salp_run_passive()* (respectively *salp_run_active()*) function the device listens for CCRN-messages. On reception of such a message the devices evaluate the clock speed offset to the transmitter. This can be done either via the CRI or via the RTT value. As the DW1000 provides the more precise CRI estimations they have been taken for this implementation. If the device knows the clock speed offset, it can calibrate itself by performing a clock-trim as described in Section 4.2.3.
For the DW1000 the trim value r_{new} can be calculated using the estimated clock skew via the CRI (in ppm) and the current trim value r_{cur} by calculating

$$r_{\text{new}} = r_{\text{cur}} - \frac{\text{CRI}[ppm]}{1.5[ppm]} \tag{4.3}$$

```
typedef struct salp_rngInfo{
  uint8_t    bool_vld;
  uint16_t   initiator;
  uint16_t   responder;
  uint64_t   ts_txInit;
  uint64_t   ts_rxResp;
  uint64_t   respDelay;
  double     cri;
  double     rtt;
  float      dist_raw;
  float      dist_cri;
  float      dist_rtt;
} salp_rngInfo;
```

Figure 4.8: Range Info: C-struct containing all relevant data of a single ranging attempt (File: LOC/SALP_DEF.H).

The value 1.5 deduces from the fact that a single increase of the trim value changes the system clock by roughly 1.5 ppm as has been evaluated during the measurements in Chapter 3.

4.3.5 Ranging

The ranging process can be invoked via the *salp_range()* function in file LOC/SALP.H. A schematic overview of the process is shown in Figure 4.9. The function takes five parameters.

```
status salp_range(const uint16_t* devList, const size_t devListSize,
    const uint16_t sliceSize_uus, salp_rngInfo* rngInfoList, size_t* nor)
```

First, it needs the IDs of the desired anchors and the number of anchors. Second, it takes the value for the desired slize size in µs. Next, a pointer to a range info list needs to be passed. The ranging results will be stored here. The content of a range info element can be seen in Figure 4.8. The according memory space needs to be provided by the caller. The last parameter will be used to return the number of responses that have been received. The function sends the INIT message, waits for the responses and calculates the according rangings, which can then be passed to the localization engine.

4.3.5.1 Localization Engine

There are two localization methods currently implemented. First, the LLS and second the MinMax algorithm. They can be found in the LOC/SALP_LOC.H file and can be accessed via the *salp_trilateration()* and the *salp_minmax()* function.

```
status salp_trilateration(salp_rngInfo* rngInfo, const size_t
    rngInfoSize, double* pos);
status salp_minmax(salp_rngInfo* rngInfo, const size_t rngInfoSize,
    double* pos);
```

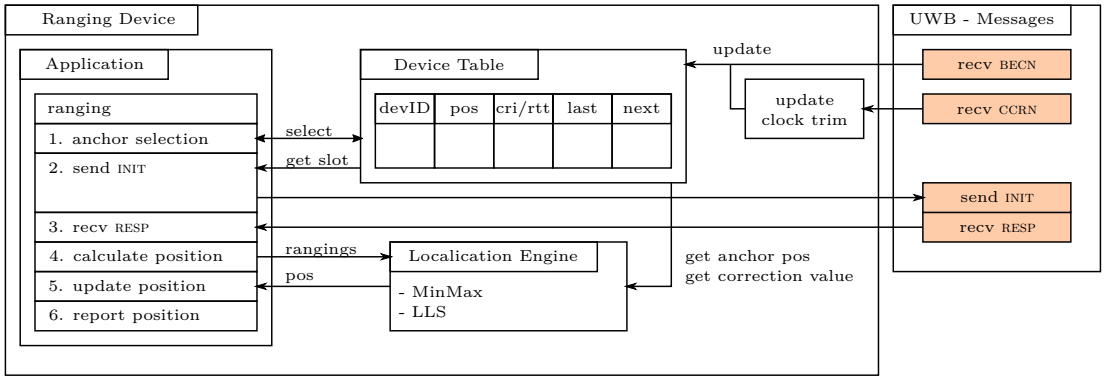


Figure 4.9: Overview of the tags ranging process.

Both functions share the same structure. They take a list of ranging information (see Figure 4.8), the size of the list, and a pointer to a field where the position will be stored. The ranging information contains all relevant data that is needed for a position calculation. It contains the *id* of the tag and the anchor, the needed time measurements as well as the CRI and RTT values. Further, it provides the distance estimations. The position of the anchors is not included. During execution the localization functions request the latest anchor positions from the device table. Taking a look at the runtime, the LLS takes 4 ms to calculate a position from six anchors. The MinMax algorithm takes below 1 ms for the same set of data.

LLS. As seen in Equation 2.13 in Section 2.2.2 to solve the LLS problem one needs to solve

$$\widetilde{x} = (A^T A)^{-1} A^T \Delta s$$

This corresponds to solving

$$(A^T A)\widetilde{x} = A^T \Delta s \tag{4.4}$$

The result of $(A^T A)$ always is a symmetric, quadratic, positive definite, 4×4 matrix. Therefore, the so called Cholesky factorization [51] can be used to solve the problem. The Cholesky solver has been implemented in the *lgs_solve_chol()* function of file MATH/LGS.H:

```
status lgs_solve_chol(matrix* L, matrix* R);
```

It takes the left and right side of the equation as parameter and solves it. The cholesky solver works as follows:
Let M be a positive definite, symmetric matrix and b be a vector. We are looking for a solution of the equation $Mx = b$. As M is positive definite and symmetric there exists a cholesky factorization of the form $M = CC^T$ where C has the shape of a triangular martix. First, C needs to be set up. Second, a forward substitution will be performed to solve $Cy = b$. Afterwards, by solving $C^T x = y$ with a backwards substitution, x can be determined. It can be seen that to solve $Mx = b$ one only needs

to perform the factorization and a simple forward- and backward substitution. In our case $M := (A^T A)$ and $b := A^T \Delta s$.
It should be noted that the solver works in-place. The implementation will override the parameters and therefore destroy them. However, as they are not needed later on this is an acceptable limitation in order to save some memory space. The result will be stored in the second parameter R.

MinMax. The MinMax algorithm has already been described in Section 2.2.1. Its implementation is really simple and does not need much explanation. After retrieving the anchor positions the perimeters of the bounding box for each anchor is calculated. For each anchor the perimeters of the bounding box will be compared to the perimeters of the intersection box. If the bounding box can minimize the intersection box, the intersection box will be updated. At the end the position is set at the middle of the intersection box.

4.3.6 Data Visualization

The visualization of the data has been done on a separate host PC. A python script has been used to retrieve the data from the node via a serial UART connection. The script writes the data to a text file. A simultaneously running second script periodically scans the text file and renders a live 3D-plot, visualizing the current tag position. The rendering has been done with the 'matplotlib' library. Further data analysis has been done, using Maltlab scripts with the open source program octave.

4.4 Related Work

There is a wide variety of different UWB-based RTLS currently available. The following section will give an overview about some solutions. A comparison to the prototype system will be done in Chapter 5.

4.4.1 Ubisense

Ubisense is a commercial localization product [57]. It is said to be one of the first commercially available UWB localization products [48]. The Ubisense system consists of a network of anchors with fixed position connected via Ethernet. Each of these anchors contains multiple antennas [48]. The to be located tag transmits an UWB signal. By its reception each anchor can calculate a range based on the timing difference among the antenna signals (TDoA) as well as an angle of arrival (AoA). Both information will be send to a central server that will calculate the tag position based on the information of multiple anchors. The combination of TDoA and AoA within a single anchor allows 3D localization with only two anchors [54].

4.4.2 Sewio

Sewio [52] is a commercial product consisting of several fixed anchors, multiple mobile tags and a server for the position calculation. Their system offers two different local-

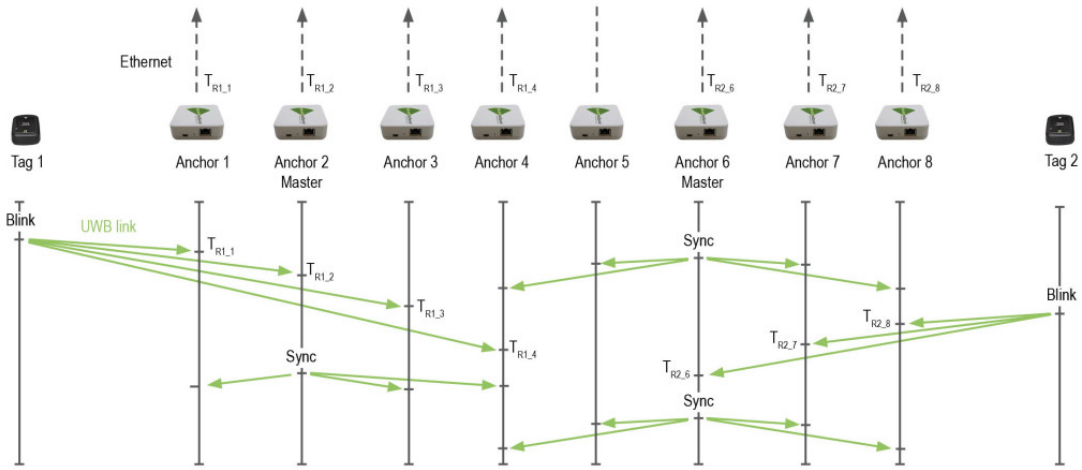


Figure 4.10: Setup of the Sewio TDoA RTLS solution. Figure taken from [52].

ization modes using TDoA or TWR schemes.
For the TDoA approach (see Figure 4.10) the anchors are interconnected via Ethernet links. The anchors synchronize via UWB. A to be localized tag sends a UWB message. The anchors report the reception of this message to the server who is then able to calculate the tag position. This approach seems to be quite common and can be seen representative for other, similar systems such as OpenRTLS [59, 43] and others.
Their second approach (see Figure 4.11) is based on individual two-way rangings among the tag and the anchors. The anchors do not need an Ethernet link. The ranging data will be transmitted to a master anchor who forwards these information to the server via an Ethernet link.

4.4.3 Decawave

Decawave build and sells its own localization solution [15]. In contrast to other commercial systems Decawave published a lot of documentation containing insights about their system.
The system consists of anchors with fixed position and mobile tags. A change of these roles is not intended. The media access follows a decentralized TDMA scheme. The system shares a common super frame (see Figure 4.12). A superframe consists of 16 beacon message slots, 15 ranging slots. Further, there are two slots (SVC) reserved for service messages and the so called almanac messages. Almanac messages contain information about the firmware- and hardware version of the system. The complete superframe takes 100 ms. The first anchor in an area (called initiator) starts sending beacons and almanac messages. The initiator thereby decides about the timing of the superframe. Other anchors will later join this network.

The anchors apply for one of the 16 beacon slots. The beacon messages of the established anchors contain a bitmap indicating which slots are believed to be free. By listening to the ongoing traffic a new anchor will thereby see if there is a free beacon

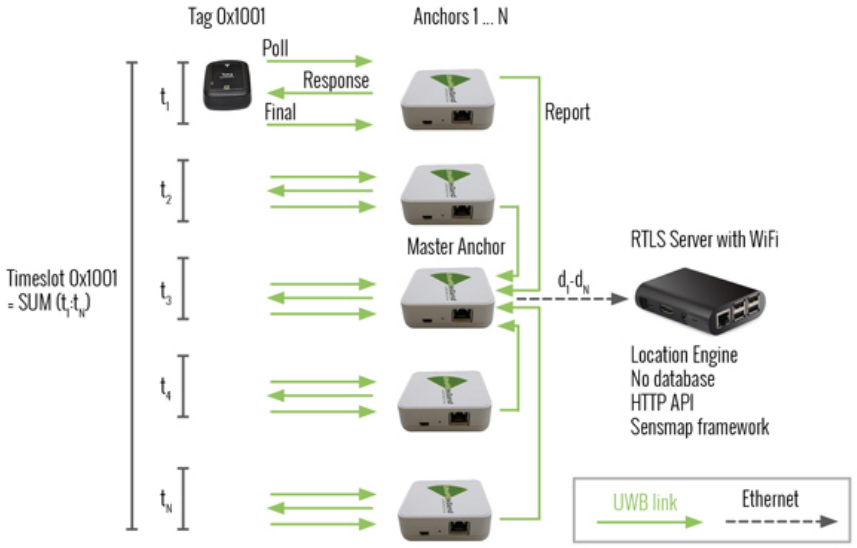


Figure 4.11: Setup of the Sewio TWR RTLS solution. Figure taken from [52].

slot. During a service message slot the new anchor can then send a join request, containing the slot number it wants to take. In the next superframe the other anchors may acknowledge the request within their beacon slot. If all other anchors agree, the new anchor may take the desired slot.

Within the beacon message the anchors send a bitmap, indicating which TWR slots are taken from their point of view. Collecting the bitmaps from all anchors, the tag is able to deduce a free TWR slot. Within the slot, tag and the anchors perform a DS-TWR (see Figure 4.12). The tag starts with a group poll message, including the desired four anchors. Ranging to more than four anchors is not intended. Further, the message includes a desired refresh rate, indicating when the next ranging is required. The selected anchors answer with a bitmap indicating free TWR slots in the desired superframe. The tag selects a free TWR slot and announces its decision in the tag response message. The anchor will acknowledge and reserve the slot in their final message. Furthermore, this final message contains all three timestamps (rx group poll; tx poll; rx response) to conclude the DS-TWR. The tag may now calculate the position.

This is just a brief overview of the Decawave system. There is more functionality in the background. For example Decawave supports over-the-air firmware updates. New anchors can update their firmware with data sent during the service message slots. Further, they implemented a routing feature to expand the network over a larger area. A gateway to connect the RTLS with the internet can also be included in the system.

Decawave presents a highly sophisticated RTLS system with a lot of features. However, it comes with some downsides. First, the system does not take advantage of an increasing number of anchors within a facility. The ranging process is always concluded

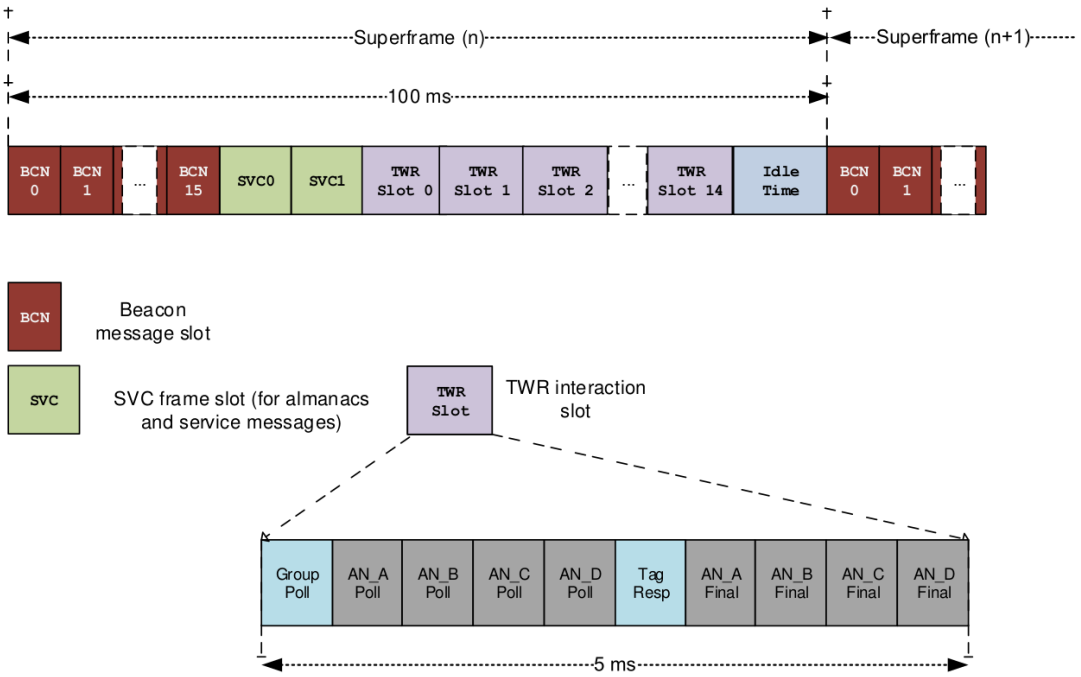


Figure 4.12: Decawave superframe structure. Figure taken from [15].

with four anchors. Other anchors will idle meanwhile.
Second, the complex TDMA scheme will in best case allow a high utilization of the channel. However, it cannot prevent collisions in all cases. Collisions may still occur, for example if two tags try to get a previously free TWR slot at the same time. In the scenario of a highly dynamic setup where tags roam through the facility, entering and leaving the range of anchors, these collisions will occur regularly.
Third, the network must be a strong connected graph. All anchors must share the same superframe. Deploying the system within a large facility thereby is hindered. Bridges must be introduced to connect the different parts of the network.

4.4.4 SurePoint

SurePoint is a RTLS solution with academical background [33]. It is an enhancement to the PolyPoint [34] system that should enable real-time tracking for a quadrotor drone in an indoor environment. SurePoint's enhancement is the additional multi-tag support. Therefore, they exploit a technique called glossy flooding [20]. Glossy flooding is a technique to spread a message across a network. The sender transmits the message. Each device receiving the message retransmits the message. Because of the UWB modulation properties the retransmissions do not destroy each other but create a constructive interference [20]. From the view of a recipient the multiple retransmissions look like echos. The reception of the original message thereby is still possible. Over multiple retransmissions the whole network can be flooded.
SurePoint uses this mechanism to manage the network. It uses a 1000 ms superframe

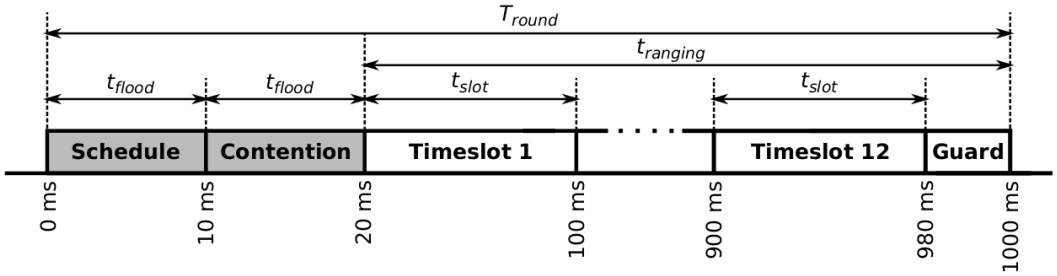


Figure 4.13: SurePoint superframe structure. Figure taken from [33].

(see Figure 4.13). The superframe consists of two flooding slots (schedule and contention slot) which is used for network management, and 12 ranging slots. A common coordinator organizes the utilization of the ranging slots. During the first flooding slot the coordinator publishes a schedule. New tags use the second flooding slot to apply for a ranging slot in the next superframe.
SurePoint uses a complex SS-TWR scheme to measure the distance between tag and anchor (see Figure 4.14). Their nodes consists of a DW1000 which can switch between three different antennas. A ranging between a tag and a single anchor consists of multiple individual rangings performed on different channels and over different antennas. Therefore, they initiate their ranging with 27 poll messages over three different channels with three different tag antennas and three different anchor antennas. Afterwards, three messages will be send by the tag by whom the anchors can calculate their clock-speed offset. Finally, the anchors send their response containing the needed ranging timestamps. The anchors send these with a random delay. As their might be collisions between the different anchors, the responses will be repeated until the tag acknowledges the reception with an additional ack message.
The reason for the high ranging effort is to account for varying antenna delays and the clock speed offset. By performing multiple rangings between two points one can identify outliers. SurePoint does not implement an exploration to detect available anchors. In case of a ranging all available anchors in range will respond. The position of the anchors seems to be a priori knowledge of the tag.

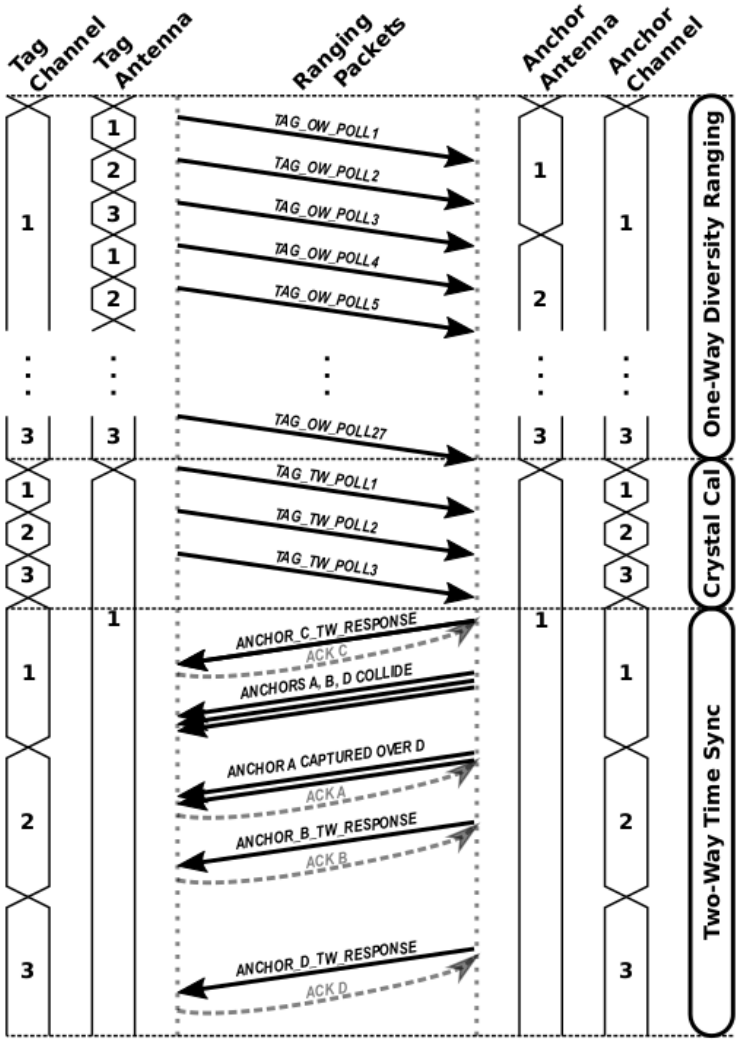


Figure 4.14: SurePoint ranging protocol. Figure taken from [33].

5 Evaluation

This chapter evaluates the performance of the proposed localization system. It will be checked if the requirements defined in Section 4.1 have been met. Therefore, this chapter will take a look at the localization performance (Section 5.1), the setup complexity (Section 5.2), the impact of the online clock calibration (Section 5.3), the number of ranging messages (Section 5.4), the achievable refresh rate (Section 5.5), and the multi-tag support (Section 5.6).

5.1 Localization Performance and Anchor Scaling Behaviour

This section takes a look at the localization performance of the presented solution in terms of its accuracy and precision.

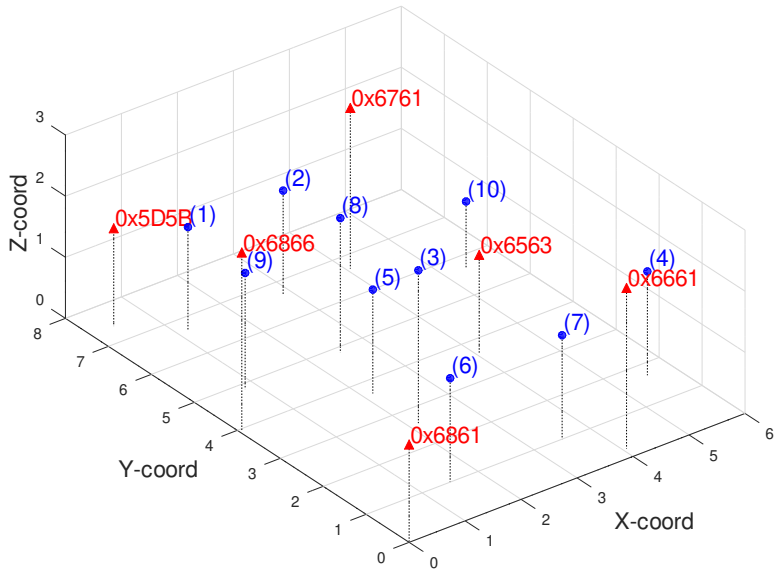


Figure 5.1: Overview of the experimental setup. Red triangles mark anchor positions. Blue dots represent the surveyed tag positions.

NodeID	X [m]	Y [m]	Z [m]
0x6861	0.00	0.00	1.60
0x6563	4.06	3.66	1.60
0x5D5B	0.41	7.41	1.60
0x6661	4.06	0.23	2.63
0x6761	4.06	6.66	2.63
0x6866	0.05	3.96	2.91

Table 5.1: Position of the anchor nodes.

position	X [m]	Y [m]	Z [m]
(1)	1.16	6.66	1.7
(2)	2.86	6.66	1.7
(3)	2.16	2.6	2.5
(4)	5.56	1.7	1.7
(5)	2.16	3.66	1.7
(6)	1.5	1	1.7
(7)	3.5	1	1.7
(8)	2.5	4.86	2.2
(9)	0.8	4.86	1.9
(10)	5.36	5.66	1.1

Table 5.2: Surveyed evaluation points.

5.1.1 Experimental Setup

The evaluation has been made in a room of roughly 4×8 metres with a height of three metres. Seven nodes have been available for the evaluation. In order to get an impression of the performance of the system, six nodes have been configured as anchors. Three anchors have been placed below the ceiling at a height of 2.60 The other three anchors have been placed on tripods at a height of 1.6 m. The exact positioning can be seen in Table 5.1 and Figure 5.1.

5.1.2 Localization Performance

Ten positions within the room have been evaluated, with two evaluation points beeing outside the convex hull of the anchors (see Table 5.2). The others have been spread randomly across the room, in different heights and locations. Per evaluation point 500 measurements have been taken. The positions have been calculated with the MinMax and the LLS algorithm. In order to show the improvement due to the correction methods three different scenarios have been evaluated:

Scenario 1: offline clock calibration with default-values according to [5].
Scenario 2: online clock calibration with CRI corrected rangings and received signal strength correction
Scenario 3: online clock calibration with EWMA-CRI ($\lambda = 0.3$) corrected rangings and received signal strength correction

The cumulative distribution function (CDF) over the position error in metre can be seen in Figure 5.2. The position error is defined as the Euclidean distance between the evaluation point and its estimates.

In scenario 1 with default calibration and without correction, meaningful localization has not been possible for both localization algorithms. The ranging offsets have been too large to give meaningful information. This corresponds to the findings of Chapter 3. Due to imprecise clock-trim values (see Figure 3.8, Section 3.4.3.3) an offset of several metres can be induced. This effect will be further amplified by the increasing response delays (see Figure 3.6, Section 3.4.3.1). High offsets in the ranging highly influence the localization algorithm. For MinMax the bounding boxes either do not intersect (negative offset) or do not further limit the intersection box (positive offset). Thereby the algorithm mostly predicts the position at the center of the first bounding box which happens to be the position of the first anchor. Meaningful localization thereby was not possible. For the LLS, offsets have a quadratic impact. Most estimated tag positions have been outside the evaluation room. As for MinMax a meaningful localization has not been possible. Therefore, in contrast to the other two scenarios, the results have not been shown in a scatter plot.

In scenario 2 the clock speed offset has been reduced via the online clock calibration method. Further, the offset due to the received signal strength has been corrected. The results can be seen in Figure 5.3 as a scatter plot.

Scenario 3 shows the impact of the EWMA applied to the CRI value. As in the second scenario the online clock calibration and the received signal strength correction has been performed. The calculated positions are shown in Figure 5.4.

As can be seen in Figure 5.3 and 5.4, MinMax cannot determine positions outside of the convex hull. MinMax estimates positions outside the convex hull to be on the edge of the convex hull. In order to compare the performance of LLS and MinMax the evaluation points outside the hull have been ignored for the further performance analysis. Table 5.3 shows the performance of the localization in all three scenarios. Therefore, the mean absolute error (MAE), the root-mean-square error (RSME) and the 90th percentile have been calculated. The MAE between the evaluation point x_0 and its n position estimations $\{x_i | i \in [1:n]\}$ has been calculated via:

$$MAE = \frac{\sum_{i=1}^{n} |x_i - x_0|}{n} \tag{5.1}$$

where $|x_i - x_0|$ represents the euclidean distance between the evaluation point and a single estimation. The RMSE is derived via:

$$RSME = \sqrt{\frac{\sum_{i=1}^{n} (|x_i - x_0|)^2}{n}} \tag{5.2}$$

Both, MAE and RSME are measurements metrics for the accuracy of estimations. In contrast to the MAE the RSME penalizes bad estimations harder, as their error

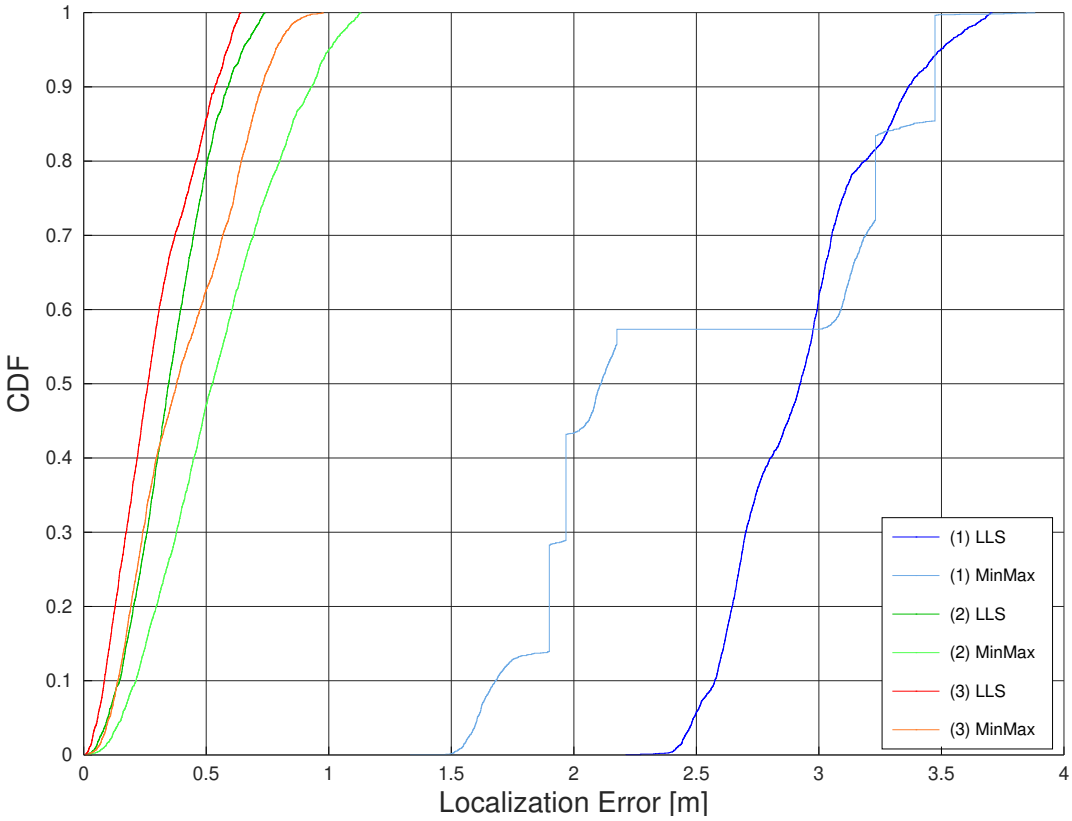


Figure 5.2: CDF of the 3D-localization error. CDF only includes measurements inside the convex hull. Scenario (1) offline clock calibration with default values (blue). Scenario (2) online clock calibration + RSS correction + CRI correction (green). Scenario (3) online clock calibration + RSS correction + EWMA-CRI correction (red).

contributes quadratic. For example if all estimations are spread evenly around the evaluation point, the MAE will be close to zero. However, he RSME will take a look at the residuals and therefore will be higher, depending on the deviation of the estimations. Reaching a good RSME value therefore is considerably more difficult than having a good MAE. The 90th percentile is another commonly used metric. It can be directly derived from the CDF (see Figure 5.2) and gives an upper bound for the error that will be reached in 90 % of the cases.

As already described a localization has not been possible in the scenario 1. A MAE of about three metres (see Table 5.3a) does not allow reliable indoor localization. In scenario 2 a vast improvement can be seen. Both localization algorithms lead to useful position data. In this scenario MinMax and LLS seems to produce similar results. Only the precision of the height estimation (z) seems to be better with MinMax (0.21 m deviation versus 0.72 m). LLS however, benefits when the tag is positioned outside the convex hull. In contrast to MinMax the estimated positions are not bound by the edge

	std-deviation σ [m]		RMSE [m]	MAE [m]	90% [m]
LLS	x	0.296	3.595	2.977	3.366
	y	0.153			
	z	0.316			
MinMax	x	0.171	2.747	2.708	3.474
	y	0.163			
	z	0.101			

(a) scenario 1

	std-deviation σ [m]		RMSE [m]	MAE [m]	90% [m]
LLS	x	0.277	0.954	0.413	0.589
	y	0.208			
	z	0.723			
MinMax	x	0.260	0.780	0.571	0.930
	y	0.216			
	z	0.212			

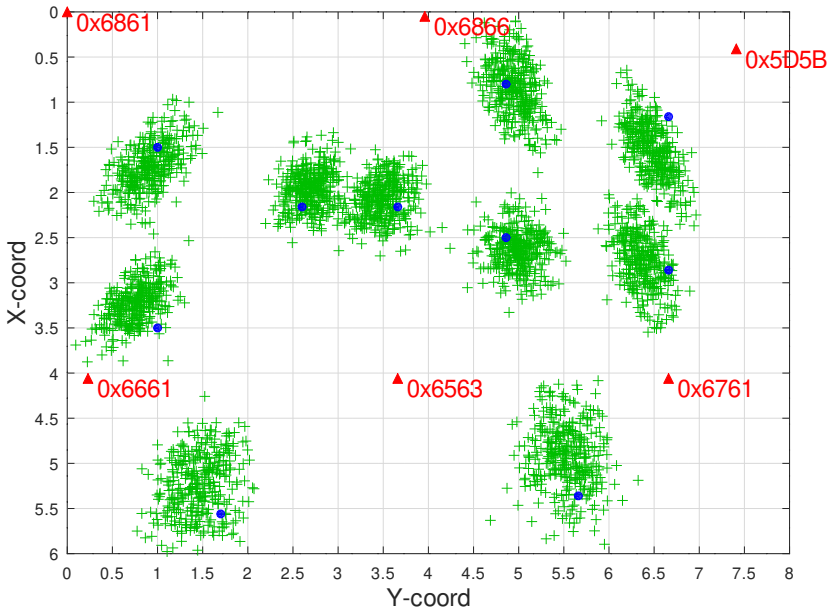
(b) scenario 2

	std-deviation σ [m]		RMSE [m]	MAE [m]	90% [m]
LLS	x	0.113	0.607	0.306	0.535
	y	0.087			
	z	0.358			
MinMax	x	0.171	0.694	0.431	0.726
	y	0.163			
	z	0.101			

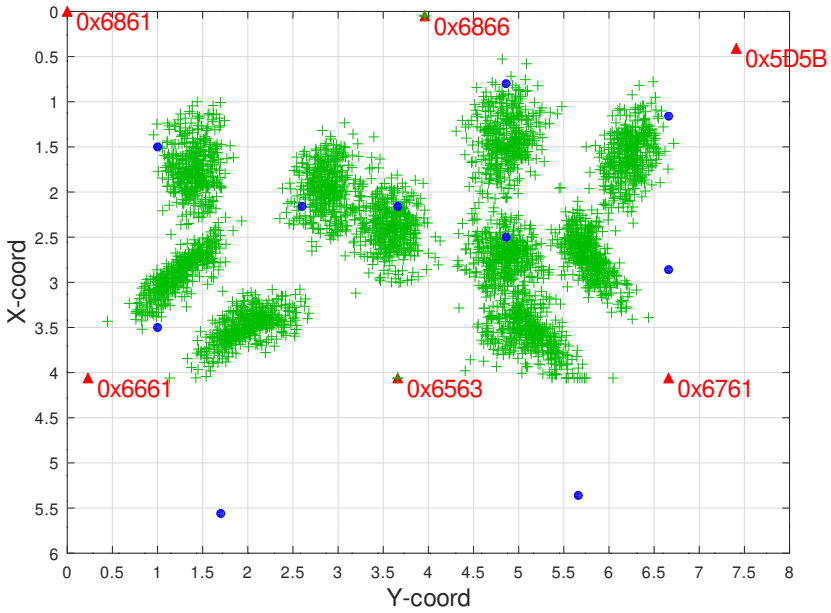
(c) scenario 3

Table 5.3: Performance of the localization algorithms, within the convex hull, in different scenarios. Best results have been achieved in the third scenario where all correction and calibration methods have been applied.

of the convex hull. Scenario 3 is based on the best ranging results that can be achieved with the implemented prototype. It can be seen that compared to the second scenario the precision improved significantly. The precision in terms of the deviation has halved for LLS localization and also the precision for MinMax has been improved. It should be noted that the EWMA only applies to the CRI correction value for the ranges and not to the position itself. Thereby the movement of the tag could be tracked without delay.

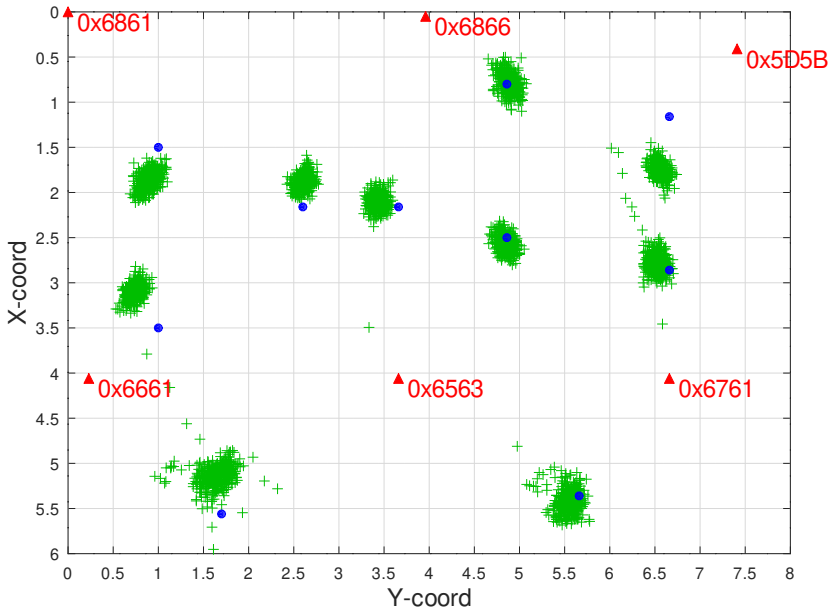


(a) LLS localization

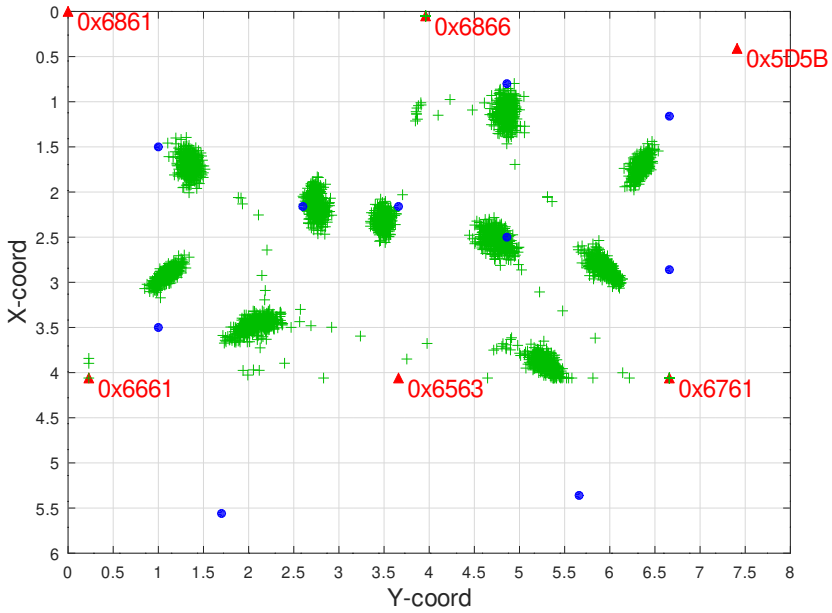


(b) MinMax localization

Figure 5.3: **Scenario 2:** Evaluation of the localization performance (view from the top). Localization with online clock-calibration and CRI-corrected rangings. Red triangles denote the anchor positions. Blue dots the tag placement and green crosses the position estimates.



(a) LLS localization



(b) MinMax localization

Figure 5.4: **Scenario 3:** Evaluation of the localization performance (view from the top). Localization with online clock-calibration and EWMA-CRI-corrected rangings. Red triangles denote the anchor positions. Blue dots the tag placement and green crosses the position estimates.

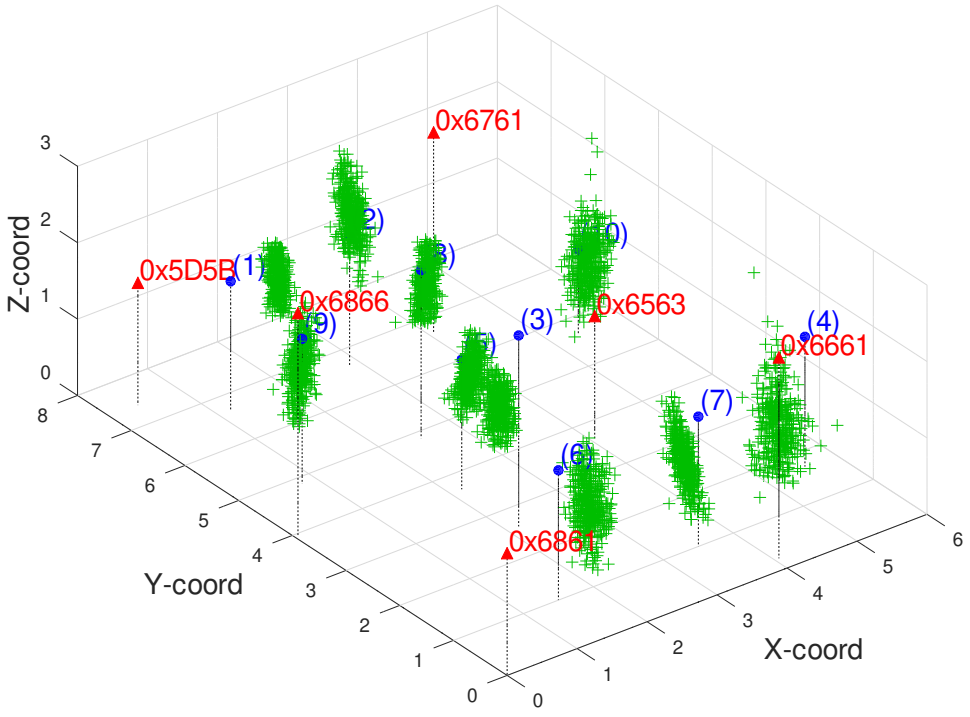


Figure 5.5: 3D view of the LLS localization with EWMA-CRI corrected rangings (Scenario 3).

5.1.3 Simulative evaluation

As the number of available nodes in the evaluation in Section 5.1 does not allow statements about the performance in terms of increasing number of anchors, this scenario has been simulated. Further, the impact of different anchor constellations and different tag placements can be tested more easily in a simulation. The simulation has been performed with a Matlab script using GNU Octave. The simulation environment is identical to the experimental setup presented in 5.1.1. The position of the six fixed anchors is the same. Further, an arbitrary number of additional anchors can be created. These devices have been placed randomly within the evaluation room. Now random tag positions can be simulated retrieving SS-TWR distance estimations based on the simulation model. The positions have been calculated using MinMax (see Section 2.2.1) and the LLS algorithm (see Section 2.2.2).

Simulation model. The simulation model has been derived from the findings of Chapter 3. The antenna has been assumed to be calibrated properly. Further, the signal strength has been assumed to be dealt with. The simulation model applies normal distributed noise upon the true distance. The model incorporates the increasing deviation due to increasing response delays and clock speed offset (see Section 3.4.3) by applying

a linear increase of the standard deviation. The ranging to the first anchor will have a deviation of 4 cm corresponding to the results of the EWMA-CRI corrected ranging measurements (see Section 3.4.4). For each following anchor, the standard deviation will be increased by 4 cm as it has been observed previously (see Figure 3.6).

Simulation step. Each simulation step consists of a random tag position within the evaluation room and the number of anchors that should be used in this step. First, the used anchors have to be chosen. Two different options have been investigated.

1. fully random - all anchors have been chosen fully random out of the given set of 25 available anchors.
2. partially random - the first six anchors have been fixed. They correspond to the anchor location in Section 5.1. The idea behind this pre-selection is to avoid a bad constellation of the randomly chosen anchors.

Second, the position is calculated by the chosen algorithms. Now the absolute error between the estimated tag position and the real tag position can be calculated. The results from several simulation steps with different tag positions and different chosen anchors result in a mean-absolute error (MAE) for a given number of anchors.

Simulation results In total, 25 anchors have been simulated. For each number of anchors 1000 simulations have been performed from which the MAE has been calculated. The results are shown in Figure 5.6. The simulation shows that the LLS algorithm as well as the MinMax algorithm both benefit from the increasing number of anchors. Further, it can be seen that a proper anchor selection is key to a good localization result. Especially the LLS with fully random anchor selection suffers from a small number of anchors. However, if the tag position is within the convex hull the results improve vastly (see cyan line in Figure 5.6). The MinMax algorithm seems to be more robust against bad anchor selection. The difference between fully random anchors and partially random is not as huge as for LLS. Therefore, it might be a viable option to use an initial position estimation using the MinMax algorithm. Having an initial position guess one may select promising anchors and perform a second localization attempt using LLS for higher precision.

Obviously it can be seen that localization performance improves with a higher number of anchors. The benefit of multiple additional rangings seems to overcome the drawback of their reduced precision. However, as a higher number of anchors imply a higher channel occupation one needs to make a reasonable trade-off between the number of anchors and the precision.

Compared to the results of the real-world evaluation (see Section 5.1.2) the simulation seems to be a conservative estimation. The real-world results seem to perform better than the simulation, especially when using MinMax. However, it should be noted that the simulation results might be worse because more unfavourable tag positions have been evaluated than it was done in the real-world evaluation.
Adler et al. [2, 3] claim a much higher MAE of the LLS approach. Their results for 2D

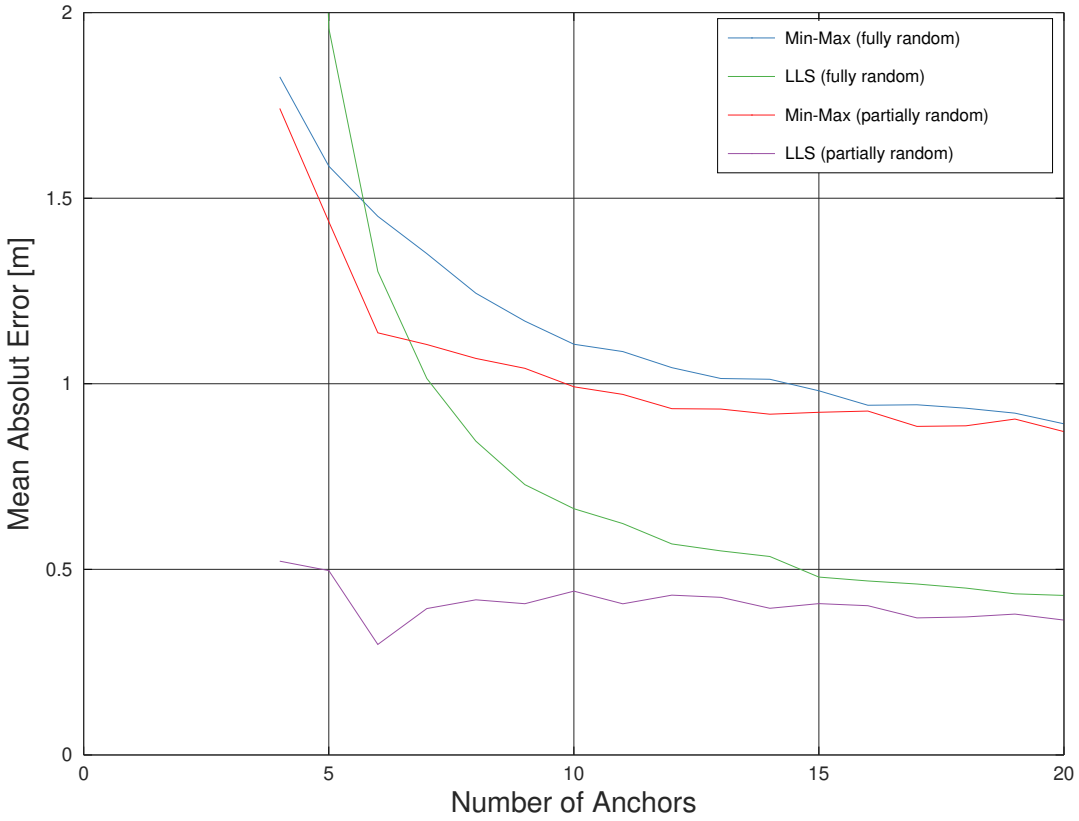


Figure 5.6: 3D-localization error for increasing number of anchors (fully and partial random chosen).

localization indicate a MAE of 12.65 m for LLS and 2.05 m for MinMax. However, their results are not directly applicable in our case as they surveyed localization with a worse performing ranging technique. They claim a standard deviation of 3.04 m whereas our deviation reaches from 0.04 m for the first anchor to up to 0.8 m for the 20th anchor. Further, comparing these results, one needs to keep in mind that their measurements seem to be biased. This bias influences the performance of LLS quite heavily as they pointed out [3].

5.1.4 Performance Comparison

Comparing the localization performance of different RTLS is not an easy task. First, there are different ways to measure the accuracy of a RTLS. Some developers state the root-mean-square error, some the mean-absolute error and others the 90th percentile. In some cases there is a value without any statement of the measurement method. Second, the localization accuracy is often used to advertise a certain system. For example Decawave states an accuracy below 10 cm [15], however, Ruiz et al. [48] measured an accuracy of 0.49 cm (MAE) in a real world evaluation. Especially the statements from commercial products therefore should be handled with care. Third, the evaluation settings have a high impact on the localization performance. For example, as seen

	Localization Algorithm	MAE [m]	RMSE [m]	90% [m]	Precision [m]
Ubisense	server based	0.15 - 1.10	1.82	2.39	-
Sewio (TWR)	server based	-	-	-	-
Sewio (TDoA)	server based	-	-	-	-
Decawave	tag based	0.10 - 0.49	0.59	1.09	-
SurePoint	server based NLLS	0.29	-	0.50	0.12
This work	tag based MinMax LLS	 0.43 0.31	 0.69 0.61	 0.73 0.54	 0.10 - 0.17 0.09 - 0.36

	Ranging Technique	Ranging Messages	Refresh Rate	References
Ubisense	TDoA, AoA	1	10 Hz	[48, 54]
Sewio (TWR)	DS-TWR	3n	-	[52]
Sewio (TDoA)	TDoA	1	0.1 Hz to 10 Hz	[47, 52]
Decawave	DS-TWR	10 (4 anchors)	10 Hz (1 tag) 10 Hz (15 tags 1 Hz (150 tags)	[48, 15]
SurePoint	SS-TWR	> 27+3+2n	12 Hz (1 tag) 1 Hz (12 tags)	[33]
This work	SS-TWR	n+1	66 Hz (1 tag) 1 Hz(12 tags) 0.1 Hz (122 tags)	

Table 5.4: Comparison of different UWB-based RTLS.

previously the anchor placement or the number of used anchors has an impact on the results. A proper comparison must tackle all these different evaluations to give a fair comparison.
Because of these differences a fair performance comparison among different RTLS is difficult. A sufficient comparison would need a standardized testing environment for all test subjects, covering as many different scenarios as possible. Otherwise a comparison has to rely on published data from different sources. Still, table 5.4 tries to give such an overview containing the RTLS introduced in Section 4.4 and compares it with the proposed localization system.

5.2 Setup Complexity

The setup complexity of the implemented prototype is low as each device can operate completely independent. It is sufficient to distribute the devices within the building. There is no need for additional infrastructure (e.g., an Ethernet backbone) as needed in other systems (see Section 4.4). All necessary calculations are performed on the nodes. Therefore, there is no need for a more powerful host PC or even a central localization

server. Therefore, there are no restrictions concerning the connectivity of the network graph. The structure also allows the deployment in a loose network. A tag can thereby localize itself with anchors that might not even know about each other. In other TWR systems like Decawave, SurePoint or Sewio all anchors need to be managed in a common network. Therefore, the loss of a single device might tear the whole network apart and prevent further localization. In a loose connected system of independent devices such as the presented prototype the loss of single devices is not relevant.

5.3 Online Clock Calibration

The online clock calibration vastly reduces the clock speed offset. Using the default values for the clock-trim induces a clock speed offset of up to 12 ppm among the devices that have been used for the prototype. After the online clock calibration the maximum offset can be reduced to below 1.5 ppm. Thereby, the SS-TWR results can be improved significantly (see Section 3.5). The online clock calibration can also be used by non-coherent devices to perform accurate rangings. As the RTT implementation is mandatory in the IEEE 802.15.4 standard, non-coherent devices may perform a RTT-based online clock calibration. Afterwards, ranging with reasonable accuracy can be performed.

5.4 Number of Ranging Messages

The number of ranging messages of the prototype depends on the number of desired anchors. As each anchors sends its response after the broadcast of a single initialization message, it takes $n + 1$ messages with n being the number of anchors. This is the minimum number of ranging messages that can be achieved following the TWR approach. Only TDoA or ToA systems need less messages as they can localize with a single transmission.
SurePoint [33] takes 27 messages to initiate rangings over three different channels with three different antennas. Further, they need three messages for a crystal calibration and at least one response for each of the three anchors. Each response gets acknowledged by the tag resulting in further three messages. This results in a total of at least $30+2n$ messages. This number can further increase as their responses are not scheduled and therefore need retransmissions in case of a collision. This may result in even more messages.
Decawave performs DS-TWR to four anchors. As the tags messages are broadcasted and the anchor responses are scheduled it only takes 10 messages to derive the position information. Allowing ranging to more than four anchors in a single ranging attempt is not intended.

5.5 Localization Runtime and Refresh Rate

The runtime of the localization algorithm is a major factor. While in theory the maximal refresh rate of the system is only limited by the duration of the ranging in praxis

Update Rate λ [Hz]	**Number of Devices**
0.01	1226
0.05	245
0.1	122
0.5	24
0.7	17
1	12
2	6
3	4
4	3
6	2

Table 5.5: Update Rate versus the number of devices.

also the runtime of the localization algorithm is crucial. Quick, consecutive rangings are useless if the data cannot be processed in time.

Refresh Rate. Not taking into account the runtime of the localization algorithm, the refresh rate only depends on the duration of the ranging. The duration can be estimated. The previous experiments for Chapter 3 showed that 3 ms per anchor are a realistic estimation (preamble length 128, PRF 64 MHz, data rate 6.8 Mbit/s). 3 ms do not only include the duration of the transmission but also the time needed for processing the message. Additional 3 ms can be assumed for the transmission of the INIT message. The theoretical refresh rate for a given number of desired anchors n can be calculated by:

$$\text{Refresh Rate}[Hz] = \frac{1}{(3[ms] + 3[ms] * n) * 10^{-3})} \tag{5.3}$$

Giving a theoretical maximal refresh rate of 66 Hz for the continuous ranging with four anchors.

Runtime. The runtime of the localization algorithm is key to the refresh rate of the tag. In our measurements the LLS algorithm took 4 ms for a setup with six anchors. The MinMax algorithm has taken well below 1 ms for the same set of data. A complete ranging process including the ranging to four different anchors and the position calculation of LLS and MinMax therefore will take below 20 ms. Therefore, the maximum refresh rate including the position estimation is approximately 50 Hz. Thereby it can be seen that performing the localization calculation on the node does not prevent a high update rate.

5.6 Maximum Number of Tags

Following Abramson [1] a random access channel will become unstable if the channel occupation is above $1/2e = 0.186$. Abramson defines an upper limit for the number of

potential interactive users k_{max} as

$$k_{max} = (2e\lambda\tau)^{-1} \tag{5.4}$$

with e being the Euler's number, λ being the desired update rate of a device and τ being the duration of a single packet. As before the duration of a single ranging with four anchors can be estimated to be 15 ms. Therefore, τ will be set to $15 * 10^{-3}$s. Lets further assume each device wants to perform a ranging each minute ($\lambda = 1/60$). Therefore, the maximum number of devices will be:

$$k_{max} = (2e * (1/60) * (15 * 10^{-3}))^{-1} = 735.8 \tag{5.5}$$

Table 5.5 shows the relation between achievable update rate and the number of devices. It can be seen that a tradeoff between scalability and update rate needs to be taken. How this tradeoff is achieved will depend on the application of the RTLS. This example assumed that only four anchors are available. If more anchors are available the prototype system will benefit from the parallel ranging via the side channels. Thereby the systems update rate will increase.
Consumer grade GPS Hardware such as the Garmin GLO [21] or the BT-Q818XT by Qstarz [45] advertise a 10 Hz update rate. However, for common, cheap GPS modules that are used in smartphones or navigation systems such as the ublox NEO-6 [58], 1 Hz seems to be sufficient.

6 Conclusion

This thesis investigated the potential of SS-TWR for real-time location systems. It has been shown how to improve the distance estimations of SS-TWR through calibration and correction techniques. The results of Chapter 3 proof that an accuracy around 4 cm is achievable.

Further, the thesis took a look at the performance of simple three-dimensional localization algorithms that can also be executed on a MCU with limited computational performance. In combination with the high ranging performance of UWB and the possibility of benefiting from an increasing number of anchors, these algorithms perform reasonably well. An accuracy below 0.5 m with three degrees of freedom is easily achievable.

The proposed prototype enables these possibilities. Its simplicity ensures high flexibility for different localization approaches. It is easy deployable and supports the localization of multiple tags. The system has been implemented on the Nucleo platform proving the position estimation can be performed in the context of an embedded system without the need of external localization engines.

6.1 Outlook

This thesis only denotes a first approach towards RTLS. There is plenty of space for further improvements. The aim of improvements should to increase the localization performance in terms of accuracy and precision and to survey and improve the system's scalability.

Improve ranging. In order to improve the ranging performance the influence of further parameters needs to be surveyed. Using different channel parameters may improve the performance. For example channel 4, 7, 11 and 15 (see Table 2.1) offer a higher bandwidth and thereby might be interesting candidates. Further, shorter preambles will reduce the duration of a message. Thereby the response delays would become shorter, allowing a reduction of the ranging error. On the other hand changes such as the chosen channel or the preamble length may influence the distribution of our correction values. These effects need further research.

Improving Localization Performance. Obviously the localization performance would benefit from an increase of ranging performance. However, there is also room for improvement in the localization algorithm itself. First, introducing weights to the localization algorithm seems to be a promising approach. Weighting measurements is a common concept for the LLS approach and also the MinMax algorithm can be weighted. A weighting model should incorporate the findings of Chapter 3, foremost the influence of increasing response delay. Further, such a model could take the anchor position and many other influences into account.
Second, the possibility of sensor fusion should be investigated. For example introducing barometer measurements can improve the quality of height estimations leading to a better localization performance.

System scalability. Furthermore, the scalability of the system needs to be investigated. Deploying the system at a larger scale e.g., a whole building complex, with multiple tags will point out further challenges. Therefore, more nodes will be needed. Also the interference with UWB devices with different properties (e.g. different antennas, different clock characteristics) needs to be surveyed. Further, a large scale deployment will also show how the localization keeps up with different effects such as non line of sight conditions (NLOS), multipath effects or non-optimal anchor placement.

Bibliography

[1] N. Abramson. The aloha system: another alternative for computer communications. In *Proceedings of the November 17-19, 1970, fall joint computer conference*, pages 281–285. ACM, 1970.

[2] S. Adler, S. Schmitt, and M. Kyas. Experimental evaluation of the spatial error distribution of indoor localization algorithms. In *Ubiquitous Positioning Indoor Navigation and Location Based Service (UPINLBS), 2014*, pages 44–53. IEEE, 2014.

[3] S. Adler, S. Schmitt, Y. Yang, Y. Zhao, and M. Kyas. Experimental evaluation of indoor localization algorithms. In *2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 291–299. IEEE, 2014.

[4] M. Altini, D. Brunelli, E. Farella, and L. Benini. Bluetooth indoor localization with multiple neural networks. In *IEEE 5th International Symposium on Wireless Pervasive Computing 2010*, pages 295–300. IEEE, 2010.

[5] B. Baumann. A practical low-cost ultra-wideband research platform. Technical report, TU Graz - Institute of Technical Informatics, 2018.

[6] C.-F. Chiasserini and R. R. Rao. Coexistence mechanisms for interference mitigation in the 2.4-ghz ism band. *IEEE Transactions on Wireless Communications*, 2(5):964–975, 2003.

[7] K. Chintalapudi, A. Padmanabha Iyer, and V. N. Padmanabhan. Indoor localization without the pain. In *Proceedings of the sixteenth annual international conference on Mobile computing and networking*, pages 173–184. ACM, 2010.

[8] A. A. D'Amico, L. Taponecco, and U. Mengali. Ultra-wideband toa estimation in the presence of clock frequency offset. *IEEE Transactions on Wireless Communications*, 12(4), 2013.

[9] Decawave. Decawave's scensor DW1000: The world's most precise indoor location and communication CMOS chip. `https://www.decawave.com/sites/default/files/resources/decawave_scensor_presentation_santa_clara_1.pdf`, 2013. presentation at IdTechEx USA 2013, Accessed: 2019-02-09.

[10] Decawave. APS010 application note - wireless sensor networks and the DW1000. Technical Report Version 1.1, 2014.

[11] Decawave. APS011 application note - sources of error in DW1000 based two-way ranging (TWR) schemes. Technical Report Version 1.1, 2014.

[12] Decawave. APS014 application note - antenna delay calibration of DW1000-based products and systems. Technical Report Version 1.01, 2014.

[13] Decawave. DW1000 datasheet. Technical Report Version 2.09, 2015.

[14] Decawave. DW1000 User Manual. Technical Report Version 2.14, 2017.

[15] Decawave. DWM1001 system overview and performance. Technical report, 2017.

[16] Decawave. DW1000 application programming interface with STM32F10x application examples. `https://www.decawave.com/software/`, January 2019. Accessed: 2019-02-14.

[17] Decawave. New version api software package available. `https://decaforum.decawave.com/t/new-version-api-software-package-available/3467`, January 2019. Forum Post, Accessed: 2019-02-14.

[18] B. Ding, L. Chen, D. Chen, and H. Yuan. Application of rtls in warehouse management based on rfid and wi-fi. In *2008 4th International Conference on Wireless Communications, Networking and Mobile Computing*, pages 1–5, Oct 2008.

[19] Federal Communications Commission (FCC). Revision of part 15 of the commission's rules regarding ultra-wideband transmission systems, 2002.

[20] F. Ferrari, M. Zimmerling, L. Thiele, and O. Saukh. Efficient network flooding and time synchronization with glossy. In *Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks*, pages 73–84. IEEE, 2011.

[21] Garmin. Garmin website. `https://buy.garmin.com/de-AT/AT/p/109827`, April 2019. Accessed: 2019-04-02.

[22] T. Gigl, G. J. M. Janssen, V. Dizdarevic, K. Witrisal, and Z. Irahhauten. Analysis of a uwb indoor positioning system based on received signal strength. In *2007 4th Workshop on Positioning, Navigation and Communication*, pages 97–101, March 2007.

[23] B. Großwindhager, C. A. Boano, M. Rath, and K. Römer. Enabling runtime adaptation of physical layer settings for dependable uwb communications. In *2018 IEEE 19th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, pages 01–11, June 2018.

[24] W. Guo, W. M. Healy, and M. Zhou. Impacts of 2.4-GHz ISM band interference on IEEE 802.15. 4 wireless sensor network reliability in buildings. *IEEE Transactions on Instrumentation and Measurement*, 61(9):2533–2544, 2012.

[25] A. Hatami and K. Pahlavan. Performance comparison of rss and toa indoor geolocation based on uwb measurement of channel characteristics. In *2006 IEEE 17th International Symposium on Personal, Indoor and Mobile Radio Communications*, pages 1–6, Sept 2006.

[26] J. I. Huircán, C. Muñoz, H. Young, L. Von Dossow, J. Bustos, G. Vivallo, and M. Toneatti. Zigbee-based wireless sensor network localization for cattle monitoring in grazing fields. *Computers and Electronics in Agriculture*, 74(2):258–264, 2010.

[27] J. S. Hunter. The exponentially weighted moving average. *Journal of Quality Technology*, 18(4):203–210, 1986.

[28] IEEE. IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirement Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs). Standard, 2007.

[29] IEEE. IEEE standard for low-rate wireless networks. Standard, April 2016.

[30] Y. Jiang and V. C. Leung. An asymmetric double sided two-way ranging for crystal offset. In *Signals, Systems and Electronics, 2007. ISSSE'07. International Symposium on*, pages 525–528. IEEE, 2007.

[31] Z. Jianwu and Z. Lu. Research on distance measurement based on rssi of zigbee. In *2009 ISECS International Colloquium on Computing, Communication, Control, and Management*, volume 3, pages 210–212, Aug 2009.

[32] M. N. Kamel Boulos and G. Berry. Real-time locating systems (rtls) in healthcare: a condensed primer. *International Journal of Health Geographics*, 11(1):25, Jun 2012.

[33] B. Kempke, P. Pannuto, B. Campbell, and P. Dutta. Surepoint: Exploiting ultra wideband flooding and diversity to provide robust, scalable, high-fidelity indoor localization. In *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM*, pages 137–149. ACM, 2016.

[34] B. Kempke, P. Pannuto, and P. Dutta. Polypoint: Guiding indoor quadrotors with ultra-wideband localization. In *Proceedings of the 2nd International Workshop on Hot Topics in Wireless*, pages 16–20. ACM, 2015.

[35] K. Langendoen and N. Reijers. Distributed localization in wireless sensor networks: a quantitative comparison. *Computer networks*, 43(4):499–518, 2003.

[36] J. X. Lee, Z. Lin, and C. P. S. Francois. Symmetric double side two way ranging with unequal reply time. In *Vehicular Technology Conference, 2007. VTC-2007 Fall. 2007 IEEE 66th*, pages 1980–1983. IEEE, 2007.

[37] G. Mace. Current makefile for stm32l476 (hal). `https://gist.github.com/macegr/11887b067116cc17fefb071c955e6ee4`, January 2017. Accessed: 2019-03-12.

[38] M. Malajner, P. Planinsic, and D. Gleich. Uwb ranging accuracy. In *2015 International Conference on Systems, Signals and Image Processing (IWSSIP)*, pages 61–64, Sept 2015.

[39] V. Malyavej, W. Kumkeaw, and M. Aorpimai. Indoor robot localization by rssi/imu sensor fusion. In *2013 10th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, pages 1–6, May 2013.

[40] F. M. Martel, J. Sidorenko, C. Bodensteiner, and M. Arens. Augmented reality and uwb technology fusion: Localization of objects with head- mounted displays. *Proceedings of the 31st International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2018)*, pages 685–692, 2018.

[41] P. Meissl, K. Stubenvoll, and J. zeher. Ein computer programmsystem zur verdichtung trigonometrischer netze. In *Mitteilungen der Geodätischen Institute der TU Graz*, volume 25. Geodätischen Institute der TU Graz, Graz, 1977.

[42] D. Neirynck, E. Luk, and M. McLaughlin. An alternative double-sided two-way ranging method. In *Positioning, Navigation and Communications (WPNC), 2016 13th Workshop on*, pages 1–4. IEEE, 2016.

[43] OpenRTLS. Openrtls website. `https://openrtls.com/`, March 2019. Accessed: 2019-03-30.

[44] Y. Qi, H.-b. Li, S. Hara, and R. Kohno. Clear channel assessment (cca) with multiplexed preamble symbols for impulse ultra-wideband (uwb) communications. In *2006 IEEE International Conference on Ultra-Wideband*, pages 675–680. IEEE, 2006.

[45] Qstarz. Qstarz BT-Q818XT website. `http://www.qstarz.com/Products/GPS%20Products/BT-Q818XT-F.htm`, April 2019. Accessed: 2019-04-02.

[46] M. Rath, J. Kulmer, M. S. Bakr, B. Großwindhager, and K. Witrisal. Multipath-assisted indoor positioning enabled by directional uwb sector antennas. In *Signal Processing Advances in Wireless Communications (SPAWC), 2017 IEEE 18th International Workshop on*, pages 1–5. IEEE, 2017.

[47] M. Ridolfi, S. Van de Velde, H. Steendam, and E. De Poorter. Analysis of the scalability of uwb indoor localization solutions for high user densities. *Sensors*, 18(6):1875, 2018.

[48] A. R. J. Ruiz and F. S. Granja. Comparing ubisense, bespoon, and decawave uwb location systems: Indoor performance analysis. *IEEE Transactions on instrumentation and Measurement*, 66(8):2106–2117, 2017.

[49] U. Rüppel, K. M. Stübbe, and U. Zwinger. Indoor navigation integration platform for firefighting purposes. In *2010 International Conference on Indoor Positioning and Indoor Navigation*, pages 1–6, Sept 2010.

[50] Z. Sahinoglu and S. Gezici. Ranging in the IEEE 802.15. 4a standard. In *Wireless and Microwave Technology Conference, 2006. WAMICON'06. IEEE Annual*, pages 1–5. IEEE, 2006.

[51] W.-D. Schuh and J. M. Brockmann. *The Numerical Treatment of Covariance Stationary Processes in Least Squares Collocation*, pages 1–36. Springer Berlin Heidelberg, Berlin, Heidelberg, 2018.

[52] Sewio. Sewio website. `https://www.sewio.net/uwb-technology/two-way-ranging/`, March 2019. Accessed: 2019-03-30.

[53] J. Sidorenko, V. Schatz, N. Scherer-Negenborn, M. Arens, and U. Hugentobler. Decawave uwb clock drift correction and powerself-calibration. *arXiv preprint arXiv:1902.11085*, 2019.

[54] P. Steggles and S. Gschwind. The ubisense smart space platform.

[55] STM. STM32Cube MCU package for STM32 L1 series (HAL, low-layer APIs and CMSIS (CORE, DSP, RTOS), USB, file system, RTOS, touch sensing, graphic - coming with examples running on st boards: STM32 Nucleo, discovery kits and evaluation boards). `https://www.st.com/content/st_com/en/products/evaluation-tools/product-evaluation-tools/mcu-mpu-eval-tools/stm32-mcu-mpu-eval-tools/stm32-nucleo-boards/nucleo-l152re.html#sw-tools-scroll`, 2019. Accessed: 2019-03-02.

[56] M. Stocker. Design of a decentralized and synchronous uwb-based location system. Master's thesis, Technical University Graz, 2018.

[57] Ubisense. Ubisense website. `https://www.ubisense.net/`, March 2019. Accessed: 2019-03-30.

[58] uBlox. ublox NEO-6 website. `https://www.u-blox.com/en/product/neo-6-series`, April 2019. Accessed: 2019-04-02.

[59] J. Wang, A. K. Raja, and Z. Pang. Prototyping and experimental comparison of IR-UWB based high precision localization technologies. In *2015 IEEE 12th Intl Conf on Ubiquitous Intelligence and Computing and 2015 IEEE 12th Intl Conf on Autonomic and Trusted Computing and 2015 IEEE 15th Intl Conf on Scalable Computing and Communications and Its Associated Workshops (UIC-ATC-ScalCom)*, pages 1187–1192. IEEE, 2015.

[60] H. Will, T. Hillebrandt, and M. Kyas. The geo-n localization algorithm. International Conference on Indoor Positioning and Indoor Navigation. IEEE, November 2012.

[61] H. Will, T. Hillebrandt, Y. Yuan, Z. Yubin, and M. Kyas. The membership degree min-max localization algorithm. In *2012 Ubiquitous Positioning, Indoor Navigation, and Location Based Service (UPINLBS)*, pages 1–10. IEEE, 2012.

[62] B. Zhen, H.-B. Li, and R. Kohno. Uwb ranging and crystal offset. In *Vehicular Technology Conference, 2006. VTC 2006-Spring. IEEE 63rd*, volume 3, pages 1445–1449. IEEE, 2006.

[63] B. Zhen, H.-B. Li, and R. Kohno. Clock management in ultra-wideband ranging. In *Mobile and Wireless Communications Summit, 2007. 16th IST*, pages 1–5. IEEE, 2007.