CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

A Framework for a Relative Real-Time Tracking System Based on Ultra-Wideband Technology

Master's thesis in Embedded Electronic System Design

Gabriel Ortiz Betancur
Fredrik Treven

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2017

MASTER'S THESIS 2017

A Framework for a Relative Real-Time Tracking System Based on Ultra-Wideband Technology

GABRIEL ORTIZ BETANCUR
FREDRIK TREVEN

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2017

A Framework for a Relative Real-Time Tracking System Based on Ultra-Wideband Technology
GABRIEL ORTIZ BETANCUR
FREDRIK TREVEN

Supervisor: Lars Svensson, Department of Computer Science and Engineering
Supervisor: Sebastian Johansson-Mauricio, Cybercom AB
Examiner: Per Larsson-Edefors, Department of Computer Science and Engineering

Master's Thesis 2017
Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Photograph of Tracking Robot

Typeset in LaTeX
Gothenburg, Sweden 2017

A Framework for a Relative Real-Time Tracking System Based on Ultra-Wideband Technology
GABRIEL ORTIZ BETANCUR
FREDRIK TREVEN
Department of Computer Science and Engineering
Chalmers University of Technology

Abstract

The growing number of applications in automated robots and vehicles has increased the demand for positioning, locating, and tracking systems. The majority of the current methods are based on machine vision systems and require a direct line of sight (LOS) between the tracking device and the target at all times for carrying out the desired functionalities. This limits the possible applications and makes them vulnerable to disturbances. The method presented in this thesis work aims to remove the continuous LOS requirement and allow for an omnidirectional and accurate tracking method using ultra-wideband (UWB) technology. This is achieved by using a *flipped* UWB positioning topology where a set of anchors keeps track of the position of a target and maintains a specific distance from it; this is in contrast to regular indoor positioning systems where a target monitors its own position in relation to a set of fixed references. The feasibility of this solution is shown by a tracking device prototype which demonstrates the capabilities of the proposed system and the UWB technology. The results show that the proposed topology is suitable for positioning, tracking and following applications that require a high degree of accuracy at short distances with the possibility of removing the continuous direct LOS requirement.

Keywords: indoor positioning, tracking, ultra-wideband, object following, autonomous navigation, line-of-sight.

Acknowledgements

We would like to thank our supervisors Sebastian Johansson-Mauricio at Cybercom and Lars Svensson at Chalmers, for their technical advice and guidance during the project. We would also like to thank Gabriel Ibañez, Gustav Lundberg, and Cybercom AB for letting us be part of their group and financially supporting the development of the prototype.

Gabriel Ortiz Betancur:

I would like to thank the Chalmers Foundation for awarding me the William Chalmers Scholarship which allowed me to come to Sweden and carry out my studies. I would also like to thank my parents and my girlfriend for the invaluable support and motivation during the course of the master's program and this thesis.

Fredrik Treven:

I want to thank my parents for their undying encouragement and enthusiasm, my sister for her infectious positivity, and my girlfriend for her patience and palpable confidence in me throughout my time at Chalmers.

Gothenburg, June 2017

Acronyms

AOA	Angle of arrival
BLE	Bluetooth low energy
BN	Blindfolded node
FL	Front left
FR	Front right
LLS	Linear least squeares
LOS	Line-of-sight
NLLS	Non-linear least squares
NLOS	Non-line-of-sight
RF	Radio frequency
RL	Rear left
RN	Reference node
RR	Rear right
RTOF	Round-trip time of flight
TDOA	Time difference of arrival
TOA	Time of arrival
TOF	Time of flight
TWR	Two-way ranging
UWB	Ultra-wideband
WLS	Weighted least squares

Contents

List of Figures xi

List of Tables xiii

1 Introduction 1

2 Background 2
2.1 Tracking Systems . . . 2
2.2 Ultra-Wideband . . . 5
2.3 Project Scope . . . 8
2.4 Societal, Ethical and Ecological Aspects . . . 8

3 Theory 10
3.1 Ranging Technologies . . . 10
3.1.1 Ultra-Wideband Properties . . . 10
3.1.2 Comparison to Similar Technologies . . . 11
3.2 Ranging Methods . . . 13
3.2.1 Received Signal Strength . . . 13
3.2.2 Angle of Arrival . . . 13
3.2.3 Time of Flight . . . 14
3.2.3.1 Time of Arrival . . . 14
3.2.3.2 Time Difference of Arrival . . . 15
3.2.3.3 Two-Way Ranging . . . 15
3.2.3.4 Symmetric Double-Sided Two-Way Ranging . . . 16
3.2.3.5 Asymmetric Double-Sided Two-Way Ranging . . . 17
3.3 Positioning Algorithms . . . 17
3.3.1 Trilateration and Multilateration . . . 17
3.3.1.1 Linear Least Squares . . . 18
3.3.1.2 Weighted Least Squares . . . 19
3.3.1.3 Non-Linear Least Squares . . . 20
3.3.2 Triangulation . . . 21
3.4 Adaptive Filtering . . . 22
3.4.1 Moving Average . . . 23
3.4.2 Exponential Smoothing . . . 24
3.4.3 Kalman Filter . . . 25

4 Implementation 27
4.1 Ranging . . . 28
4.1.1 Hardware . . . 28
4.1.1.1 Central Unit . . . 28
4.1.1.2 UWB Modules . . . 29
4.1.1.3 Target Control Unit . . . 29

4.1.1.4 Components connection . . . 30
4.1.1.5 PCB Design . . . 30
4.1.2 Software . . . 32
4.1.2.1 One-to-one Ranging . . . 32
4.1.2.2 Four Anchors . . . 36
4.1.2.3 Offset Correction . . . 37
4.2 Positioning . . . 39
4.2.1 Multilateration Linear Least Squares . . . 39
4.2.2 Moving Average and Exponential Smoothing . . . 43
4.2.3 Kalman Filter . . . 44
4.3 Movement . . . 48
4.3.1 Movement Algorithm . . . 48
4.3.2 Motor Control . . . 48
4.3.3 Chassis Structure . . . 49

5 Results **50**
5.1 Ranging . . . 51
5.2 Positioning . . . 52
5.3 Tracking . . . 55
5.4 Following . . . 62
5.5 Timing and Power Analysis . . . 62

6 Discussion **65**
6.1 Ranging Implementation and Results . . . 65
6.2 Positioning Implementation and results . . . 66
6.3 Tracking Results . . . 66
6.4 Following Results . . . 67
6.5 Possible Applications . . . 68

7 Conclusion **69**

8 Future Work **70**

9 Bibliography **72**

Appendices **76**

A Circuit Schematics **77**

List of Figures

2.1 Target recognition algorithms . . . 3
2.2 Machine vision tracking devices . . . 3
2.3 Laser-based tracking devices . . . 4
2.4 Basic topology of an UWB indoor positioning system . . . 6
2.5 Flipped anchor topology for tracking robots . . . 7
2.6 Flipped anchor topology implemented in a quadcopter . . . 7

3.1 Power Spectral Density masks of UWB and narrow-band technologies . . . 11
3.2 Accuracy and scale of positioning technologies . . . 12
3.3 Angle of arrival examples: (a) t_b greater than t_a, (b) t_b equal to t_a, and (c) t_b less than t_a. . . 14
3.4 Transmitter positioning using TDOA . . . 15
3.5 TWR message exchange and timing diagram . . . 16
3.6 SDS-TWR message exchange and timing diagram . . . 16
3.7 Two situations of a trilateration solution . . . 18
3.8 Three beacons measuring angles from target reference point in 2-D coordinate system . . . 21
3.9 Values updated in moving average array . . . 23
3.10 Moving average filter length comparison . . . 23
3.11 Exponential smoothing for different values of α . . . 24
3.12 Kalman Filter cycle . . . 25

4.1 Functional block diagram of the system . . . 27
4.2 Hardware block diagram of tracking device . . . 28
4.3 Arduino M0 Pro . . . 29
4.4 Decawave DWM1000 . . . 29
4.5 Arduino Pro Mini 3.3V . . . 30
4.6 DWM1000 UWB module connection to Arduino M0 Pro . . . 30
4.7 Breakout board for the DWM1000 . . . 31
4.8 Arduino shield for central unit and anchor connections . . . 31
4.9 Target device PCB (coin for scale) . . . 32
4.10 Anchor PCB . . . 32
4.11 Software flow chart for ranging function on the anchor side . . . 35
4.12 Software flow chart for ranging function on the target side . . . 36
4.13 Ranging messages exchange . . . 37
4.14 Measurements without offset correction . . . 38
4.15 Measurements with offset correction . . . 38
4.16 Definition of relative coordinate system with respect to the tracking device . . . 39
4.17 Multilateration With Exact Distances . . . 41
4.18 Multilateration With Measured Distances . . . 42
4.19 Simulations of Kalman Filter Performance for Different R Values . . . 46
4.20 Simulations of Kalman Filter Performance for Different Q Values . . . 47
4.21 Olimex BB-L298 dual motor driver board . . . 49
4.22 Physical implementation of the system mounted on the 4WD chassis . . . 49

5.1 Graphical user interface . . . 51
5.2 Example of test setup for ranging functionality . . . 51
5.3 Example of setup for positioning tests . . . 53
5.4 Positioning error for three different system configurations . . . 54
5.5 Example of setup for NLOS positioning tests . . . 55
5.6 Example of setup for tracking tests . . . 56
5.7 Tracking target moving with radius of 160 cm: 40 cm setup . . . 56
5.8 Tracking target moving with radius of 100 cm: 40 cm setup . . . 57
5.9 Tracking target moving with radius of 160 cm: 40 cm setup NLOS . . . 57
5.10 Tracking target moving with radius of 160 cm: 30 cm setup . . . 58
5.11 Tracking target moving with radius of 100 cm: 30 cm setup . . . 59
5.12 Tracking target moving with radius of 160 cm: 30 cm setup NLOS . . . 59
5.13 Tracking target moving with radius of 160 cm: 20 cm setup . . . 60
5.14 Tracking target moving with radius of 100 cm: 20 cm setup . . . 61
5.15 Tracking target moving with radius of 160 cm: 20 cm setup NLOS . . . 61
5.16 Example of setup for following tests . . . 62
5.17 Current drawn by the tracker device . . . 63
5.18 Current drawn by the target . . . 64

A.1 Arduino M0 shield schematics . . . 78
A.2 Target module schematics . . . 79

List of Tables

3.1 Comparison of Positioning Technologies 13

4.1 DWM1000 Configuration parameters 33

5.1 Ranging results for DWM1000 52
5.2 Ranging results for DWM1000 in NLOS scenario 52
5.3 Static positioning statistics for three configurations 54
5.4 Static positioning results for three configurations with NLOS 55
5.5 Following results 62

1
Introduction

The growing number of applications in automated robots and vehicles has increased the demand for positioning, locating, and tracking systems that allow these devices to interact with moving objects in an autonomous and reliable way. This trend can be observed in the more than 35 scientific papers published in the field of automated target tracking since 2015. Most of the studies use machine vision to address this problem [1–6], but complex and performance demanding algorithms have to be applied for locating, tracking, and following the desired target. Other researchers rely on laser scanning techniques [7–10] or sound based systems [11–14] for accomplishing this task. All these methods require a direct line-of-sight (LOS) between the tracking device and the target at all times which limits the possible applications and makes them vulnerable to disturbances. The method proposed in this thesis work aims to remove the continuous LOS requirement and allow for an omnidirectional and accurate tracking method using ultra-wideband (UWB). Research on the UWB radio technology has increased significantly since the year 2000 [15], and since then it has been used in ranging applications due to its high accuracy capabilities and low power consumption. A basic UWB positioning system consists of at least three anchors [16], or reference nodes, that communicate with a target device to calculate its position relative to the anchors.

The goal of this project is to implement a framework for relative real-time tracking using a *flipped* UWB positioning system where a set of anchors calculates the position of a target. The feasibility of this solution will be shown by developing a tracking device prototype which can demonstrate the capabilities of the proposed system and the UWB technology.

There are several benefits of using the flipped UWB system. One such advantage is that the continuous LOS requirement present in most tracking systems and methods can be removed due to the material penetration capabilities of UWB. According to previous work and studies [16] UWB is capable of achieving a higher level of accuracy compared to other technologies. Furthermore, given that the proposed flipped method does not depend on fixed or static reference points, it allows the system to be used in a wide range of environments such as: indoors, outdoors, crowded places, urban, rural, and even in interference-sensitive locations such as hospitals or military premises. There are also certain drawbacks such as the necessity of installing a tag or module in the target object which might not be desired in some applications. Additionally, the UWB technology suffers from its wide bandwidth due to receiving interference from narrow-band communication systems (if a low-complexity receiver is used) as well as needing to adhere to strict regulations in applications [17]. The effects of noise from narrow-band signals on the communication system can lead to diminished performance or the necessity to incorporate complex receivers, but the restrictions on the use of technology are handled by limiting the use of the system to a short distance such as in indoor positioning or in moderate distance tracking such as the application that is being addressed in this project.

The implementation of a framework for a relative time tracking system based on UWB will allow Cybercom AB to develop different applications according to their interests or their clients' requests. There is a wide range of fields where this type of system can be used and, as only one tracking and following system was found in literature using the flipped anchor method along [18], it can contribute to research for this new approach that can be useful to the autonomous robots and vehicles industry.

2

Background

This chapter presents an overview of previous work done in the field of positioning and tracking systems and describes methods, technologies, and applications that have been implemented in the past. Furthermore, it presents a deeper review of previous usage of UWB and other radio-frequency technologies in positioning systems.

2.1 Tracking Systems

The world's most common tracking system is the global positioning system (GPS) [19], which became available to the public in 1993. GPS consists of several satellites orbiting the planet while continuously broadcasting ranging messages to receiver devices on Earth. The ranging message contains a code and a time that allows the device to identify the satellite and a timestamp that allows the device to calculate the time of flight (TOF) of the message. Based on the TOF of messages received from at least 4 different satellites, the device on earth is able to calculate its position.

GPS has not yet found a rival in the field of global and outdoor positioning and tracking. However, it has two significant limitations: accuracy and line-of-sight requirement. GPS services available for public use are usually able to estimate a location within around ± 10 meters of the actual position depending on the device and scenario [16]. The performance of GPS can be improved upon by use of multiple satellites or augmentation systems, but these solutions are often large and costly which limits their use to professional applications [20]. Furthermore, the positioning device must have direct line-of-sight with the four ranging satellites in order to obtain a useful measurement. These disadvantages limit the GPS to outdoor applications where high accuracy (within centimeters) is not needed.

Other technologies have been developed or used for applications where GPS can not fulfill the requirements. These application are usually related to indoor scenarios or high accuracy positioning. Some of these technologies are: machine vision, lasers, ultrasound, WiFi, RFID, UWB, Cellular networks and Bluetooth among others [21].

Optical systems using cameras or machine vision rely on data known a priori and different sets of image processing algorithms to position the object by recognizing different landmarks, markers, shapes or colors in a defined environment. These methods can also be used for tracking an object or person.

Fig. 2.1. Target recognition algorithms [2,3]

Most of the recently presented tracking systems based on machine vision use Microsoft Kinect as their main vision sensor [1–3] or a combination between Kinect and another technology [4–6]. Fig. 2.1 shows examples of the target tracking algorithms used in the cited studies. It can be seen that these types of systems require direct LOS with the target in order track its position. Losing LOS for a small period of time can lead to target loss and functionality failure. Machine vision systems are also sensitive to disturbances such as objects appearing similar to the target, unknown objects, and lighting conditions. This can be the reason why the images shown in Fig. 2.1 are taken in specific scenarios where the target can be easily differentiated from its surroundings.

Machine vision systems also use performance heavy algorithms so the processing units needed for carrying these kinds of tasks are usually computers. This requirement increases the size of the tracking device making it bulky, harder to control, and more expensive. This can be seen in the examples of the devices built in the previously mentioned machine vision studies shown in Fig. 2.2.

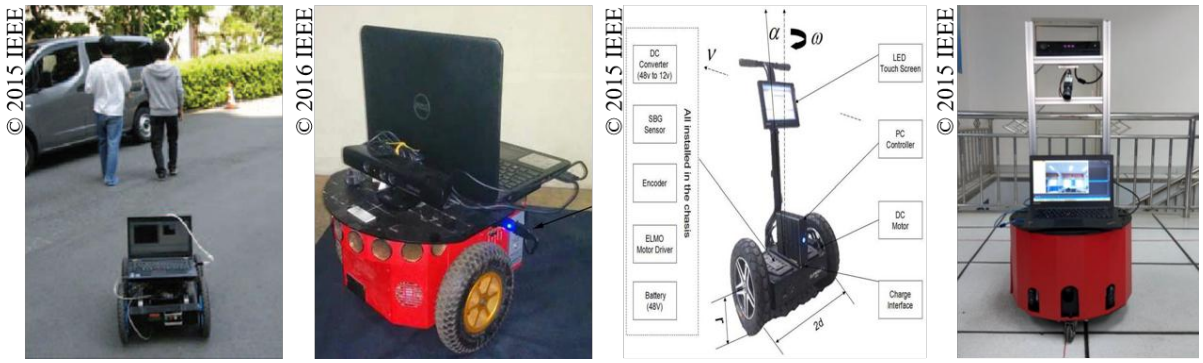

Fig. 2.2. Machine vision tracking devices [2–4,6]

As previously mentioned, it is common that machine vision systems use a combination of different technologies in order to achieve better functionality, accuracy, or reliability. Adding more components to the system will further increase the complexity, size, and cost of the system.

Another technology widely used in tracking applications is laser scanning or ranging. This type of system uses laser emitters and detectors to generate a 3D map of the environment around the robot or tracking device [7–10]. The main advantage of laser scanners compared to machine vision is a more evident obstacle detection as three dimensional objects can be differentiated from the target and therefore avoided. However, given the nature of the data acquired from the laser scan, more complex algorithms are needed to define the target and to keep it in focus while it is in motion.

Laser-based systems suffer from similar disadvantages as machine vision systems. First, as seen in Fig. 2.3, large chassis are needed for mounting the laser scanners and the processing units. Second, LOS

between the scanners and the target is strictly required at all times. Finally, lasers may also be sensitive to disturbances such as obstacles, moving objects, and objects with similar physical characteristics as the target.

Fig. 2.3. Laser-based tracking devices [7, 8, 22]

Fewer studies use sound or ultrasound as a medium for positioning objects [11–14]. These studies use transceivers to find the distance between a set of references and a target. Then they use a multilateration algorithm to find the position of the target in relation to the references. Although the topologies are similar the implementations and results vary from one study to another. The accuracy obtained with this kind of technology varies from a few centimeters to one meter [23], and it can be deduced that the more complex the signal conditioning and processing, the better the accuracy. Furthermore, the LOS requirement remains present in ultrasound-based systems, and the orientation of the transceivers limits performance due to their narrow beams.

Until now, three different technologies or methods have been analyzed: machine vision, lasers and ultrasound. These technologies share most of the same drawbacks: LOS requirement, high complexity, performance heavy algorithms, large size implementations, disturbance sensitive, and high cost. In order to solve some of these problems, researchers and engineers have started using different radio frequency (RF) technologies such as WiFi, ZigBee, RFID, BLE, and UWB.

RF positioning uses the time that takes an electromagnetic wave to travel from one transceiver to another to calculate the distance between the two. The speed of an electromagnetic wave is known so the distance between the two transceivers can be calculated using basic arithmetic operations. However, as the time of flight of the electromagnetic waves is in the order of nanoseconds, very complex and precise systems must be implemented if an accurate result is to be obtained.

WiFi is the most commonly used RF technology nowadays with almost every office, house, hospital or school equipped with a WiFi network. Therefore, WiFi has been the most widely used technology for indoor positioning applications during the past three or four years. Only in 2016, 98 articles with the keywords "WiFi indoor positioning" were published in the IEEEXplore library. Studies presented since 2000 show that WiFi doesn't have the best accuracy compared to other RF technologies with an average of 4 meters of error when observing all the presented results [23]. This is probably due to the fact that most WiFi systems rely on signal strength rather than on time of flight ranging methods (see Section 3.2).

Other RF technologies such as Bluetooth, BLE, ZigBee, and RFID can be grouped into the same category. Other than in some special cases time of flight methods are used for ranging between the reference modules and the target. The accuracy obtained with these technologies ranges from 1 to 20 meters depending on the type and complexity of the implementation [16, 23, 24].

The positioning methods based on RF technologies reviewed until now have several advantages such as:

small size, low power, low sensitivity to disturbances (non-electromagnetic interference), the ability to perform positioning and tracking of targets simultaneously, and non-line-of-sight (NLOS) functionality under certain conditions. Furthermore, the complexity of the system reduces as the technology ages, and the ranging functionality is embedded in the transceivers and can be seen as a black box to the end user. Nevertheless, there are other disadvantages such as the need for several transceivers, lack of complete NLOS functionality, sensitivity to electromagnetic interference (EMI), low accuracy (compared to machine vision systems), and, finally, the use of fixed reference points which limit the application to specific rooms or pre-arranged environments.

2.2 Ultra-Wideband

UWB made its first appearance in late 1990's, and since then it has been used in several fields such as communications, positioning, and radar. Due to its extremely large bandwidth, it allows a very fine delay resolution and, as a result, better accuracy can be achieved compared to other RF technologies [25]. Other benefits of UWB include: robustness to fading, enhanced obstacle penetration, interference rejection, low cost, and coexistence with other narrow bandwidth systems. However, because of strict regulations on the use of UWB, the transmission power is limited, restricting the range of application to approximately 50 meters in normal conditions.

Studies in UWB can be divided into two broad groups: technology studies and application studies. Technology studies are related to new research focused on improving the functionality of UWB systems and usually deal with topics such as antennas, hardware, signal processing, protocols, and transmission channels. Application studies focus on the use of UWB in different scenarios and topologies that either improve the results of previous designs or allow for new uses of the technology. This section describes some of the application studies and how they have changed over time.

The first application studies published around 2004 describe possible applications and topologies where UWB could be used [26–29]. Previous technology development research had shown the capabilities of UWB [30, 31], but the technology and the transceivers were still too complex and expensive for most researchers and the public. Chu and Ganz [29] present "a UWB-based 3D location system for indoor environments" and prove feasibility by simulating the environment and the location scheme.

Most of the actual UWB applications started appearing after 2009 when developers started building their own transceivers and began testing the actual capabilities of the technology, specifically in the field of indoor positioning. A basic topology for an indoor positioning system is presented in [32]. As shown in Fig. 2.4 a target UWB transceiver is surrounded by four other transceivers (usually called anchors) that find the distance by interchanging messages with the target and measuring the time of flight or the received signal strength. In this case, the four anchors are connected to a host PC which is in charge of running the positioning algorithms for finding the location of the target in relation to the anchors. Other studies published regarding indoor positioning with UWB systems aim to improve the functionality in NLOS conditions [33] or improve the accuracy of the system [34].

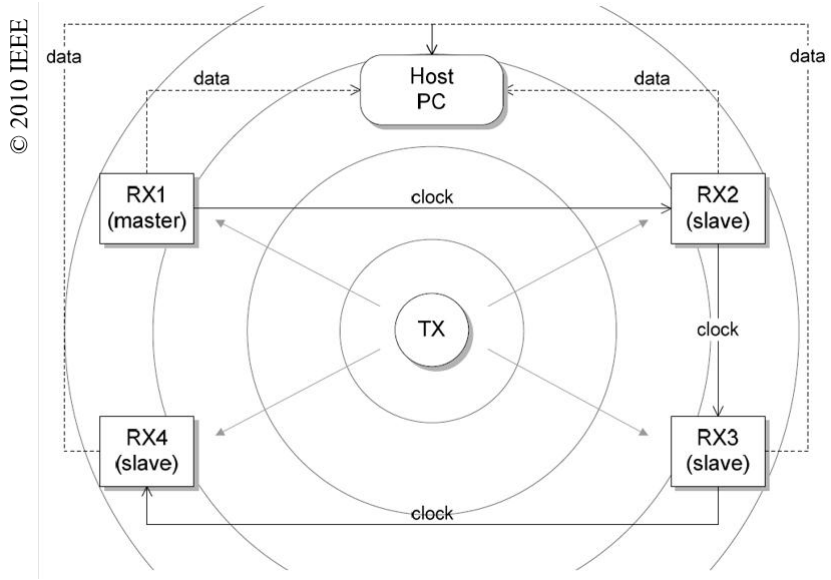

Fig. 2.4. Basic topology of an UWB indoor positioning system [32]

Apart from indoor positioning, other applications have been found for UWB by taking advantage of its high accuracy ranging property. Studies in the medical field have used UWB for positioning surgical instruments and other high precision medical devices [35, 36]. In the automotive industry UWB can be used (amongst other methods) for precisely parking electric vehicles in order to enable wireless charging [37]. Autonomous robots are using UWB to find their way through offices or factories [38], and biomechanical engineers are using this approach for determining body position based on the measurements from several UWB tags placed on different parts of the body [39].

In recent years more and more UWB systems have become available in the market with the most noticeable being UBISENSE [40] and ZEBRA [41] who sell already developed indoor positioning systems for different applications. Other manufacturers, like DECAWAVE [42] and Bespoon [43], have focused their market on low-cost UWB transceivers that allow researchers and developers to implement their own system without having to deal with the complex RF hardware and signal management.

On average, studies have presented an accuracy of around 30 cm for UWB with the best case being 1 cm accuracy and 150 cm in the worst case [23]. These results, together with the proposed applications and the large quantity of publications in the field prove the potential of UWB technology and show that there are still more applications to develop and new problems to solve by using UWB. The application proposed in this thesis aims to give indoor positioning topologies a new perspective by locating the anchors in a small device or platform, while tracking a target (or several) that move(s) around this platform.

After an extensive literature review, only two publications that implement a similar topology to the one proposed in this thesis could be found. In addition, Alarifi et al. [44] presented a review of recent advances in UWB positioning technologies in May 2016 where 39 publications are analyzed and none of them include a system as the one proposed here.

The first reference to the the flipped tracking system was presented in 2010 by Cheok et al. [45]. The aim of the study was to develop an UWB-based local positioning system (LPS) for tracking mobile robots within a radius of 100 meters. As can been seen in Fig. 2.5. The base station consists of four arms with a UWB transceiver mounted on each one. These find the distance to a transceiver mounted on the robot and triangulate its position. Even though the principle proposed by Cheok et al. is the same as the one proposed in this thesis, it can be observed that the base station is still several meters wide and it is impractical for the desired application.

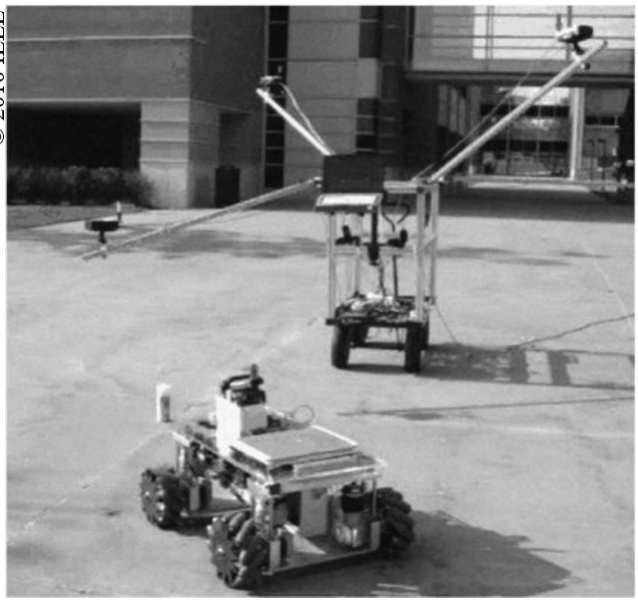

Fig. 2.5. Flipped anchor topology for tracking robots [45]

The second reference was presented in October 2016 (published December 2016) by Hepp et al. [18]. In this work the authors placed four UWB transceivers in a quadcopter in order to track and follow a target consisting of another UWB transceiver. Fig 2.6 shows a diagram of the implementation. The authors developed a ranging scheme in order to obtain a high enough data rate to control the quadcopter. In this scheme, the *master* performs a symmetric two-way ranging with the *slave* while the *listeners* passively receive messages from the ranging protocol. The communication between the *master* and *listeners* can limit the minimum size of the implementation, as the UWB communication of the transceivers is affected when placed too close to one another.

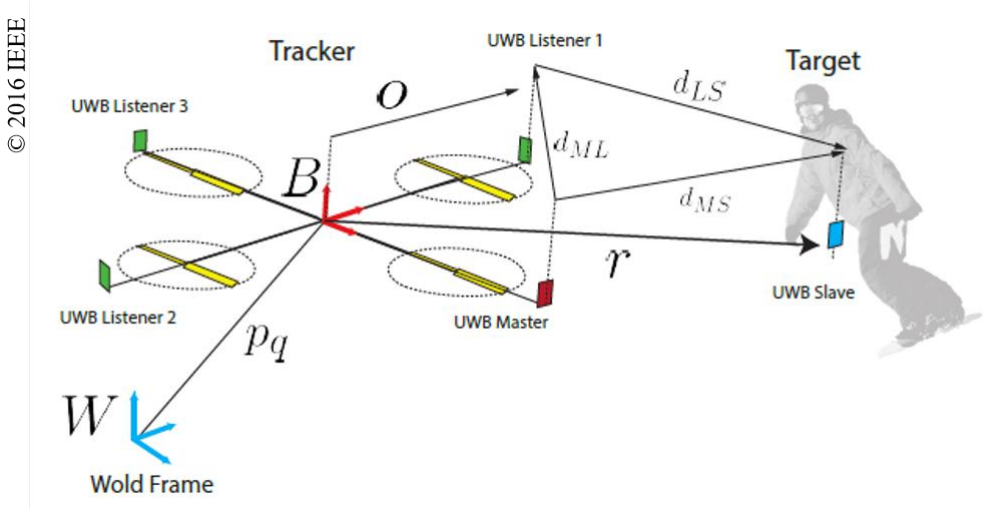

Fig. 2.6. Flipped anchor topology implemented in a quadcopter [18]

The system proposed in this thesis aims to remove the continuous LOS requirement present in all machine vision implementations by using UWB technology. Furthermore, attempts to remove the fixed references used in regular indoor positioning by replacing them with four anchors mounted on a small

platform. This configuration can be used as a following robot that tracks and follows a target or just a portable high accuracy tracking system. Such a system can be placed in various settings (outdoors and indoors) to provide accurate positioning of objects or persons around it.

2.3 Project Scope

The scope of the project states explicitly what was intended to be covered during the course of the thesis and what the final report includes, and, just as importantly, what it does not include. The following is a list of tasks that are firmly within the scope of this thesis and were completed by the deadline.

- Research on how well the UWB technology handles the task of target tracking including specific descriptions on performance in terms of accuracy, precision, complexity, robustness, scalability, and cost.
- Displayed capabilities of a system using the flipped method for interpreting the position of a target relative to the device containing four anchor modules.
- A mobile robot prototype which has the ability to successfully track and follow an object with the use of the proposed flipped UWB method. This includes removing LOS between the anchors and the target and demonstrating that the performance remains sufficient.

The capabilities of UWB are discussed in the thesis, but there were no other technologies that were tested to be compared outside of research already performed by others. This is due to the fact that the thesis focuses only on what can be accomplished with UWB and not on how the same task can be performed with other similar technologies.

The largest limiting factors throughout the thesis work (as they often are) were time and money. If more time was available, some of the desired expanded functionality that is important for such a device to be marketable could have been implemented. This includes obstacle avoidance and collision detection, and improvement upon NLOS performance. Furthermore, if a greater amount of resources were available, the prototype perhaps could have been extended to a product that may have been commercially viable. As it stands, the robot is simply a proof of concept that demonstrates the potential capabilities of such a system.

2.4 Societal, Ethical and Ecological Aspects

The development of new electronic devices or models most of the time (if not always) has an impact on society. This impact can be beneficial or detrimental depending on the final application of the device. Possible ethical quandaries related to how the UWB tracker may affect society are presented in this section.

Some of the applications proposed for the tracking device are in the military field. Although there is a possibility that the tracker is used for harming people or destroying structures, it can also enable unmanned military operations that will reduce the number of casualties in different armed conflicts around the world. These applications include: mine-sweeping, ground and air recognition, mapping and surveillance etc.

The tracking and following capabilities of the proposed system can also lead to personal privacy dilemmas. In a worst case scenario, a device could be used for tracking, following or monitoring a specific person's actions to the point of perturbing the privacy of the individual. However, the need for a UWB tag to be installed on the target makes this situation less probable with this type of system than with other machine vision methods. Another application where society can be affected is in the field of vehicle electronics. The tracking system can be used for positioning a vehicle in relation to other vehicles or objects. In this case, safety plays a very important role and the system should be designed in a reliable

and robust way with fault tolerant and failsafe capabilities.

Due to the nature of the project and the components selected it may be necessary to interact with open source IP blocks; it is important to read and understand the licenses for use and reproduction of each block and take the necessary precautions before including them in the work. It is our responsibility to state clearly which part of the project was developed using IP blocks.

From an ecological point of view, care should be taken to use only RoHS compliant components, reduce the number of parts in the system, and use recyclable materials whenever possible. Reducing power consumption is also a way of mitigating environmental impact, and this has been taken into account during the development of the project.

3

Theory

This chapter compares UWB to other technologies used to perform ranging, and analytically explains how the properties of UWB are attained as well as how these attributes are useful when performing positioning. Furthermore, the chapter evaluates several different approaches to achieving the goal of creating a mobile tracking device that can accurately follow a target. The different methods of ranging, positioning, and filtering are investigated from a theoretical perspective in order to give background as to which mechanism should be used in the implementation of the solution. Ultimately, asymmetric double-sided two-way ranging by way of time of flight was implemented. This along with multilateration with linear least squares, and a basic Kalman Filter make up the final design. This chapter gives explanations of other methods in order to give a clear background on why certain choices were made.

3.1 Ranging Technologies

In this section the properties of UWB are discussed along with a comparison to other technologies which have been popular for similar applications. Positive and negative aspects of UWB are evaluated and contrasted with other short range communication systems as well.

3.1.1 Ultra-Wideband Properties

UWB derives its name from its most obvious characteristic: the large amount of bandwidth occupied by the transmitted signals. UWB is defined as any transmission that has a fractional bandwidth larger than 0.2 or which occupies a bandwidth of more than 500 MHz [46]. Fractional bandwidth is defined in (3.1) where f_H and f_L are the highest and lowest frequencies of the transmission respectively.

$$B_{frac} = \frac{2(f_H - f_L)}{f_H + f_L} \tag{3.1}$$

In order to use such a large bandwidth in cooperation with pre-existing narrow-band communications some fairly strict limits are placed upon UWB in terms of signal power. This is done so that there is minimal interference with other communication protocols, and it prevents use over long (>100 m) distances. A comparison of the power spectral density (PSD) of UWB and other technologies is shown in Fig. 3.1

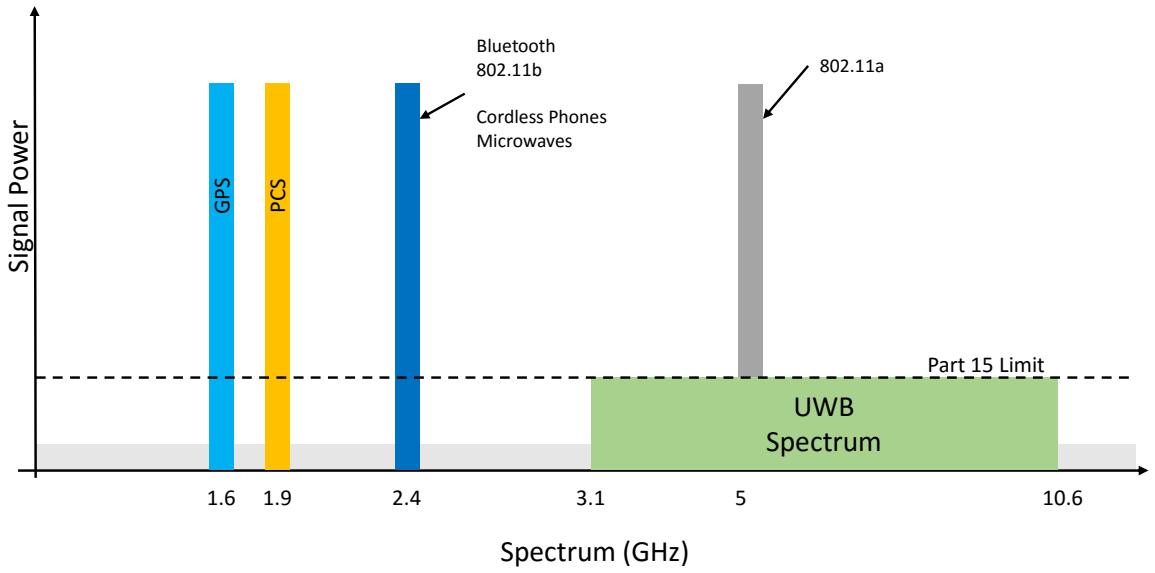

Fig. 3.1. Power Spectral Density masks of UWB and narrow-band technologies [47]

The figure shows that UWB has a much larger bandwidth than the other technologies but also operates at a significantly lower power level (under the Part 15 Limit: a regulation set forward by the FCC in the United States to restrain power of these types of communication methods). The large bandwidth of the technology allows for very high channel capacity leading to high data rates as shown by Shannon's capacity equation in (3.2) where B is the bandwidth, C is channel capacity, and SNR is the signal-to-noise ratio.

$$C = B \cdot \log(1 + SNR) \tag{3.2}$$

According to (3.2), bandwidth and channel capacity are directly proportional so that, in theory, larger bandwidth will always lead to higher channel capacity. UWB achieves this large bandwidth by utilizing extremely short pulses ($\sim$1.5 ns) to send information which also allows for high material penetration and minimal distortion leading to multipath diversity and high-accuracy ($<$30 cm error) ranging [48]. Data transmission using UWB can be done without the use of a carrier frequency due to the fact that the bandwidth itself covers the frequency range in which a carrier frequency is generally used. Due to this property the need for an RF mixing stage is removed allowing data sending to avoid the use of up/down sampling. Therfore, the entire UWB transceiver can be integrated as a single-chip CMOS without the need for a high-complexity receiver [48]. This implementation leads to the low cost, small size, and low power requirements for UWB modules.

UWB can be considered a spread-spectrum technology with a very high spreading factor and can thus use common spread-spectrum approaches such as frequency hopping, orthogonal frequency division multiplexing (OFDM), direct-sequence spread spectrum using code division multiple access (CDMA), and time-hopping impulse radio which applies quadrature amplitude modulation (QAM) [24].

3.1.2 Comparison to Similar Technologies

Ideal usage of UWB involves short distance ranging and positioning due to the restrictions placed on signal power. Some other technologies that are often used in similar applications are WiFi, Bluetooth, infrared (IR) sensors, and Zigbee networks. Methods for ranging over larger distances include GPS and

mobile cellular networks. In order to make a comparison between UWB and these other methods it is important to have a set of parameters that should be measured. The parameters deemed most important for location technologies are accuracy, precision, and cost [16], and they are defined in the following list.

- **Accuracy** is the value of mean distance errors when ranging with a certain communication protocol. Accuracy determines how close the measurement comes to the true location of the target.
- **Precision** is used to determine how consistently the technique returns similar values in similar circumstances. It is possible that a high maximum accuracy can be reached, but this does not mean that the solution is feasible unless that accuracy can be achieved consistently.
- **Cost** takes into account the monetary requirements to get proper functionality out of a proposed ranging technique, the time it takes to set up a system to use the technology effectively, the size and weight of the solution, and the energy expended by such an implementation.

Liu and Darabi performed a survey of wireless positioning techniques and systems in 2007, and the results they found are laid out in Fig. 3.2.

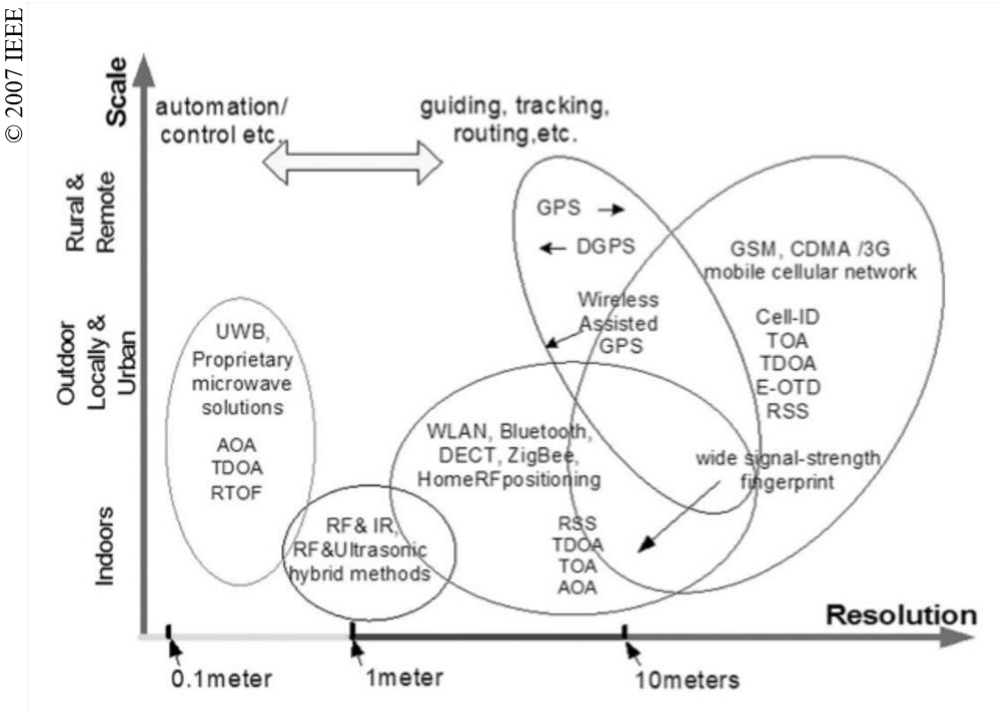

Fig. 3.2. Accuracy and scale of positioning technologies [16]

Fig. 3.2 shows the accuracy of various positioning technologies including UWB and their scale in terms of outdoor or indoor positioning use cases. From the information it is gleaned that UWB using angle of arrival (AOA), round-trip time of flight (RTOF), or time difference of arrival (TDOA) can pinpoint an object with higher accuracy than other techniques used for local positioning. Table 3.1 shows statistics about several different positioning technologies for key parameters when choosing a technique to implement a tracking solution [16, 24, 49, 50].

Table 3.1 – Comparison of Positioning Technologies

Technology	Accuracy	Precision	Complexity	Cost	Power	Range
UWB	<30cm	99% within 30 cm	Application Based	Application Based	30mW	<30m
2.4 GHz Zigbee	>2m	Up to 99%	Low	Low	20mW - 40mW	<30m
2.4 GHz WiFi	>2m	Depends on standard	High	High	500mW - 1W	<100m
IR	<1m	50% within 1m	Medium to High	Medium to High	High Variation	20m - 30m
Bluetooth	2m	95% within 2m	Medium	Medium	60 mW	30m - 50m

Table 3.1 shows that UWB stacks up favorably in terms of accuracy, precision, and power against other popular methods for ranging over short distances. The complexity and cost parameters often depend on which purpose the technology is being used for. For example, if the device is required to support multi-user capacity a higher complexity receiver with more sophisticated coding techniques will be required in order to resolve multipath energy [48]. Furthermore, the low limit on power reduces the effective range at which UWB can be useful and, consequently, the applications for which it can be used. Taking these critical metrics into account it was decided that UWB would be the most effective communication method to use for the short distance tracking required for this project. The research showed that UWB has the best accuracy of any other local positioning method currently available while maintaining low power consumption and high precision.

3.2 Ranging Methods

This section describes some of the methods used for measuring distance by means of RF technologies. There are three main parameters upon which these methods are based: signal strength (power), AOA, and TOF.

3.2.1 Received Signal Strength

This method is based on the assumption that the following parameters are known [51]:

1. The transmitted signal power.
2. The received signal power.
3. The relation between distance and power loss.

With this information it is possible to calculate the distance at which the transmitter is from the receiver by processing the received signal power. Although this is one of the simplest methods for obtaining a distance measurement, the transmitted signal suffers from path-loss which generates inaccuracies in the distance to power loss relation and, as a result, in the distance measurement [52]. Complex algorithms must be implemented in order to mitigate the effects of the mentioned phenomenons in the resulting measurement.

3.2.2 Angle of Arrival

This method differs from the others presented in this section as the obtained result is an angle and not a distance. This approach is of interest for this thesis because this angle can be used for positioning an

object in space since it represents the direction from where the signal is coming from. A basic angle of arrival (AOA) system consists of an anchor with two or more antennas (antenna array) [51]. The anchor measures the time of arrival at each of the antennae and then calculates the angle from where the signal was emitted. Fig. 3.3 shows an example of how an anchor can find the angle of arrival given the times of arrival t_a and t_b. A larger time difference means a larger angle.

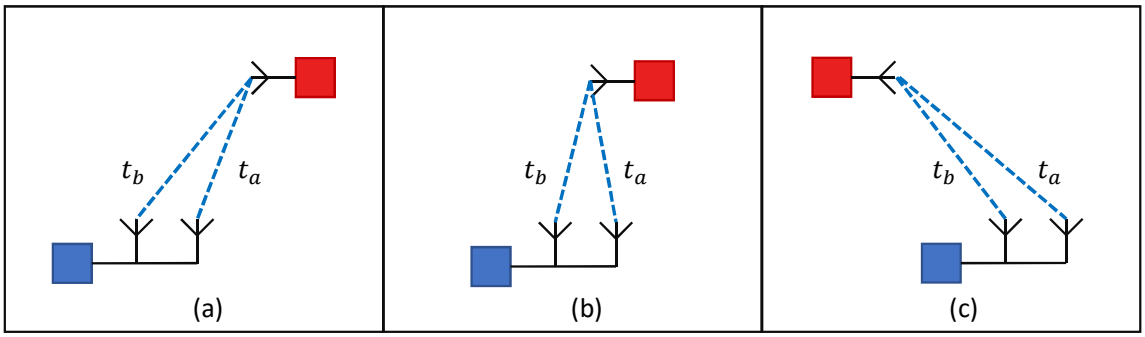

Fig. 3.3. Angle of arrival examples: (a) t_b greater than t_a, (b) t_b equal to t_a, and (c) t_b less than t_a.

Given the proximity of the antennae in the array, the TDOA is usually very small and needs a highly accurate system in order to generate valid results. In addition, adding more antennae to each of the anchors increases both the cost and the complexity of the system.

3.2.3 Time of Flight

Time of flight (TOF) is one of the most commonly used metrics for ranging and positioning when using RF technologies. It is based on the time an electromagnetic wave takes to travel a certain distance, in this case, between a transmitter and a receiver. With the propagation speed of the wave (c) it is possible to calculate the distance at which the transmitter is located from the receiver using (3.3). Due to the signal's high speed, the time of flight range is usually in nanoseconds and advanced systems are needed for measuring it. Furthermore, small clock drift or timing errors can lead to significant deviations from the actual distance. Several methods use TOF as a metric for calculating distance and include different setups or message interchange protocols in order to mitigate the error in the measurements [53]. Some of these methods are described in the following sections.

$$distance = TOF \cdot \mathrm{c} \tag{3.3}$$

3.2.3.1 Time of Arrival

Time of arrival (TOA) is the simplest time based method used for calculating a distance between two RF devices. As its name indicates, it measures the time at which a message arrives at the receiver. The time at which the message left the transmitter (t_s) is subtracted to the arrival time (t_r) obtain the TOF as in (3.4). The transmission time can be defined a priori or embedded in the message, either way, the transmitter and the receiver clocks must be synchronized to be able to subtract the transmitted and received times. The resulting TOF is then multiplied by the speed of light to obtain the distance between the two devices [51].

$$TOF = t_r - t_s \tag{3.4}$$

3.2.3.2 Time Difference of Arrival

The time difference of arrival (TDOA) method uses more than one receiver with synchronized clocks as opposed to TOA where only one set of synchronized receiver and transmitter is used. The synchronized receivers measure the time at which a message arrives to each of them and then the time difference is calculated [53].

From the differential time measurements (Δt) of every pair of receivers it possible to construct hyperbolas with foci at the receivers, as shown in Fig. 3.4. The transmitter is located at a point where two or more hyperbolas intersect.

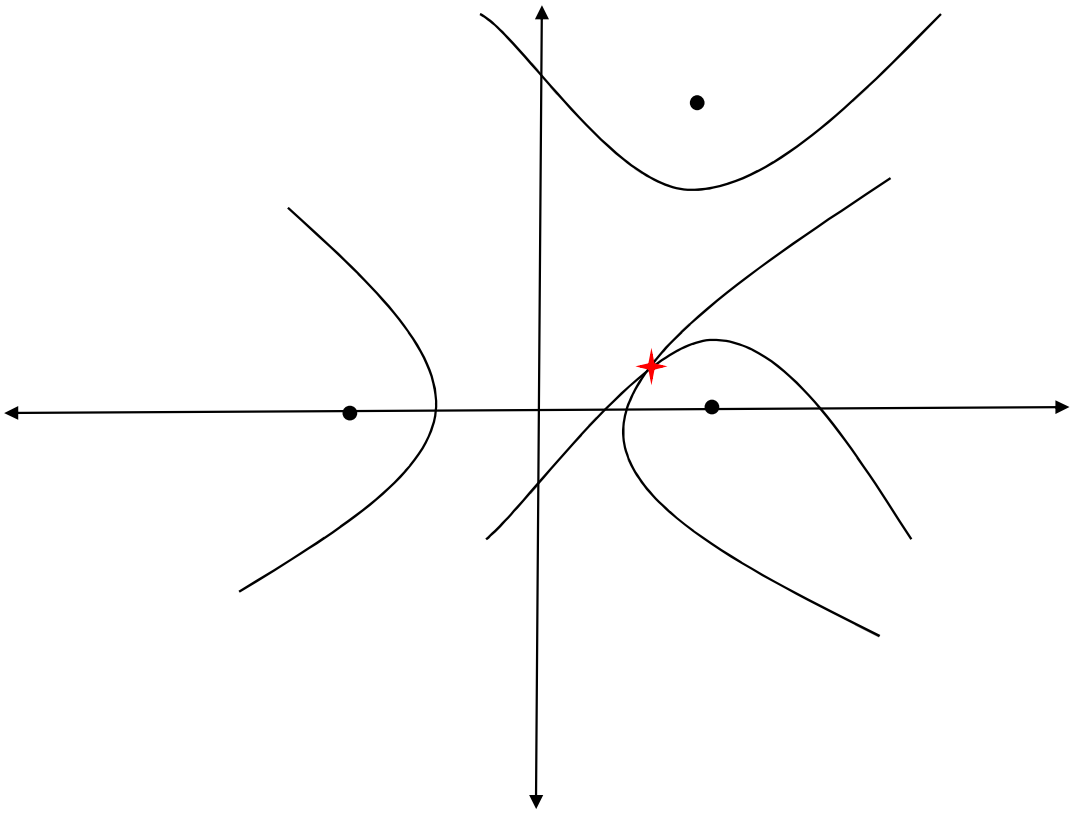

Fig. 3.4. Transmitter positioning using TDOA

3.2.3.3 Two-Way Ranging

A system using two-way ranging (TWR) consists of at least two or more RF transceivers that interchange messages between each other [54]. In contrast to the methods presented before, TWR does not require any type of synchronization between the transceivers which makes the implementation easier in some application scenarios. Fig. 3.5 shows an operation diagram of the TWR method between two devices.

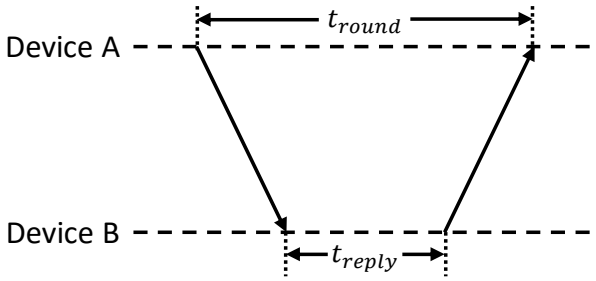

Fig. 3.5. TWR message exchange and timing diagram

The ranging protocol is initiated in device A with a poll message. Device B receives the poll message and replies with an acknowledge message. t_{reply} is the time it takes from the moment the message is received to when the acknowledge message is sent. This time can be embedded in the acknowledge message (if previously defined) or sent in an additional message. Once device A receives the acknowledge message, it can calculate the round trip travel time (t_{round}). The TOF between the two transceivers can then be calculated using (3.5).

$$TOF = \frac{t_{round} - t_{reply}}{2} \tag{3.5}$$

The TWR method suffers from high sensitivity to clock drift, specifically from the replying device. As $t_{reply} >> TOF$ small variations in the clock can generate errors in the TOF estimation that vary between 0.1 ns and 200 ns depending on the tolerance of the crystal and the duration of the reply time [54]. This error in the TOF can deviate the distance calculation from centimeters to several meters.

3.2.3.4 Symmetric Double-Sided Two-Way Ranging

This method aims to mitigate the clock drift effect from the basic TWR by adding an additional message to the exchange protocol. A message exchange and timing diagram for a symmetric double-sided two-way ranging system is shown Fig. 3.6.

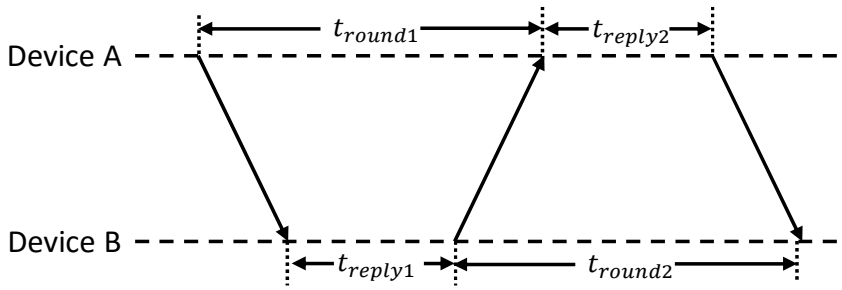

Fig. 3.6. SDS-TWR message exchange and timing diagram

As can be observed, the new message allows the system to have two round trip times and two reply times. If the reply times are equal it is possible to subtract the reply time from the round trip time that was measured with the same clock as presented in (3.6). Using the same time bases for calculating the time of flight reduces the error generated by the crystal tolerances in each of the transceivers' clocks [54].

$$TOF = \frac{(t_{round1} - t_{reply2}) + (t_{round2} - t_{reply1})}{4} \tag{3.6}$$

3.2.3.5 Asymmetric Double-Sided Two-Way Ranging

Since different transceivers do not always have the same reply time the asymmetric double-sided two-way ranging (ADS-TWR) method was introduced [55]. This new method deals with the problem of different reply times and achieves the same minimum error as in SDS-TWR, without increasing the number of messages in the protocol. The TOF is calculated using (3.7).

$$TOF = \frac{(t_{round1} \cdot t_{round2}) - (t_{reply1} \cdot t_{reply2})}{t_{round1} + t_{reply1} + t_{round2} + t_{reply2}} \tag{3.7}$$

3.3 Positioning Algorithms

After ranging is performed and angles or distances have been calculated these values must be interpreted in a way that can predict the location of the target. In this section various methods and algorithms that can be used in order to perform sufficient tracking are discussed.

3.3.1 Trilateration and Multilateration

Trilateration is a method of positioning which involves measuring distances between the object to be tracked and three beacons known as the reference nodes (RNs) each of which has a location known in relation to one the other RNs. Three of these nodes are required in order to track an object, often referred to as the blind-folded node (BN), in 2-D coordinates. Trilateration uses exactly three of these beacons in order to express the location of the tracked target in terms of relative coordinates. Multilateration utilizes a similar technique but is extended to more than three anchors. The method of making the position estimate is almost identical other than a few linear algebra operations differing. The reason for using more than three RNs stems from the desire to diminish the mean square error which occurs when, inevitably, the distance measures from RN to BN are not exact. When more than three anchors are used the system of equations to solve becomes overdetermined leading to to better results in position estimation [56].

To perform the lateration process, the distances between each RN must be known and the distance from each anchor to the target must be measured. When attempting to find the position of the target, some reference points must be used to define a relative coordinate system characterized by the known locations of the reference nodes. Fig. 3.7 gives graphical representations of how trilateration is used to locate a target.

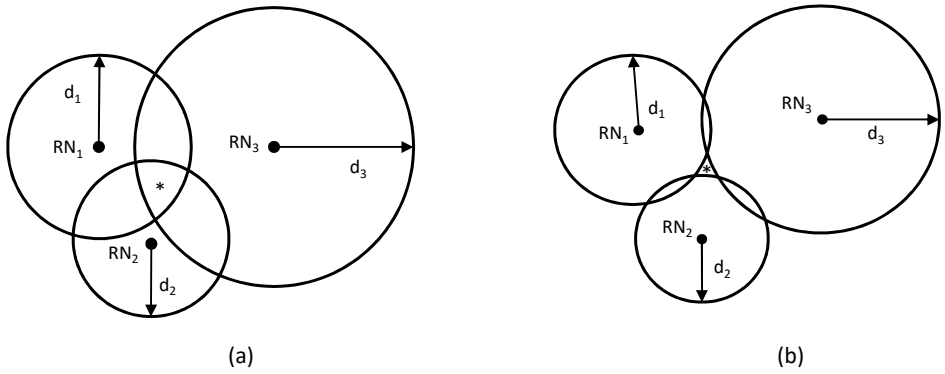

Fig. 3.7. Two situations of a trilateration solution

Each distance is taken to be a radius of a circle, and where these circles intersect determines the most likely zone of location for the target. In Fig. 3.7 situation (a) shows a case in which the most likely zone is inside the created circles whereas situation (b) shows that shaded zone is exterior to all circles. This same approach is expanded to multiple circles when multilateration is used [57].

The following algorithm can be performed to estimate the position of the blindfolded node. In Cartesian coordinates the distance between the i_{th} reference node and the target position (given by coordinates x and y) in two dimensions is

$$d_i = \sqrt{(x - x_i)^2 + (y - y_i)^2}. \tag{3.8}$$

It is desired to find a set of equations that can lead to solving for the unknown coordinates linearly. To set up the equation in this way both sides of (3.8) are squared to get

$$d_i^2 = (x - x_i)^2 + (y - y_i)^2 = x^2 - 2x \cdot x_i + x_i^2 + y^2 - 2y \cdot y_i + y_i^2 \tag{3.9}$$

From (3.9) it is required to remove the squared terms of the target position: x^2 and y^2. One method of doing this is simply subtracting the final squared distance calculation (d_N^2) which itself contains these terms. Doing this yields

$$d_i^2 - d_N^2 = x^2 - 2x \cdot x_i + x_i^2 + y^2 - 2y \cdot y_i + y_i^2 - (x^2 - 2x \cdot x_N + x_N^2 + y^2 - 2y \cdot y_N + y_N^2) \tag{3.10}$$

which simplifies to

$$d_i^2 - d_N^2 = -2x(x_i - x_N) + x_i^2 - x_N^2 - 2y(y_i - y_N) + y_i^2 + y_N^2. \tag{3.11}$$

Equation (3.11) gives a system of N equations which can be solved in a number of ways with varying complexity and predictive accuracy. Some of these methods are discussed in the following subsections.

3.3.1.1 Linear Least Squares

The linear least squares (LLS) method is the simplest way to solve the positioning algorithm's set of linear equations [58]. The fundamental linear algebra equation extrapolated from (3.11) is

$$\mathbf{b} = \mathbf{A} \cdot \mathbf{p} \tag{3.12}$$

where

$$\mathbf{A} = -2 \cdot \begin{bmatrix} x_1 - x_N & y_1 - y_N \\ x_2 - x_N & y_2 - y_N \\ \vdots & \vdots \\ x_{N-1} - x_N & y_{N-1} - y_N \end{bmatrix},$$

$$\mathbf{b} = \begin{bmatrix} d_1^2 - x_1^2 - y_1^2 - d_N^2 + x_N^2 + y_N^2 \\ d_2^2 - x_2^2 - y_2^2 - d_N^2 + x_N^2 + y_N^2 \\ \vdots \\ d_{N-1}^2 - x_{N-1}^2 - y_{N-1}^2 - d_N^2 + x_N^2 + y_N^2 \end{bmatrix}, \text{and}$$

$$\mathbf{p} = \begin{bmatrix} x \\ y \end{bmatrix}.$$

The goal is to solve for the **p** matrix which contains the predicted coordinates of the BN. In trilateration, when exactly 3 anchors are used (N=3), the system is solved by matrix inversion and multiplication

$$\mathbf{p} = \mathbf{A}^{-1}\mathbf{b}. \tag{3.13}$$

When the system of equations is overdetermined as with multilaterion (N>3) the same procedure is followed but, since there is no inverse of a non-square matrix, the pseudo-inverse must be used

$$\mathbf{p} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}. \tag{3.14}$$

3.3.1.2 Weighted Least Squares

The linear least squares method assumes constant error variance across all measurements [59]. When this is not the case, as it may be for sensor variations or malfunctions, LLS will lead to erroneous or low confidence positioning predictions. To remedy such an issue the Weighted Least Squares (WLS) method can be used. This algorithm incorporates a diagonal matrix that takes into account the variance of the measurements from each node and mitigates positioning inaccuracies by assigning higher weights to RNs with better precision. The matrix, **W**, uses weights from each node, i, defined as $w_i = 1/\sigma_i^2$ where σ is the standard deviation measured from the node and σ^2 is the variance. These weights are placed on the diagonal and for a system with N reference nodes the matrix is defined as

$$\mathbf{W} = \begin{bmatrix} w_1 & 0 & \cdots & 0 \\ 0 & w_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_N \end{bmatrix}.$$

WLS then applies **W** to give larger significance to anchors with more stable measurements by augmenting the LLS equation to

$$\mathbf{p} = (\mathbf{A}^T\mathbf{W}\mathbf{A})^{-1}\mathbf{A}^T\mathbf{W}\mathbf{b}. \tag{3.15}$$

As the weight given is the inverse of the variance, a node with high variance will have less impact on the result than one with more stable measurements. The drawbacks to this method are that variance must be either calculated or estimated which may introduce further inaccuracies depending on the dependability of these approximations. However, WLS gives improvement over LLS when certain nodes are not functioning as well as they should be [59].

3.3.1.3 Non-Linear Least Squares

The Non-Linear Least Squares (NLLS) approach extends the use of LLS to functions of almost any class as long as they can be expressed in closed form equations [60]. To explain this process it is recalled that the fundamental problem that needs to be solved is to minimize the error between the estimated distance and the true distance. The error for each i_{th} node can be expressed as

$$f_i(x,y) = d_i - \hat{d}_i = \sqrt{(x-x_i)^2 + (y-y_i)^2} - \hat{d}_i \tag{3.16}$$

where $\hat{d}_i$ is the distance measured and d_i is the actual distance. The sum of these errors squared is what needs to be minimized and this is shown as

$$F_i(x,y) = \sum_{i=1}^{N} f_i(x,y)^2. \tag{3.17}$$

To find a minimum the derivative is taken with respect to all dimensions; performing a partial derivative with respect to x of (3.18) yields

$$\frac{\partial F(x,y)}{\partial x} = 2\sum_{i=1}^{N} f_i(x,y)\frac{\partial f_i(x,y)}{\partial x}, \tag{3.18}$$

and this is done for the y direction as well. The matrix known as the Jacobian can then be defined as

$$\mathbf{J} = \begin{bmatrix} \frac{\partial f_1(x,y)}{\partial x} & \frac{\partial f_1(x,y)}{\partial y} \\ \frac{\partial f_2(x,y)}{\partial x} & \frac{\partial f_2(x,y)}{\partial y} \\ \vdots & \vdots \\ \frac{\partial f_N(x,y)}{\partial x} & \frac{\partial f_N(x,y)}{\partial y} \end{bmatrix},$$

and a one dimensional array for each error calculation is given to be

$$\mathbf{f} = \begin{bmatrix} f_1(x,y) \\ f_2(x,y) \\ \vdots \\ f_N(x,y) \end{bmatrix}.$$

Using these matrices an estimate for position can be given in a similar way to the LLS approach but using the Jacobian instead of linear equations. A scheme known as Newton iteration has been found to be an optimal approach to solve this issue using NLLS [61]. The method consists of iteratively calculating new

position estimates and subtracting from the previous estimate to quickly approach an accurate result. This is depicted mathematically as

$$\mathbf{p}_{k+1} = \mathbf{p}_k - (\mathbf{J}_k^T \mathbf{J}_k)^{-1} \mathbf{J}_k \mathbf{f}_k \tag{3.19}$$

where $\mathbf{p}_k$ is the k_{th} iteration of $\mathbf{p}$ which contains the x and y coordinates of the target position as described in previous sections. The calculations are performed until the value converges to a small enough difference between the k and $k+1$ iterations designated by the process being performed, with the first estimate ($\mathbf{p}_0$) being given by the usual LLS method. A diagonal matrix can also be used to augment the $\mathbf{J}^T\mathbf{J}$ multiplication such that the algorithm searches for the path of steepest descent. The NLLS process is the most accurate method of performing position estimation [61]. However, while this technique can be computationally quick, it does require multiple iterations which is a disadvantage because the time required to come to a solution is increased in comparison with LLS. Furthermore, NLLS is quite a bit more complex than LLS and WLS and more difficult to implement. The decision on which regression method to use is based upon the results being achieved already and whether or not the implementation is worth the added accuracy.

3.3.2 Triangulation

Triangulation sets out to solve the same problem as trilateration but, rather than using distances from the BNs to the target, angles are used to estimate the target's position. There are many algorithms that apply this concept but here only one will be given as an example as to how this can be performed. This algorithm was presented by Pierlot and Van Droogenbroeck in 2011 [62]. As the name states, triangulation makes use of three angles and thus three anchors are required to get the desired results. The beacons are placed at known distances from one another in a defined coordinate system (in this case two dimensions) and they are given coordinates (x_i, y_i) for each i_{th} beacon. The target reference orientation (θ) is given as the heading direction of the BN (the direction it is facing), and angle measurements (α_i) from this reference are made to each anchor as shown in Fig. 3.8.

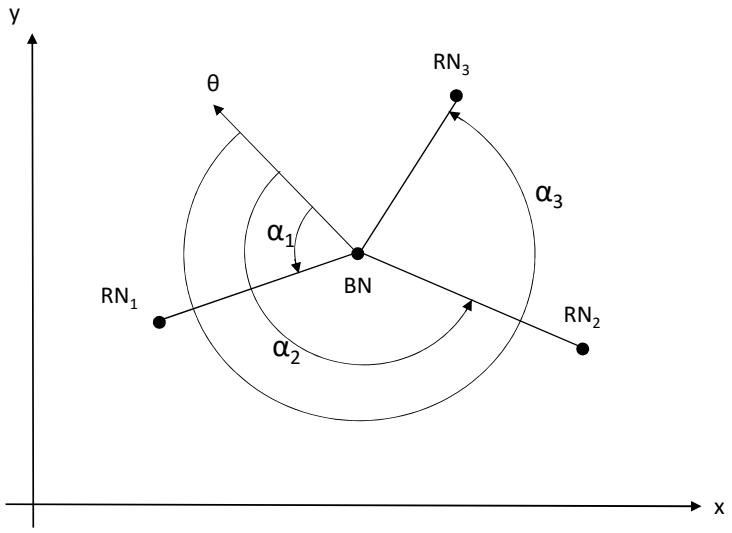

Fig. 3.8. Three beacons measuring angles from target reference point in 2-D coordinate system

With these values defined the algorithm can begin by first giving ”modified” beacon coordinates for BNs 1 and 3 as

$$x_1' = x_1 - x_2$$
$$y_1' = y_1 - y_2$$
$$x_3' = x_3 - x_2$$
$$y_3' = y_3 - y_2.$$

Next 3 ”R” values are found using the cotangent function and differences between angle measurements

$$R_{12} = \cot(\alpha_2 - \alpha_1) \tag{3.20}$$
$$R_{23} = \cot(\alpha_3 - \alpha_2) \tag{3.21}$$
$$R_{31} = \frac{1 - R_{12}R_{23}}{R_{12} + R_{23}} \tag{3.22}$$

which is followed by computing modified circle center coordinates $(x'_{i,j}, y'_{i,j})$ for all three circles as

$$x_{12}' = x_1' + R_{12}y_1'$$
$$y_{12}' = y_1' - R_{12}x_1'$$
$$x_{23}' = x_3' - R_{23}y_3'$$
$$y_{23}' = y_3' + R_{23}x_3'$$

$$x_{31}' = (x_3' + x_1') + R_{31}(y_3' - y_1')$$
$$y_{31}' = (y_3' + y_1') - R_{31}(x_3' - x_1').$$

The fourth step is to find k'_{31} which is the center of the circle passing through beacons 1 and 3 divided by two. This is found by

$$k_{31}' = x_1'x_3' + y_1'y_3' + R_{31}(x_1'y_3' - x_3'y_1'), \tag{3.23}$$

and then the position of the target, (x_T, y_T), is found

$$x_T = x_2 + \frac{k_{31}'(y_{12}' - y_{23}')}{D} \tag{3.24}$$
$$y_T = x_2 + \frac{k_{31}'(x_{23}' - x_{12}')}{D} \tag{3.25}$$
$$\tag{3.26}$$

where $D = (x_{12}' - x_{23}')(y_{23}' - y_{31}') - (y_{12}' - y_{23}')(x_{23}' - x_{31}')$. As stated previously in the section there are many ways of performing triangulation and only one novel algorithm applying the theory is presented here. The choice about which model to use depends upon the application which is also the case when deciding whether or not to use trilateration or triangulation for a certain purpose.

3.4 Adaptive Filtering

Filtering is necessary when recording measurements and attempting to locate a target because there will be a certain amount of ”noise” causing scattered data collection or certain erroneous values taken in or

calculated that lead to improper positioning. Furthermore, adaptive filtering can be used to predict behavior of the moving object. This section analyzes methods of adaptive filtering from simple averaging to a slightly more complex predictive algorithm known as the Kalman filter.

3.4.1 Moving Average

One of the least complex methods of adaptive filtering is known as the moving average filter. In this type of filter a set number of samples are read in and the mean of these measurements is used as the new value to be processed. Fig. 3.9 gives a graphical view of an array being updated with new samples.

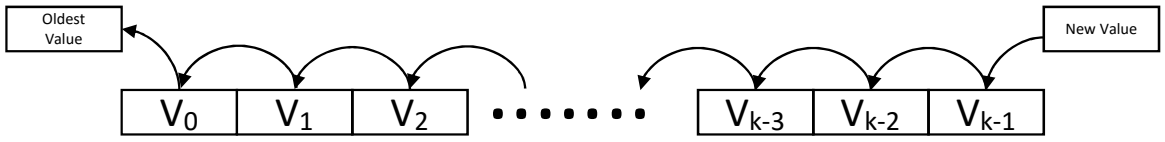

Fig. 3.9. Values updated in moving average array

The figure shows that the most outdated value (V_0) is replaced by the most recent reading and all other values are shifted one sample over. The arithmetic mean is then found across all k values. The length of the filter determines its responsiveness to noise or "smoothing" factor. A filter of this sort with a longer length will be smoother but will also update slower. A comparison of two different filter lengths are shown in Fig. 3.10.

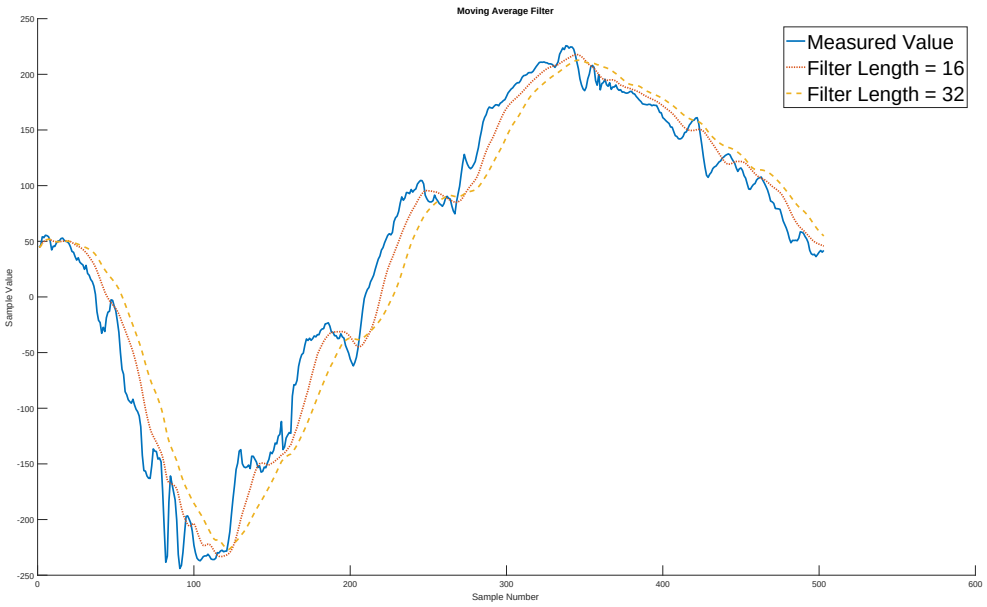

Fig. 3.10. Moving average filter length comparison

The figure demonstrates the moving average filter's ability to remove values that deviate heavily from the actual trend of the data. Issues with this method include wait time when initially populating the averaging array and old values which are no longer relevant effecting the calculated average. A way to mitigate the former problem is to simply take the average of the samples that have been taken prior to filling the entire array, and for the latter issue a method of weighted averaging can be introduced in which older samples are given less importance than newer values by way of some diminishing factor.

3.4.2 Exponential Smoothing

Exponential smoothing is similar to the moving average in the sense that previous samples are used to cancel out outliers or moments of incorrect data. However, in this case all previous samples contribute to the calculated value. Older samples are given less weight by a factor of α. The method is expressed mathematically as

$$s_k = \begin{cases} x_k, & k = 0 \\ \alpha \cdot x_k + (1 - \alpha) \cdot s_{k-1}, & k > 0 \end{cases}$$

where $0 < \alpha \leq 1$, x_k denotes the sample read in, and s_k is the k_{th} smoothed value. A comparison for different values of α can be seen in Fig. 3.11 for the same samples as in Fig. 3.10.

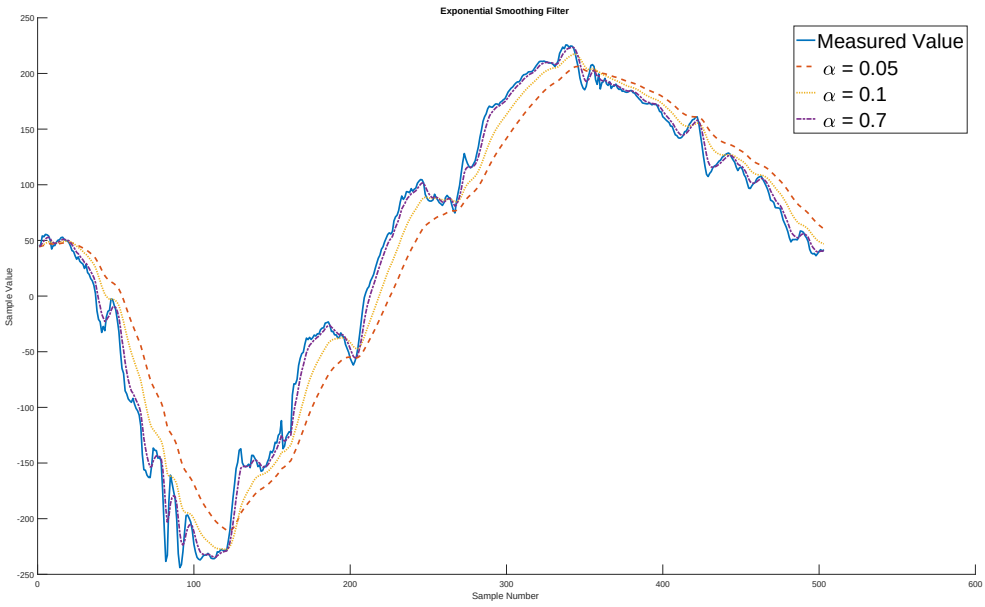

Fig. 3.11. Exponential smoothing for different values of α

The figure shows that the performance of the exponential smoothing filter gives less noisy results for lower values of α but, as was the case with the moving average, the filter allowing for less noisy values is more susceptible to be too slow to follow a moving target at an acceptable rate. The goal in this situation is to find a balance between noise reduction and speed which can be done with simulations such as the one shown in Fig. 3.11. While the exponential smoothing method shows very similar results to the moving average approach it is easier to implement since previous values don't need to be stored; they are instead factored in as a consequence of the formula.

3.4.3 Kalman Filter

The Kalman filter is a set of mathematical equations that provides an efficient, recursive, solution of the least-squares method. The filter operates by predicting the way a process will behave based on feedback control [63]. The filter operates as an iterative two-step process in which a prediction of the process state is made, and the measurement of that process is *corrected* by that prediction. This operation is shown in Fig. 3.12

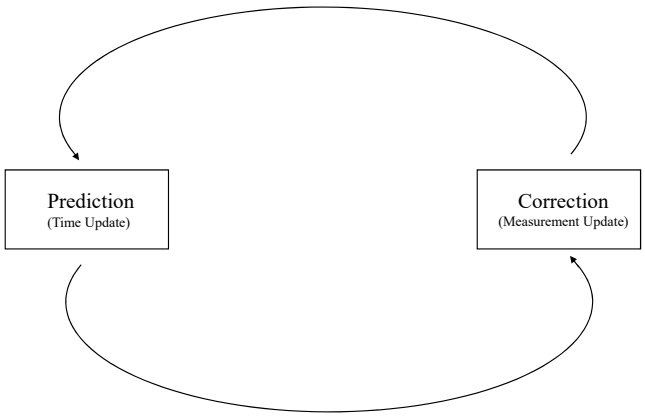

Fig. 3.12. Kalman Filter cycle

The Kalman achieves its goal by the use of several matrices which are defined based on the process which the Kalman filter is being used for. The matrices are

- **A** : system matrix
- **C** : measurement matrix
- **R** : expected measurement noise
- **P** : covariance matrix
- **Q** : process noise

The way these matrices are formatted and defined will be discussed in the implementation section about the Kalman filter as the parameters vary based on performance and application use. This section will simply demonstrate the matrix algebra necessary to enact the Kalman filter.

The Kalman gain matrix is defined to be

$$\mathbf{K}_k = \mathbf{P}_k\mathbf{C}^T(\mathbf{C}\mathbf{P}_k\mathbf{C}^T + \mathbf{R})^{-1}. \tag{3.27}$$

The state-variable, x_k, and the covariance matrix, $\mathbf{P}_k$ can then be tracked recursively over time at $k = 1, 2, ...$ samples. The first step of the algorithm starts with the measurement update (or correction)

$$x_k^+ = x_k + \mathbf{K}_k(y(k) - \mathbf{C}x_k) \tag{3.28}$$

where $y(k)$ is the data input at sample k. This is followed by the uncertainty correction in the covariance matrix

$$\mathbf{P}_k^+ = \mathbf{P}_k - \mathbf{K}_k\mathbf{C}\mathbf{P}_k. \tag{3.29}$$

The "+" superscript in (3.28) and (3.29) is intended to show that these are updates to the previous values but remain in the same k_{th} iteration as they are used for the state and uncertainty predictions of the next

repetition of the cycle. The predictions of the next state and uncertainty are then performed:

$$x_{k+1} = \mathbf{A}x_k^+ \tag{3.30}$$

$$\mathbf{P}_{k+1} = \mathbf{A}\mathbf{P}_k^+\mathbf{A}^T + \mathbf{Q}. \tag{3.31}$$

These calculations are executed upon each reception of new data and allow for predictive analysis of a moving target by incorporating its current position and velocity. The next ”state” of the system (x_{k+1}) gives the predicted values of the x and y coordinates.

The Kalman filter allows for smoothing in a similar way to methods mentioned previously but also makes predictions based on earlier recorded data. This makes the filter ideal for use in tracking applications. The Kalman can be improved upon and other methods can be used that perform similar functionality such as the particle filter or elliptical gating, but they will not be discussed in this thesis work.

The theory chapter has given background information on the different approaches considered for each block of the design. In the following chapter, the decisions made for implementation of each block are presented, and the reasons for why these choices were made are provided.

4

Implementation

The implementation of the system was divided into three major parts in order to simplify the way the tasks were split between team members as well as how each portion was scheduled. These three main areas emerge from the general functional diagram presented in Fig. 4.1 and are: ranging, positioning, and movement. Since the beginning of the project some parameters, such as the inputs and outputs of the functional blocks, were defined to allow work to be done in parallel without being stalled due to waiting for the completion of any of the other segments. To this end each of the anchors were given a name in order to differentiate them through all functions. The names given are: target (T), front left (FL), front right (FR), rear right (RR), and rear left (RL) with the last four being the ones placed on the tracking device. This chapter presents the implementation of each of the parts of the system and describes, in detail, the software and hardware development process and final product.

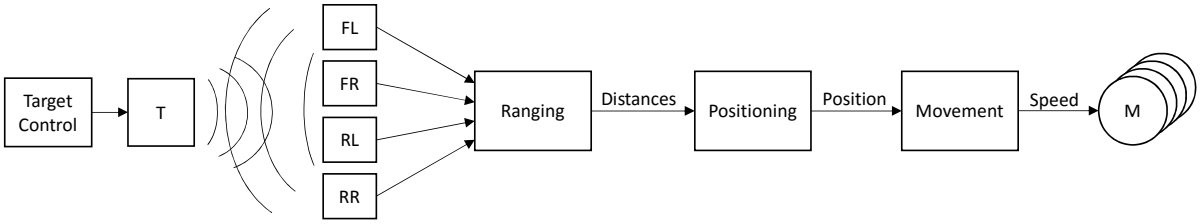

Fig. 4.1. Functional block diagram of the system

The general hardware diagram of the tracking device is presented in Fig. 4.2. The central unit is in charge of communicating with the UWB modules, controlling the ranging protocol, and running the positioning and adaptive filtering algorithms. The movement algorithm is run in a separate unit called motor controller in order to avoid interfering with the ranging protocol. The motor driver receives the outputs from the motor controller and supplies the motors with the necessary current for achieving the desired speed. The target hardware consists of a microcontroller unit and an UWB module.

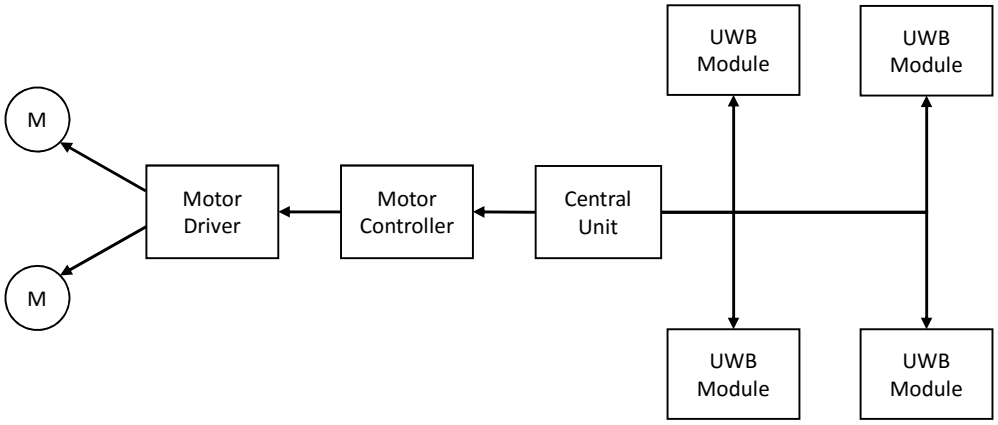

Fig. 4.2. Hardware block diagram of tracking device

4.1 Ranging

The ranging function finds the distance between the target and each of the anchors. This function was divided into two different parts: hardware and software. The hardware part deals with the connection between the UWB modules, the microcontrollers, and other components. The software part deals with the program running on the microcontrollers and device configurations.

4.1.1 Hardware

The hardware components needed for the ranging functionality are: the central unit, UWB Modules, and target control unit. As shown in Fig. 4.2 the central unit is connected to four UWB modules, and the target control unit is only connected to one UWB Module. The components were selected based on the general requirements, hardware specifications, cost, and availability. Below the selected components are presented together with a justification of the selection and a brief description of each component.

4.1.1.1 Central Unit

- **Arduino M0 Pro [64]:** This Arduino board (Fig. 4.3) is based on the Atmel's SAMD21, a 32-bit microcontroller from the ARM Cortex M0+ family. The Arduino platform was selected as per Cybercom's request. As Arduino is widely known by embedded and non-embedded developers this choice makes it easier for Cybercom engineers to continue the development of the tracking device. In addition, there is plenty of documentation, examples, and libraries that make the development phase easier and faster. The *arduino-dw1000* library, available online [65], was developed to setup and control the selected UWB module using the Arduino platform. The Arduino M0 Pro board includes other features that make it stand out from other Arduino boards. These features are: 48MHz clock speed, 3.3 V operating voltage, 256KB flash memory, and 20 digital I/O pins.

Fig. 4.3. Arduino M0 Pro [64]

4.1.1.2 UWB Modules

- **Decawave DWM1000 [66]:** Few manufacturers working with UWB technology develop and sell transceivers that can be used as peripherals in an embedded system. One of these manufacturers is Decawave which develops the DWM1000 module (as seen in Fig. 4.4) which is based on their DW1000 UWB transceiver. The module includes an integrated antenna, RF circuitry, power management, and clock circuitry. It has been designed specifically for real-time location applications based on the different time based ranging methods. The module, running at 3.3 V, is controlled by a host processor through a serial peripheral interface (SPI). Other features include: low power, low cost, wide range of configuration, and IEEE 802.15.4-2011 compliance. One disadvantage of the DWM1000 module is that the clock synchronization functionality from the DW1000 is not available, therefore ranging methods requiring synchronization between devices can not be implemented.

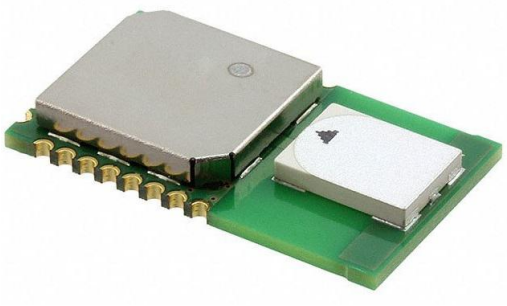

Fig. 4.4. Decawave DWM1000 [66]

4.1.1.3 Target Control Unit

- **Arduino Pro Mini 3.3V [67]:** The target control unit requires a small, low power microcontroller capable of controlling the ranging functionality of the target UWB module via SPI. The Arduino Pro Mini is a 1.7x3.3cm microcontroller board (Fig. 4.5) base on the 8-bit ATmega328. It runs at 8 MHz and an operational voltage of 3.3 V which makes it ideal for communicating with the DWM1000.

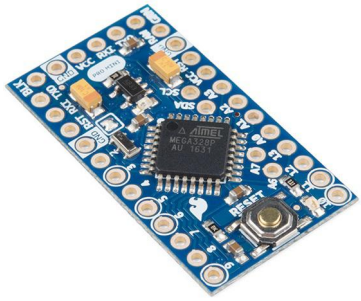

Fig. 4.5. Arduino Pro Mini 3.3V [67]

4.1.1.4 Components connection

The four anchors on the tracking device are connected to the Arduino M0 Pro using the SPI pins. Additionally, reset, interrupt, and chip-select need to be connected to independent pins on the Arduino. Fig. 4.6 shows a diagram of how one DWM1000 module is connected to the Arduino. For more details about the hardware implementation please see circuit schematics in Appendix A.

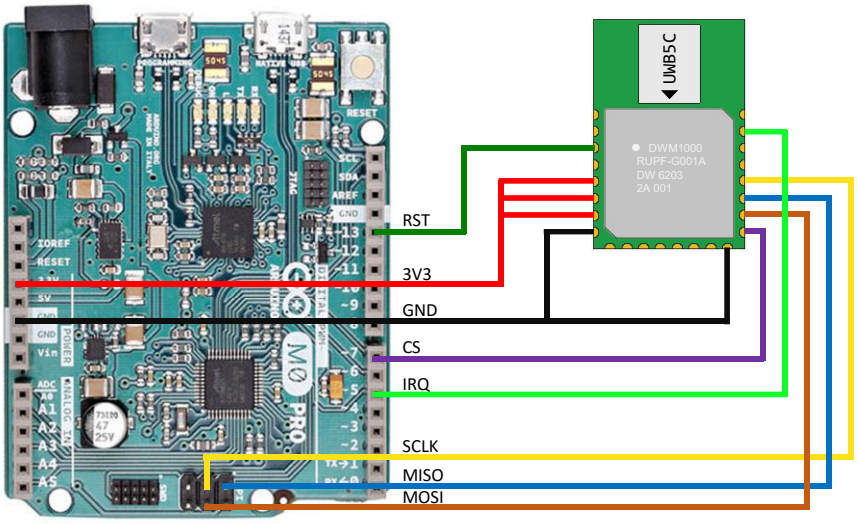

Fig. 4.6. DWM1000 UWB module connection to Arduino M0 Pro

On the target the DWM1000 is connected the same way as in Fig. 4.6 with the only difference that the microcontroller board used is an Arduino Pro Mini. The Arduino Mini Pro SPI pins are specified in the datasheet and the rest of the required signals can be connected to any of remaining 11 I/O pins.

4.1.1.5 PCB Design

Four PCBs were designed for this project. The first PCB was designed early in the project to allow for work with the DWM1000 module on a breadboard and to handle the module easily. This breakout board give us access to the SMD pins of the DWM1000 by using headers, as shown in Fig. 4.7. The board size is 25.91x25.65mm and fits a regular breadboard without issues.

Fig. 4.7. Breakout board for the DWM1000

The final PCBs were designed after the system was tested using breadboards to connect all the components together. The PCB shown in Fig. 4.8 is the central unit PCB. It was designed as a shield to the Arduino M0 Pro with one IDC connector for each of the anchors. A 3.3 V linear voltage regulator is also included in the board for powering the anchors as the regulator on the Arduino was overloaded during the tests. The shield can be powered using a micro USB cable or a separate connector on the BAT header with a 5 V supply. It also includes headers connection for the motor controller unit and a Bluetooth module.

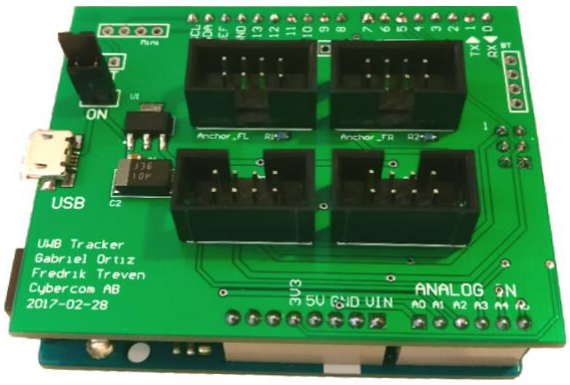

Fig. 4.8. Arduino shield for central unit and anchor connections

Another PCB was designed for the target control unit and UWB module. In order to save space and make the target as small as possible, the Arduino Mini Pro is placed under the UWB module. In addition, a 3.3 V linear voltage regulator is placed under the Arduino Mini pro. The regulator is used for powering the UWB module as the regulator in the Arduino Mini Pro is only rated for 50 mA while the module can draw up to 180 mA. Fig. 4.9 shows the assembled target PCB with the Arduino Mini and the UWB module. To power the target device it is possible to use a micro USB cable or a separate connector on the BAT header with a 5 V supply. Either of these connections power the Arduino Mini as well.

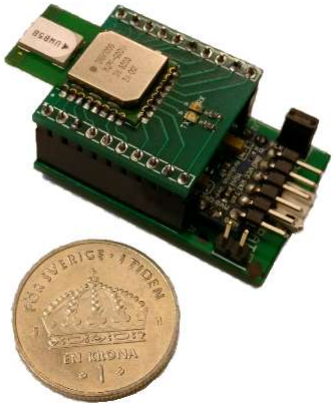

Fig. 4.9. Target device PCB (coin for scale)

The last PCB was designed specifically for the anchors as an extension for the breakout board designed at the beginning of the project. This PCB provides a better ground for the DWM1000 module and has an 8-pin 2mm header connector for connecting the anchors to the central unit. As shown in Fig. 4.10 the anchors PCB has two 10-pin female headers were the breakout board with the DWM100 module is mounted.

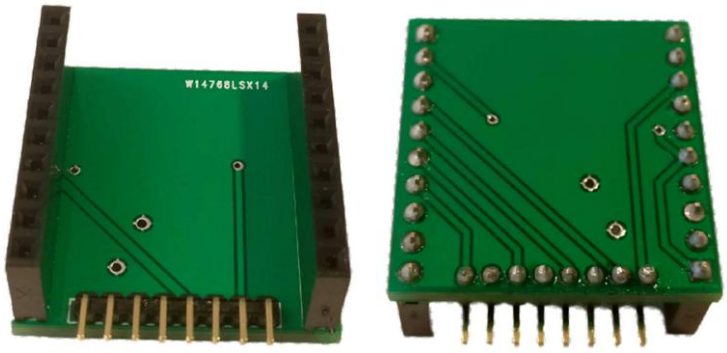

Fig. 4.10. Anchor PCB

4.1.2 Software

An asymmetric double-sided two-way ranging scheme was chosen for obtaining the distance between each of the anchors and the target. It is worth noting that a TDOA method could have been more appropriate for this kind of application and setup but unfortunately the DWM1000 does not allow clock synchronization between devices therefore it is not possible to implement this method. This decision is further discussed in Section 6. The ranging communication is initiated in the tracking device (central unit) in order to have better control over the program flow through the ranging of each of the anchors, the positioning algorithms, and the movement data transfer to the motor controller. Below we describe how the ranging is achieved, starting from a one-to-one ranging function between two UWB modules to the whole system with the four anchors.

4.1.2.1 One-to-one Ranging

The DW1000 module allows a wide range of configurations that need to be carefully setup before the ranging starts. Some of the main parameters set are shown in Table 4.1. Center frequency and bandwidth

are set by selecting the transmission channel of the module. The center frequency possibilities range between 3.5 GHz and 6.5 GHz. It is also possible to select either 500 MHz or 1330 MHz as transmitting bandwidth. A combination between lower frequency and higher bandwidth achieves a longer range but also leads to an increase in power consumption. The data rate can be set to 110 kbps, 850 kbps, or 6.8 Mbps; low data rates are used for long range applications while higher data rates are used in short range applications. The preamble is a sequence of pulses used for preparing the receiver module for an incoming ranging message. The preamble size defines how many times the sequence is repeated; a long preamble gives improved range whereas a short preamble reduces air time allowing faster ranging and lower power consumption. The pulse repetition frequency (PRF) is the frequency as which the preamble is repeated; higher PRF values can improve the accuracy on the first path time stamp at a higher power consumption cost.

Table 4.1 – DWM1000 Configuration parameters

Centre Frequency	Bandwidth	Data Rate	Preamble Length	PRF	Data Length
6489.6 MHz	499.2 MHz	6.8 Mbps	128	16 MHz	16 Bytes

The chosen configuration is based on the application requirements and empirical experience. As this is a short range application where the target is expected to be within ten meters of the tracking device, a higher data rate was selected together with higher center frequency and short preamble. The PRF was set to 16 MHz as the 64 MHz didn't show any improvement in accuracy, moreover, it reduced the reliability of the ranging and increased power consumption. This set of parameters showed a good stability in the communication between modules with only a few lost messages. It also achieved a higher ranging frequency compared to other configurations, that is, more completed ranges per second.

After setting up the module, the ranging protocol is initiated in the anchor side as stated before. Fig. 4.11 and Fig. 4.12 show the software flow chart for the ranging function in the anchor and the target side respectively. The anchor begins by sending a POLL message and going into receive mode waiting for a POLL ACKNOWLEDGE message. A timestamp is stored with the time at which the POLL message left the module (timePollSent). The target device is started in receiving mode waiting for a POLL message. As soon as the target receives the POLL message from the anchor, a timestamp with the time the message was received is stored (timePollReceived). After a delay time previously set to 1000 us, the target replies with a POLL ACKNOWLEDGE message , saves the time at which the message left the module (timePollAckSent), and goes into receiving mode waiting for a RANGE message. The anchor receives the POLL ACKNOWLEDGE message and generates a timestamp (timePollAckReceived); after the defined delay time, the anchor sends a RANGE message and creates a timestamp for it (timeRangeSent). The target receives the RANGE message and generates the timestamp (timeRangeReceived), then, it replies with the last message in the protocol. This last messages includes all the timestamps that were created on the target side. When the anchor receives the last message, it extracts the timestamps in order to calculate the distance between the two modules using the following equations:

$$t_{round1} = timePollAckReceived - timePollSent$$

$$t_{reply1} = timePollAckSent - timePollReceived$$

$$t_{round2} = timeRangeReceived - timePollAckSent$$

$$t_{reply2} = timeRangeSent - timePollAckReceived$$

$$TOF = \frac{(t_{round1} \cdot t_{round2}) - (t_{reply1} \cdot t_{reply2})}{t_{round1} + t_{reply1} + t_{round2} + t_{reply2}}$$

$$distance = TOF \cdot \mathrm{c}$$

The DWM1000 module is able to generate interrupts for several events or states, in this case only the message sent and message received events interrupts are used. When a message sent interrupt occurs, the microcontroller creates the timestamp for the message leaving the module; when a message received interrupt occurs, the microcontroller identifies the type of message, extracts the timestamps if necessary, and decides on the following actions depending on the received message.

A watchdog timer is also implemented in both the anchor and the target control units to avoid getting stuck at different states of the protocol, for example, when one device is sending a specific message and the other device is waiting for a different one. This happens regularly as some messages might get lost and never reach its destination. The watchdog resets both devices to initial conditions, that is, the anchor will send a POLL message and the target will be in receiving mode waiting for a POLL message.

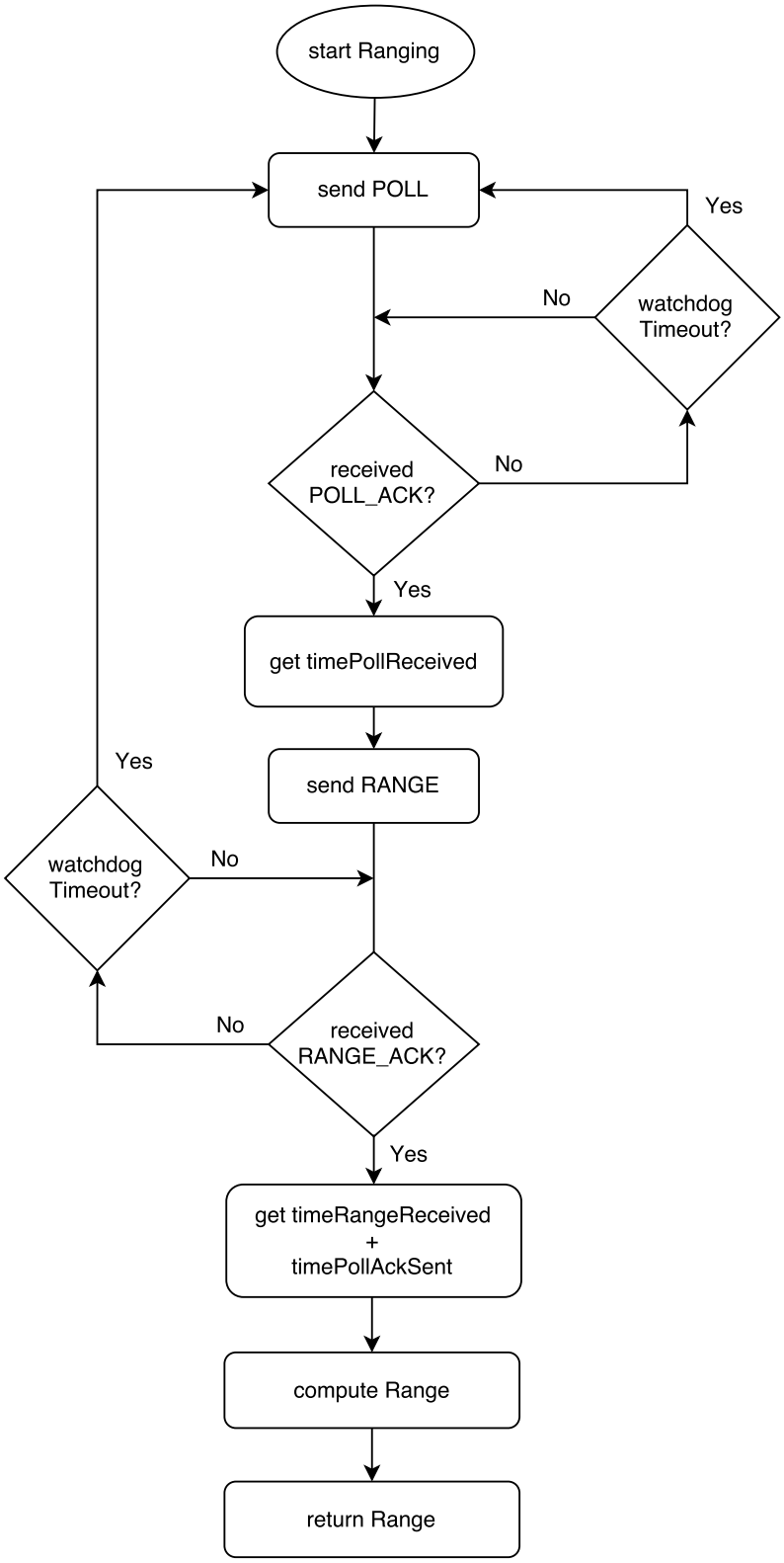

Fig. 4.11. Software flow chart for ranging function on the anchor side

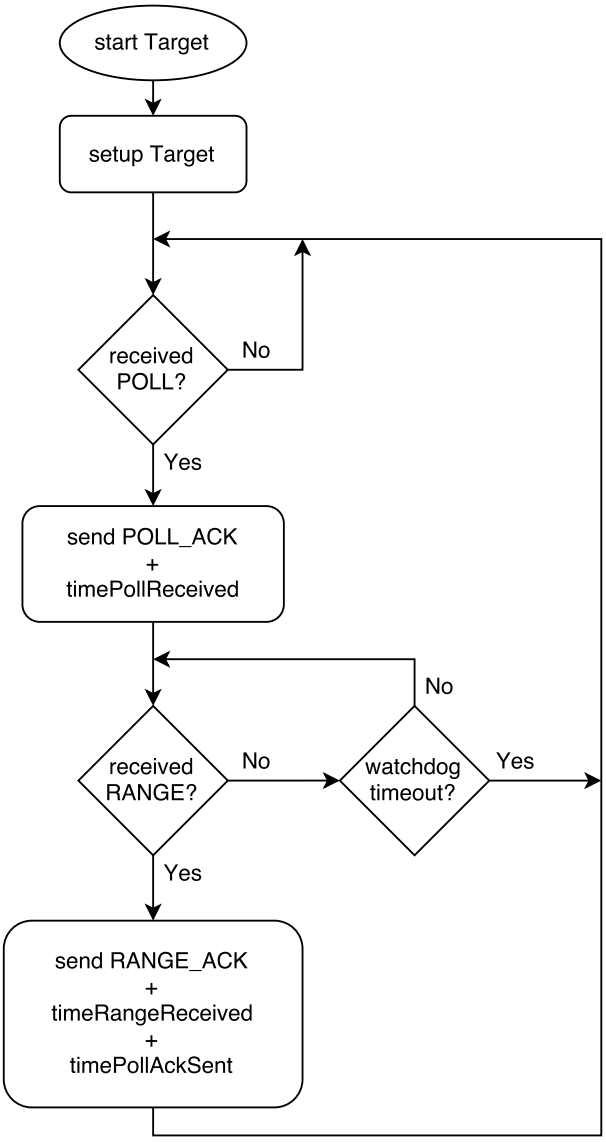

Fig. 4.12. Software flow chart for ranging function on the target side

4.1.2.2 Four Anchors

The ranging with four anchors is done in the same way as the one-to-one ranging. The function presented in Fig. 4.11 is called sequentially for each of the anchors from a main program running in the central unit. Each anchor has its own library files where the interrupts, reset pin and chip select pins are linked to. These libraries are used for communicating independently with the desired anchor. When an anchor finishes its ranging task it is set to idle mode until the next ranging has to be done. On the target side, the function presented in Fig. 4.12 is placed inside an infinite loop so it starts over again when the ranging is completed. The target runs the same ranging function regardless of which anchor initiated the ranging scheme. Fig. 4.13 shows a time and messages diagram of how the ranging is done for the first two anchors, this sequence is repeated for the other two anchors and then restarted.

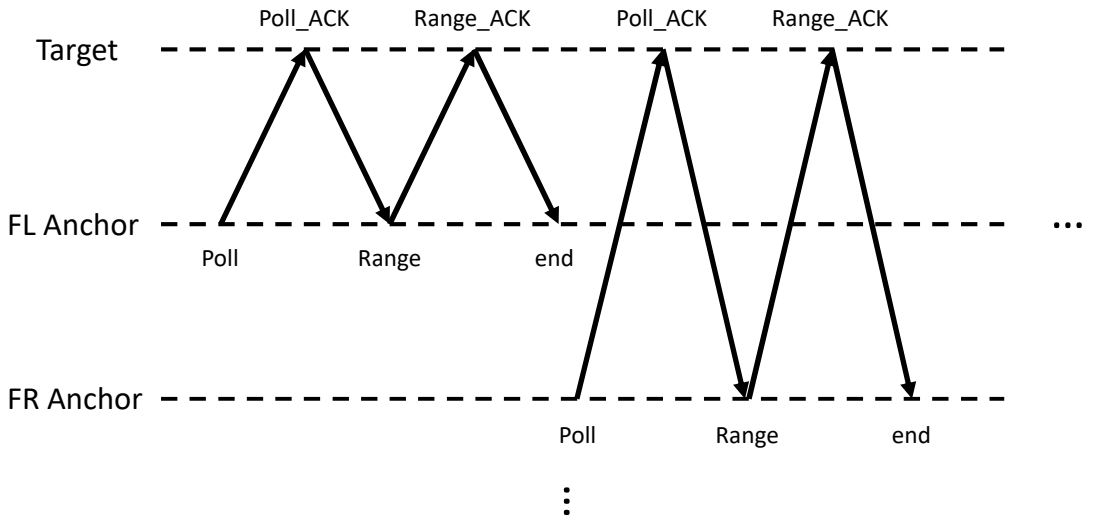

Fig. 4.13. Ranging messages exchange

The resulting distance calculated for each of the anchors is used as an input to the positioning function which calculates the actual position of the target in relation the tracking device. The positioning function is called every time an anchor finishes ranging, that is, every time a new distance value is available. The positioning function implementation is described in the next chapter.

4.1.2.3 Offset Correction

After doing some tests using the ranging functionality, it was found that there is an offset between the distance measured by the UWB modules and the actual distance. Fig. 4.14 shows a graph with the measured distance compared to the ideal scenario. The ideal scenario occurs when the measured distance is the same as the actual distance. As the measured distance shows a linear behavior, it is easy to implement an offset correction where the measured values are projected into the ideal scenario. Fig. 4.15 shows the measured distance using the offset correction.

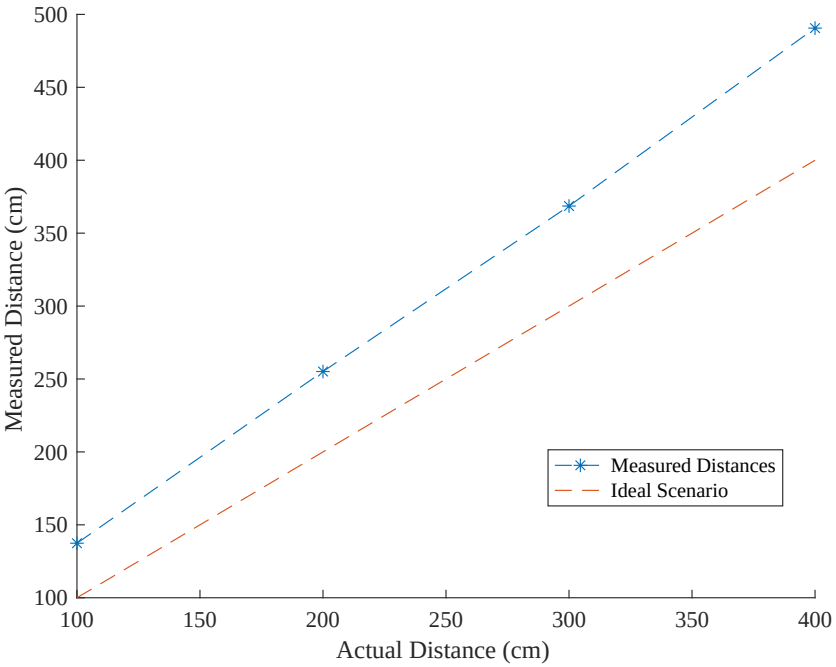

Fig. 4.14. Measurements without offset correction

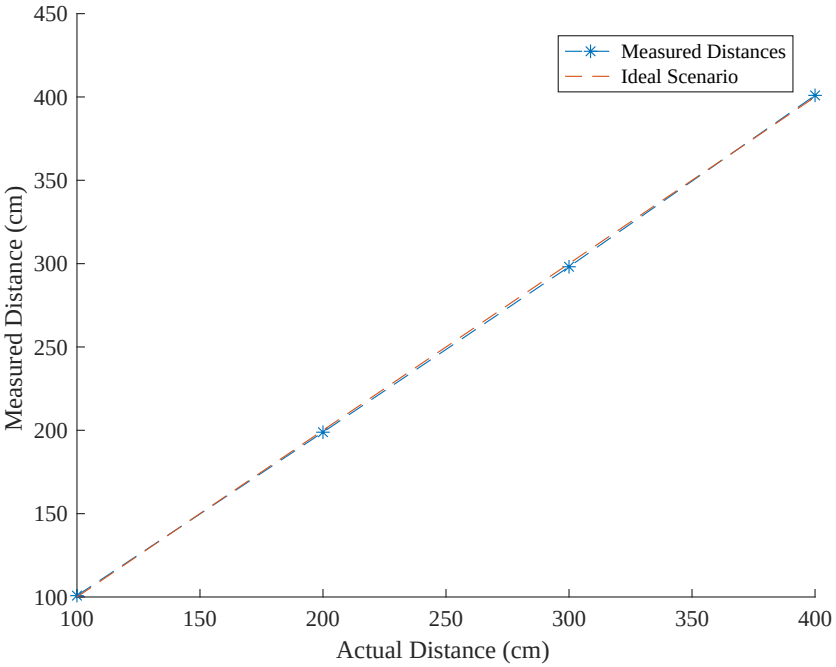

Fig. 4.15. Measurements with offset correction

4.2 Positioning

When distances are recorded from ranging with each anchor on the tracking device it is necessary to find a method which utilizes this data in a way that can locate the target. Positioning algorithms provide mathematical ways to manipulate the data received from sensors in order to perform this task.

4.2.1 Multilateration Linear Least Squares

Due to the inherently high accuracy of UWB when measuring distance, the decision was made to use a form of lateration as a positioning technique since the distances gauged are used directly when finding position. Multilateration with 4 anchors was the chosen implementation method, initially due to the symmetry with which these anchors can be placed with respect to one another facilitating the definition of the coordinate plane. Furthermore, this method provides capability for better estimates when calculating the target position than trilateration due to the increase in measurements. The coordinate system is defined such that the origin is at the center of the robot as shown in Fig. 4.16.

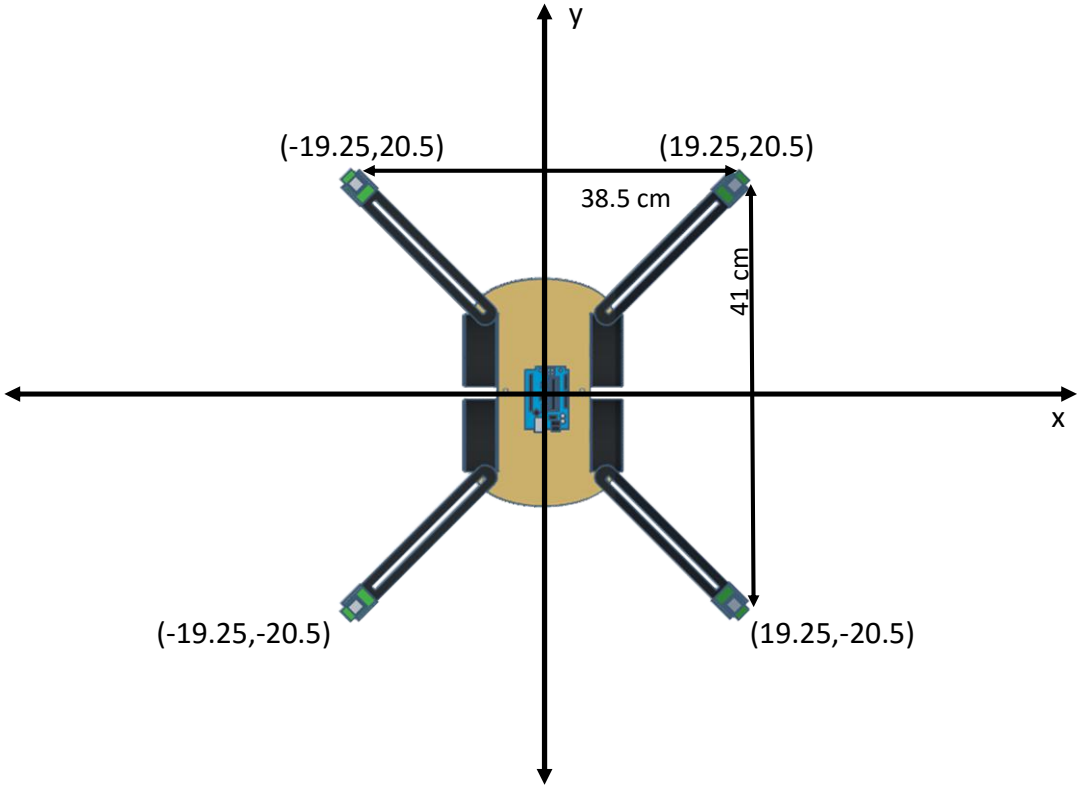

Fig. 4.16. Definition of relative coordinate system with respect to the tracking device

The coordinate system is defined in centimeters and the coordinates of each anchor shown in Fig. 4.16 are used for the multilateration algorithm to function appropriately. A goal of this thesis is to see how close the anchors can be to one another and still perform accurate positioning. The setup seen in Fig. 4.16 is the initial configuration of the robot in which the functionality was more than sufficient for tracking and following purposes. This configuration was changed to see how the behavior was affected at shorter gaps between the anchors. These tests were performed to see how small a device like this could be when implemented for real world applications, and the Results/Discussion chapters give more information about this.

The multilateration process is completed by the linear least squares algorithm which was implemented to reach a reasonable estimate of the location being found. The reason for not extending this to the weighted least squares method was that each sensor acts in a similar way and the confidence derived from each node is the same. The argument could be made that WLS could have been implemented to give greater weight to the anchors which measured a closer distance to the target since the error in such a node would be the lowest. However, since the anchors can be placed relatively close to one another which leads to negligible differences in confidence between them, the complexity required for implementing this type of ”revolving” weighted least squares method was ruled out. The non-linear least squares method was also dismissed due the added complexity of implementing such a solution, and that the results being generated were already sufficient for the goals of this project. The LLS method works well in this application since the robot is following a quite slow moving target and the linear estimation is sufficient to generate an appropriate prediction. If the tracking device was to follow a faster moving object such as a vehicle, then NLLS may need to be used as the behavior of such an object would deviate from linear standards in a more significant way than a slow moving target.

Using multilateration with LLS was first tested by simulating the performance of the algorithm with known coordinates to see if the values could be predicted properly when using only the distance that would be recorded from this point to each of the anchors. The algorithm functioned perfectly when testing in this way because the distances were derived from an already known point in space. When applied in practice, however, it became clear that ranging errors in the anchors would lead to distances that could not originate from one single point. Fig. 4.17 shows how the algorithm functioned with a given coordinate with exact distances calculated with $d_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$.

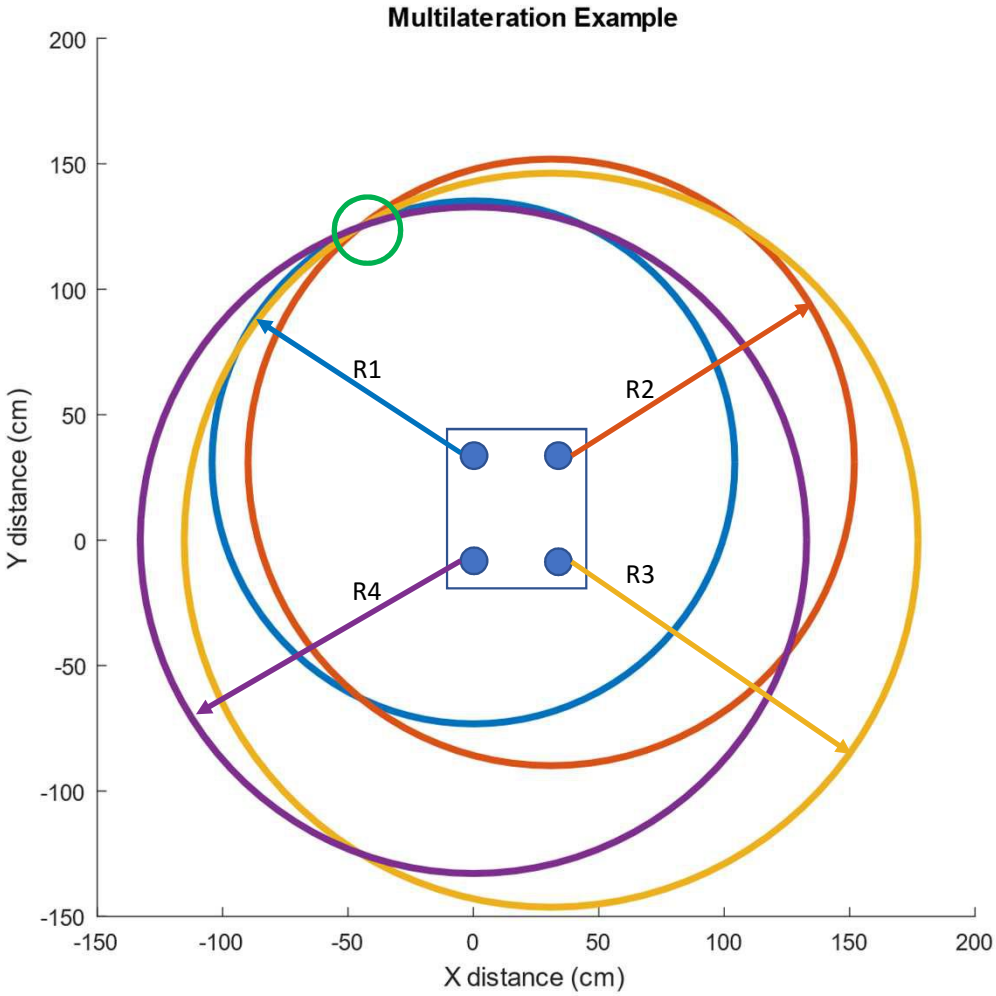

Fig. 4.17. Multilateration With Exact Distances

It is clear in Fig. 4.17 where the location of the target should be estimated to be, and, in this circumstance with exact distances, the predictive algorithm finds the position accurately. When using the distances received through ranging the area of possible location becomes much larger due to errors (however small) in the recorded values. Fig. 4.18 gives a visualisation of one such calculation.

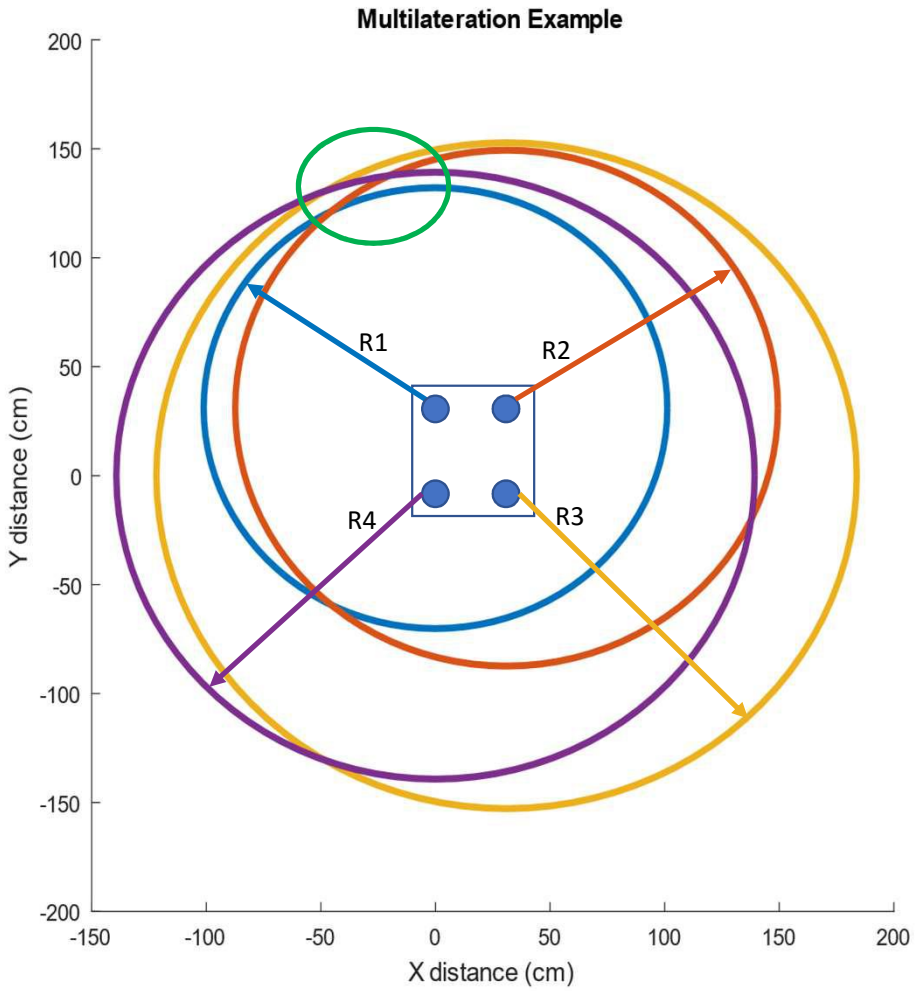

Fig. 4.18. Multilateration With Measured Distances

The position in this case is not as clear and the accuracy of the prediction from the algorithm suffers as a result.

When performing tests with the measurements read in by the anchors it was discovered that the prediction being made was consistently at a point beyond the radius of any of the circles formed. As it was impossible that the target was actually in the location estimated a decision was made to augment the prediction slightly. At first, this was done by attempting to simply find the intersection of the two anchors reading the shortest distances to the target. This method allowed for more accurate readings when the object being tracked was stationary, however, when moving and necessitating a switch between two anchors being used for positioning, large instability was introduced, and the system could not re-calibrate itself properly for practical use. The two-circle intersection method was compared extensively with the first multilateration method and then was discarded as it did not offer the precision necessary. The comparison was done through observation of the movement of the robot and with a GUI used to visualize the position of the target with respect to the tracking device. This GUI allowed for the viewing of multiple solutions at once and seeing which approach performed better. This was also useful for different filter parameters; more on this can be seen in the Results section. To mitigate the position "overshoot" caused by the LLS algorithm it was decided that, after the predicted value was given, the point should be projected down onto the circle with the shortest measured radius. This position would then be used to track and follow the target as it gave, if not more accurate, a more reasonable estimate

of the target's position. This projection was deemed acceptable as the predictive algorithm was giving the location at an accurate angle from the origin of the robot but simply missing the distance at which the target was from it. Mathematically this projection is done by updating the coordinates as follows:

$$x'_T = \text{sgn}(y_T) \cdot x_T \frac{r}{d} \tag{4.1}$$

$$y'_T = |y_T| \frac{r}{d} \tag{4.2}$$

where r is the radius of the circle that the point is projected down to and d is the distance from the origin measured by the originally calculated coordinates, x_T and y_T, to the initial point ($d = \sqrt{x_T^2 + y_T^2}$).

This method was implemented to perform a new position calculation with each incoming distance value recorded. This means that the algorithm does not wait for all four anchors to record new distances prior to estimating the target position, but, rather, a new distance value being read in by any anchor triggered the multilateration positioning. This was done to increase the number of predictions being made per second in order to update the position of the target at a higher rate. A drawback to this method is that erroneous values or calculations could lead to quick jumps that may have been avoided if the position was found less often. This issue can be solved with higher data rates (meaning that the accurate predictions will greatly outweigh those that are incorrect) and adaptive filtering which will be discussed in the following subsections.

4.2.2 Moving Average and Exponential Smoothing

The moving average and exponential smoothing filters are both special cases of discrete-time filters. The former was an early consideration for filtering the data to remove samples that deviated greatly from previous values. The results when testing this approach for only one-to-one ranging were positive as the spikes from inconsistent data were removed, and the data was filtered in a way in which the trend of the actual measurements were still followed closely. Fig. 3.10 shows the performance of this filter. The moving average filter was implemented by setting up a buffer with a length equal to the number of distances the average was to be taken over for each anchor. When applied and conjoined with the movement algorithms it was discovered that the robot was over-adjusting and oscillating while stagnant in positions due to the fact that the filter was putting too much significance on older values. At first, the consideration was made to add weighting in order to place higher significance on more recent data. However, when testing with shorter and shorter filter lengths the performance of the robot improved until the filter was removed in its entirety which seemed to fix the issue altogether (it is important to note that at this point the Kalman filter had also been implemented thus facilitating the discovery that the averaging filter could be removed).

Upon the elimination of the moving average method the robot was functioning in a stable way, but in order to see if performance could be improved further the exponential smoothing filter was considered. This was simple to implement as the algorithm implicitly keeps track of previous values in the result of the equation $s_{k+1} = \alpha x_k + (1 - \alpha) s_k$. This removes the necessity of keeping a buffer of previously read in distances. This approach uses the parameter α to determine the rate of smoothing where $\alpha \approx 0$ is the maximum amount of smoothing (and thus the slowest response time) and $\alpha = 1$ is no smoothing at all. While the implementation of this method was simpler the results suffered from the same problems as those from the moving average case. The older values simply contributed too much to the decision and by the time the robot had moved the new prediction was not updated quickly enough again leading to oscillation. Therefore, the decision was made not to use any discrete-time filter on the distance readings and rather only filter the position estimate with the Kalman Filter.

The reason these are included in the implementation section is because they were a large part of discovering the behavior of adaptive filters and the moving average filter was the initial method that was used to improve upon ranging and positioning performance without altering the actual process with

which the measurements were being carried out. Furthermore, there is still a function contributed to the ranging filter which removes any negative distances or values caused by memory overflow. The implementation of the Kalman Filter was not only demonstrated to be an improvement over both of these methods, but also led to the discovery that, when used, it could circumvent the need for other types of adaptive filtering altogether.

4.2.3 Kalman Filter

In the Theory chapter the Kalman Filter is discussed in terms of how the various matrices required are manipulated and how the algorithm uses them in order to perform the cycle of state predicting and updating. This section focuses on how the values and structures of each of these matrices are chosen with respect to the specific application for which they are being used. In this case the application is dynamic relative real-time tracking of an object. Recall that the matrices in question are:

- **A** : system matrix [4x4]
- **C** : measurement matrix [4x2]
- **R** : expected measurement noise [2x2]
- **P** : covariance matrix [4x4]
- **Q** : process noise [4x4]

To define the system matrix we consider the finite difference approximation for computing the derivative over time (speed) of the position x at time t characterized in discrete time by $t = k\Delta t$.

$$\dot{x}(t)|_{t=k\Delta t} \approx \frac{x(k\Delta t + \Delta t) - x(k\Delta t)}{\Delta t} \tag{4.3}$$

is the finite difference equation in question where Δt is the sampling time required to derive a discrete-time state-space model of the form

$$s(k+1) = \mathbf{A}s(k) + w(k). \tag{4.4}$$

In (4.4) $s(k)$ denotes the discrete time form corresponding to $k\Delta t = t$. In order to relate this to two-dimensional positioning with x and y as coordinates, the vector $\vec{s}(t)$ is defined to be

$$\vec{s}(t) = \begin{bmatrix} s_1(t) \\ s_2(t) \\ s_3(t) \\ s_4(t) \end{bmatrix} = \begin{bmatrix} x(t) \\ \dot{x}(t) \\ y(t) \\ \dot{y}(t) \end{bmatrix}$$

which shows that

$$\begin{aligned} \dot{s}_1(t) &= s_2(t) \\ \dot{s}_2(t) &= w_x(t) \\ \dot{s}_3(t) &= s_4(t) \\ \dot{s}_4(t) &= w_y(t). \end{aligned}$$

From here (4.3) is applied to get equations for the position and velocity in both the x and the y direction:

$$\dot{s}_1(k) \approx \frac{s_1(k+1) - s_1(k)}{\Delta t} = s_2(k)$$
$$\dot{s}_2(k) \approx \frac{s_2(k+1) - s_2(k)}{\Delta t} = w_x(k)$$
$$\dot{s}_3(k) \approx \frac{s_3(k+1) - s_3(k)}{\Delta t} = s_4(k)$$
$$\dot{s}_4(k) \approx \frac{s_4(k+1) - s_4(k)}{\Delta t} = w_y(k)$$

in which w_x and w_y are additive noise in the x and y directions respectively. The system of equations can now be solved to get

$$\vec{s}(k+1) = \begin{bmatrix} s_1(k) + \Delta t \cdot s_2(k) \\ s_2(k) + \Delta t \cdot w_x(k) \\ s_3(k) + \Delta t \cdot s_4(k) \\ s_4(k) + \Delta t \cdot w_y(k) \end{bmatrix} = \mathbf{A}\vec{s}(k) + \vec{w}(k).$$

Solving for **A** gives

$$\mathbf{A} = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where Δt is the "time step" of the process, and in this application should be fixed to the sampling rate of the tracking device. In the case of the tracking implementation the data rate is approximately 60 Hz for each anchor ($\frac{60}{4} = 15$ Hz for all anchors to read in new data)thus the time step (Δt) should be around $\Delta t = \frac{1}{15} = 66.7$ ms. This is not a strict equation but can be used as an early prediction as to where the ideal time step may lie; the time step most suitable to the task at hand can then be found through iterative testing and monitoring which value results in the desirable behavior.

At this point the measurement equation used to define matrix **C** is introduced to be

$$z(k) = \mathbf{C}s(k) + v(k) \tag{4.5}$$

and in vector form it is seen that

$$\vec{z}(k) = \mathbf{C} \cdot \begin{bmatrix} x(t) \\ \dot{x}(t) \\ y(t) \\ \dot{y}(t) \end{bmatrix} + \begin{bmatrix} v_x(k) \\ v_y(k) \end{bmatrix}.$$

It is realized that the position in the x direction is affected by the first two values in $\vec{s}(k)$ in addition to the noise parameter in the x direction while the latter two values and the noise parameter in the y direction are the deterministic factors of the position in the y direction. Thus, **C** needs to affect these values appropriately and is defined to be

$$\mathbf{C} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

such that

$$\vec{z}(k) = \begin{bmatrix} x(t) + \dot{x}(t) \\ y(t) + \dot{y}(t) \end{bmatrix} + \begin{bmatrix} v_x(k) \\ v_y(k) \end{bmatrix}$$

where v denotes the additive noise. The **R** matrix is set to reflect expected measurement deviations caused by electronic/random noise inherent to the antenna. **R** is defined as the standard deviation of the sensors multiplied by the identity matrix ($\mathbf{R} = \sigma^2\mathbf{I}$). The standard deviation for all anchors at up to 400 meters was found to have an average of approximately 1.8 cm (more on this in the Results chapter). This shows that the UWB implementation has very strong precision and the measurement noise is quite low. Fig. 4.19 shows simulations of the Kalman filter for a few different values of **R** based on σ^2. The simulations use a value of **Q** which has already been found to be effective for this application (shown later in this section) in order to best demonstrate how changing the **R** matrix affects the behavior of the filter.

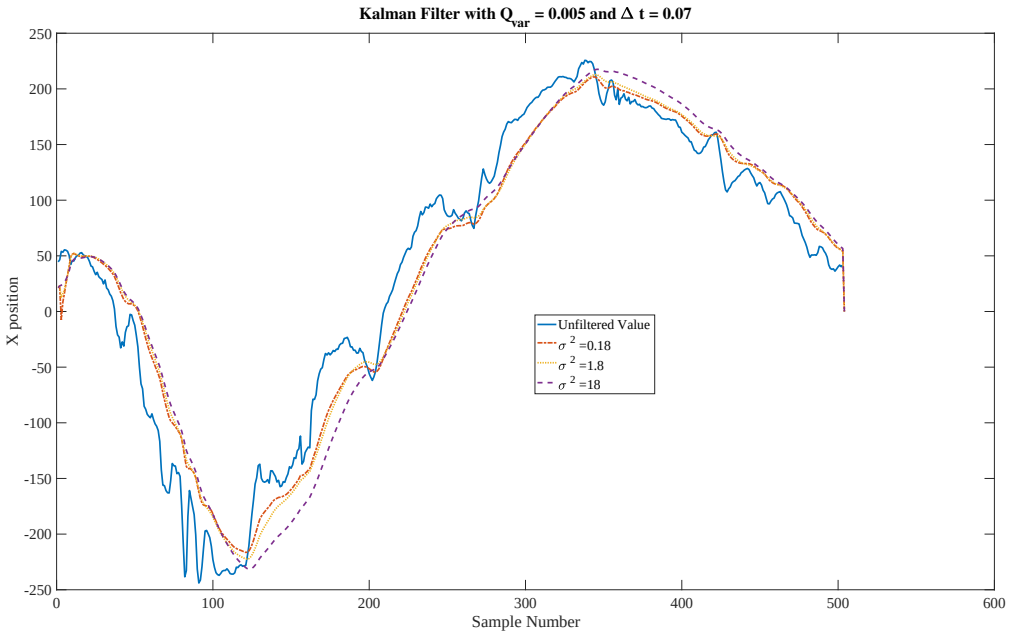

Fig. 4.19. Simulations of Kalman Filter Performance for Different R Values

Based on the information from the simulations and the known expected standard deviation the measurement matrix was set to

$$\mathbf{R} = \begin{bmatrix} 1.8 & 0 \\ 0 & 1.8 \end{bmatrix}.$$

The process noise matrix **Q** is slightly more challenging to quantify as it represents the feature that the state of the system changes over time in an unknown way. Since perfectly modelling a system is impossible, a simplification is performed such that the ”true” state is the previously predicted value and then shifted by this process noise. Thus, if the values in **Q** are large it means that the confidence in the original prediction is low because it is deemed necessary to shift the prediction significantly. For small values in **Q** a more subtle ”smearing” of the prediction is made signifying high trust in the prediction. Since there is no reason to think that the confidence in each prediction differs across the state of the system the matrix **Q** can be chosen to be $Q_{var} \cdot \mathbf{I}$. Fig. 4.20 shows simulations performed for different Q_{var} values with positions measured by the tracking device in the x direction.

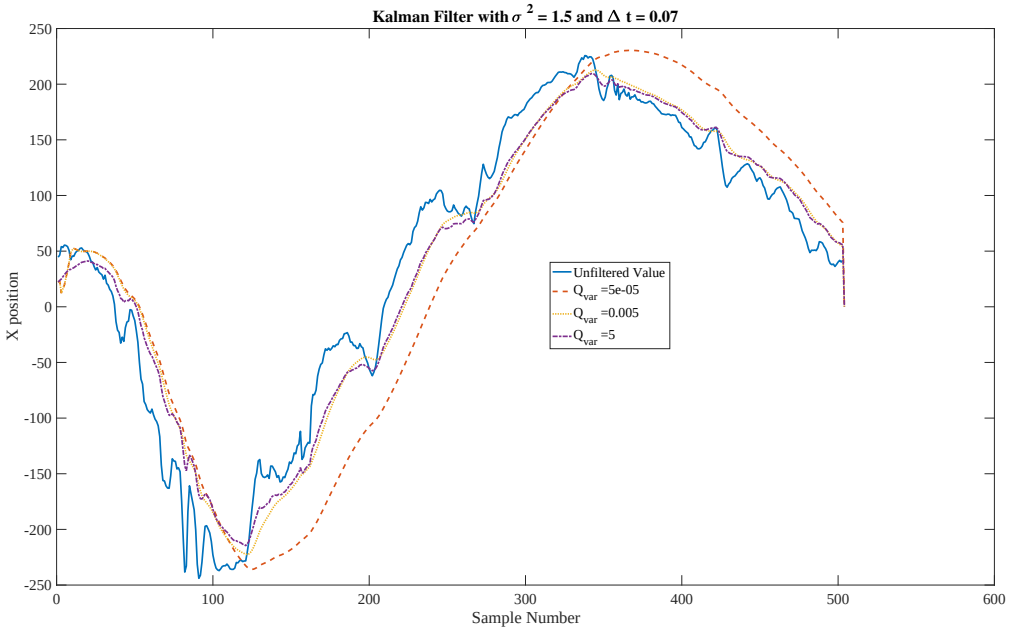

Fig. 4.20. Simulations of Kalman Filter Performance for Different Q Values

Based on these simulations using the set values of $\Delta t = 0.07$ seconds and $\sigma^2 = 1.8$ cm^2, $\mathbf{Q}$ was set to be

$$\mathbf{Q} = \begin{bmatrix} 0.005 & 0 & 0 & 0 \\ 0 & 0.005 & 0 & 0 \\ 0 & 0 & 0.005 & 0 \\ 0 & 0 & 0 & 0.005 \end{bmatrix}.$$

Fig 4.20 and Fig. 4.19 show that there will always be a trade-off when choosing acceptable matrices. The trade-off is between the speed at which the filter will react to intended movement and the amount of noise that it is able to filter out. It is important to find a balanced approach for a Kalman Filter that is not overly susceptible to noise but can still react in the appropriate amount of time to the movement of the target.

The final matrix to be described is the covariance matrix: $\mathbf{P}$. This is the component that will be updated along with the state of the system in order to characterize the uncertainty of the position. The matrix is a diagonal matrix that has to be set to some initial value before the Kalman process begins to update it recursively. This initial value could simply be set to the identity matrix but if you want the initial state to be found quickly some knowledge about the start state can be useful. For example, when testing the Kalman Filter the robot was often placed 2 meters away from the target, thus, the initial covariance matrix, $\mathbf{P}_0$, was set to be

$$\mathbf{P}_0 = \begin{bmatrix} 200 & 0 & 0 & 0 \\ 0 & 200 & 0 & 0 \\ 0 & 0 & 200 & 0 \\ 0 & 0 & 0 & 200 \end{bmatrix}.$$

A covariance matrix set with values closer to the first the system state allows the predicted state to reach the true state before a covariance matrix with values further from the true state. Furthermore, after a

few samples, regardless of the initial state of the system, the state prediction will behave the same way, which demonstrates that the initial covariance matrix only affects the early stages of the adaptive filter.

It takes extensive testing both with a static tracking device and a dynamic one to determine which values to assign to the Kalman parameters and how to define the matrices. The values chosen in the implementation of the robot were based on observations of these tests. Concrete data of the system's performance employing the Kalman algorithm can be seen in the Results chapter. The Kalman Filter is the final processing done with the readings taken in from the anchors and the initial position estimates given by the multilateration technique. The updated coordinates are then used to control the motors in order to determine the dynamic behavior of the robot.

4.3 Movement

The movement function is in charge of controlling the motors in order to follow the target. The result from the positioning function is sent via I2C from the central unit to a motor controller unit. The decision of having a separate microcontroller for controlling the motors was made with the objective of reducing possible disturbances in the ranging protocol timing and program flow. The movement function was implemented in an Arduino Mini Pro, which receives the target position coordinates via I2C and runs a proportional controller for setting the speed of the motors using the PWM outputs of the microcontroller.

4.3.1 Movement Algorithm

The movement algorithm receives the target position coordinates in X and Y. With that information it calculates the angle of the target in relation to the X axis, and uses the Y coordinate as the distance to the target. The algorithm will try to keep the target in front of the robot at all times. A proportional controller is implemented so that a bigger angle of deviation from the front of the robot leads to faster turns by the motors. Similarly, the farther the distance from the target to the robot, the faster the robot will move forward. The robot is kept at 2 meters minimum from the target to avoid a possible collision between them. Once the robot has reached the 2 meter boundary it will only rotate on its axis trying to keep the target in front.

4.3.2 Motor Control

The motor controller has similar requirements as the target control unit therefore the same Arduino Mini Pro board was selected for this task. For powering the motors, a L298 dual full-bridge driver [68] with a maximum output current of 4A is used. It allows us to power and control the four motors from the Arduino Pro Mini without any additional power stages. The Olimex BB-L298 board showed in Fig. 4.21 includes an L298 driver, safety components, and inputs and outputs connectors.

Fig. 4.21. Olimex BB-L298 dual motor driver board [69]

The motor controller, driver and motors connection was done following the BB-L298 datasheet. An additional 6V supply is needed for powering the motors and achieve the desire speed. The inputs to the motor driver are connected to the PWM pins on the Arduino Pro Mini (pins 3, 5, 6, and 9) so that the speed of the motors can be varied using the movement algorithm.

4.3.3 Chassis Structure

The chassis of the robot is a 4 wheel drive commercial chassis available at several electronics and robotics distributors. As the robot itself is not one of the main scopes of this thesis, a low cost and easy to build and control platform was chosen. Fig. 4.22 shows the final implementation of the system mounted on the 4WD chassis.

Fig. 4.22. Physical implementation of the system mounted on the 4WD chassis

5

Results

Several tests were run in order to evaluate the metrics and functionality of the system. The values deemed most crucial to asses were defined in the project scope of the background chapter of this thesis. These were found to be: accuracy, precision, complexity, robustness, scalability, and cost. To gauge these features appropriately, four separate testing scenarios were used:

- Ranging: one-to-one communication
- Positioning: locating the target while both it and the anchors are static
- Tracking: locating the target while it is moving and the anchors remain still
- Following: both the target and the anchors are moving and the robot prototype maintains a certain distance behind the tag

This chapter sets out to explain how these tests were set up and performed while also revealing the outcomes of the different procedures. Comments on the meaning of the results and their potential implications are then shown in the Discussion chapter.

A graphical user interface (GUI) was developed for capturing all data which makes it easier for setting up each test and verifying correct functionality. The GUI was developed in Visual Studio using C#. It allows the user to log the sampling frequency, ranging values, received power, raw positioning and positioning after Kalman filter in a text file with CSV (comma-separated values) format. The design of the GUI is shown in Fig. 5.1.

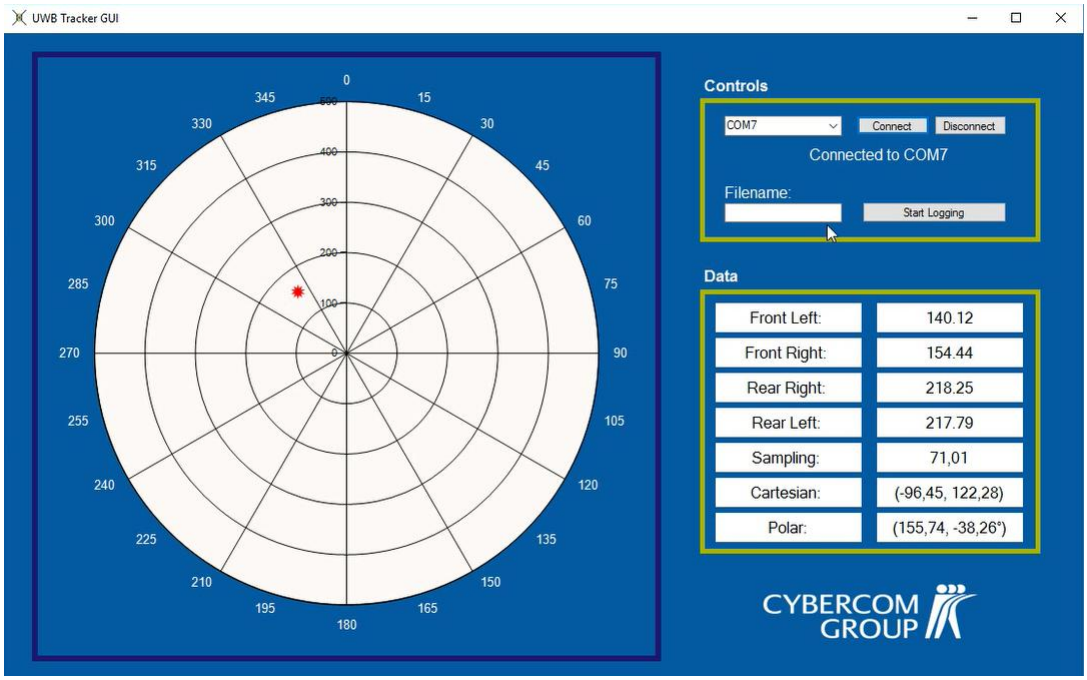

Fig. 5.1. Graphical user interface

5.1 Ranging

One-to-one ranging tests were set up by placing two modules separated by the distance to be tested facing one another, as seen in Fig. 5.2, and then taking 500 samples of the ranging results. These tests were carried out at ten distances ranging from 25 cm to 10 m and repeated using four different UWB modules for each distance. The objective of using different UWB modules was to ensure that all anchors on the robot produce similar ranging results and have similar accuracy values; this ensures that the results are repeatable across multiple modules. The tests were repeated three times with each module, and the samples taken were used to find the average error, standard deviation of the measurements, and the error percentage. These results are shown in Table 5.1.

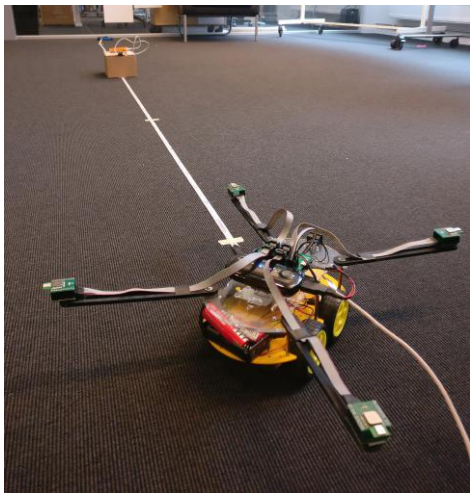

Fig. 5.2. Example of test setup for ranging functionality

Table 5.1 – Ranging results for DWM1000

Actual Distance (cm)	Measured Distance (cm)	Absolute Error (cm)	Percent Error (%)	Standard Deviation (cm)
25	20.98	4.02	16.10	1.69
50	52.28	2.28	4.56	1.40
100	101.63	1.63	1.63	1.43
200	199.29	1.71	0.86	1.40
300	299.66	1.34	0.45	1.37
400	399.58	1.42	0.35	1.48
500	511.44	11.44	2.29	1.49
600	580.10	19.90	3.32	1.79
800	760.27	39.73	4.97	1.81
1000	931.52	68.48	6.85	2.28

The absolute error is found by taking the difference between the measured distance and the actual distance, the error percentage is the ratio of absolute error and the actual distance between the modules (d), and the standard deviation is found in the conventional way for N samples. These methods are shown mathematically in (5.1), (5.2), and (5.3) respectively:

$$\xi_{\text{ERR}} = |\hat{d} - d|, \tag{5.1}$$

$$\xi_{\%} = \frac{\xi_{\text{ERR}}}{d} \times 100, \tag{5.2}$$

$$\sigma = \sqrt{\frac{1}{N}\sum_{s=1}^{N}(x_s - \mu)}. \tag{5.3}$$

The next step for checking UWB capabilities in one-to-one ranging is to see how the performance is affected when line-of-sight is removed. In this test setup an obstacle was placed between the target and the anchor at 150 cm and ranging was done for the target placed at 300 cm. The tests done to see the effect of blocked line-of-sight were run only at this distance. Again the ranging experiment was done with all four anchors; the average values found for each parameter are shown in Table 5.2.

Table 5.2 – Ranging results for DWM1000 in NLOS scenario

Actual Distance (cm)	Measured Distance (cm)	Absolute Error (cm)	Percent Error (%)	Standard Deviation (cm)
300	302.81	2.81	0.97	5.86

5.2 Positioning

The positioning tests were performed with both the target and the reference anchors remaining static throughout each test, as shown in Fig. 5.3. This method aims to determine the accuracy of the flipped

UWB system using the implemented multilateration technique. The test was performed by placing the target at a known position (x and y) relative to the robot. Then 500 samples were taken at each point three separate times to find the average accuracy and precision of the system in this type of application.

Fig. 5.3. Example of setup for positioning tests

One objective of the thesis was to show that using UWB allows for anchors to be placed in close proximity to one another and to observe how the distance between the anchors affects the functionality of the system. Therefore, all tests were done for 40 cm between anchors as in Fig. 4.16, for 30 cm between anchors, and for 20 cm between anchors. The measurements for all three configurations are shown in this section. In order to simplify the way the results are presented here the actual position is defined as the distance in the x and/or y direction from the origin of the robot. For example an actual position of ”x” cm includes tests taken at (0,x), (-x,0), (x,x), (x,-x) etc. The results of these experiments are shown in Fig. 5.4

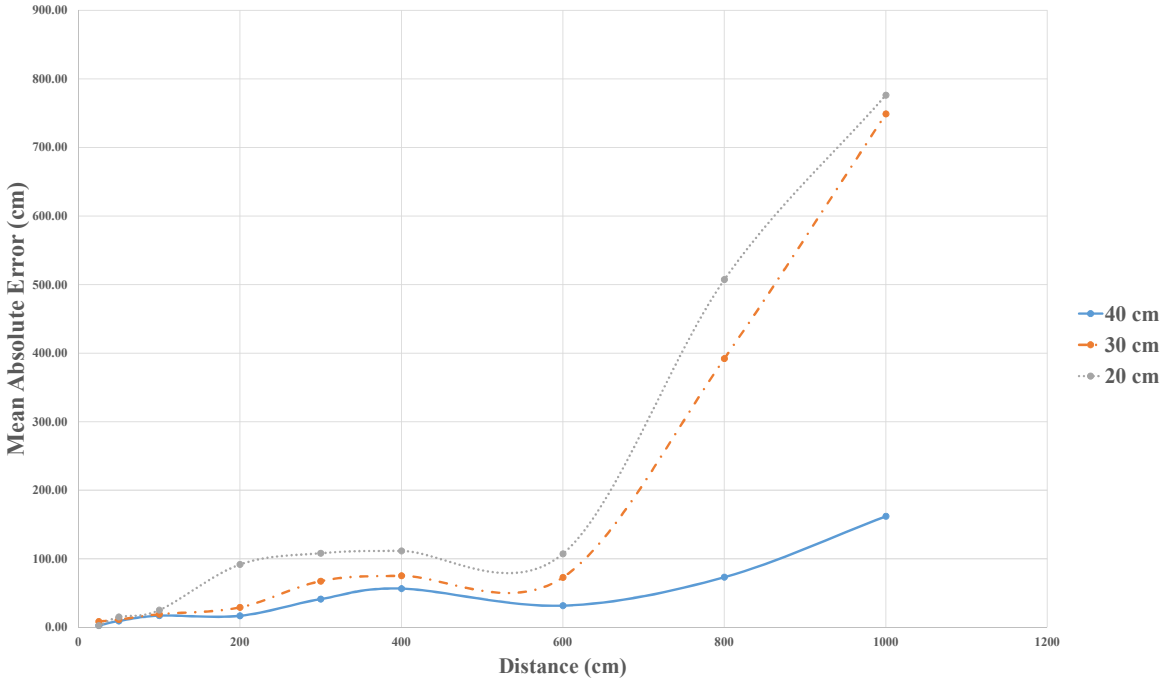

Fig. 5.4. Positioning error for three different system configurations

Error percentages at these distances as well as the standard deviation of the measurements are shown in Table 5.3.

Table 5.3 – Static positioning statistics for three configurations

Actual Position (cm)	Percent Error (%)			Standard Deviation (cm)		
	40 cm	**30 cm**	**20 cm**	**40 cm**	**30 cm**	**20 cm**
25	10.96	33.96	9.68	1.38	1.30	1.47
50	18.28	22.22	29.78	2.13	2.82	3.44
100	16.90	19.07	24.91	3.78	5.52	5.53
200	8.36	14.50	45.81	7.16	10.63	11.36
300	13.71	22.44	35.99	17.39	23.25	20.91
400	14.10	18.76	27.84	17.07	22.06	35.27
600	5.25	12.11	17.87	28.46	101.15	90.30
800	9.14	48.99	63.38	94.73	384.63	343.86
1000	16.19	74.87	77.62	104.67	147.04	408.01

The error for two-dimensional positioning is found to be

$$\xi_{\text{ERR}} = \sqrt{(\hat{x} - x)^2 + (\hat{y} - y)^2} \tag{5.4}$$

where $\hat{x}$ and $\hat{y}$ denote the position calculated by the system. Standard deviation is found as it was previously in (5.3).

The positioning tests were performed again with an obstacle placed equidistant from the anchors and the target to see how the system behaves when LOS is impeded. The outcomes for these tests are shown in Table 5.4.

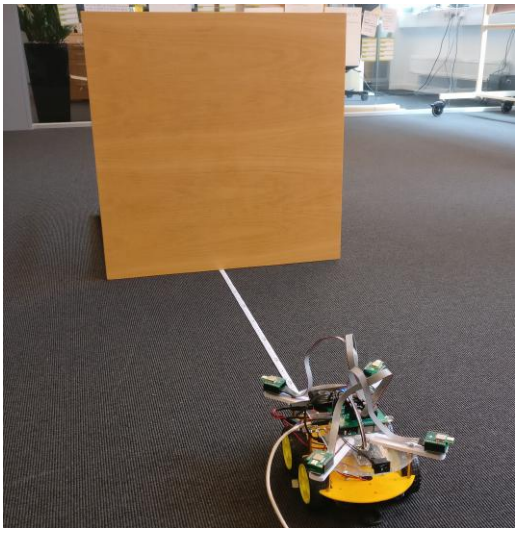

Fig. 5.5. Example of setup for NLOS positioning tests

Table 5.4 – Static positioning results for three configurations with NLOS

Actual Position (cm)	**Position Error (cm)**			**Standard Deviation (cm)**		
	40 cm	**30 cm**	**20 cm**	**40 cm**	**30 cm**	**20 cm**
200	26.97	71.62	66.18	7.27	11.74	17.74
300	27.76	99.85	88.23	37.17	28.06	35.78

5.3 Tracking

When tracking a moving object, a third dynamic of the UWB system's capabilities is tested: the performance of the adaptive Kalman Filter. In order to asses how well the robot tracks a moving target while remaining static, a smaller toy robot (the mBot V1.1 from MakeBlock) was fitted with the tag module and programmed to travel in a circle around the UWB tracker, as shown in Fig. 5.6. This was done to ensure that the target would be moving with constant speed and at a known radius from the tracking device.

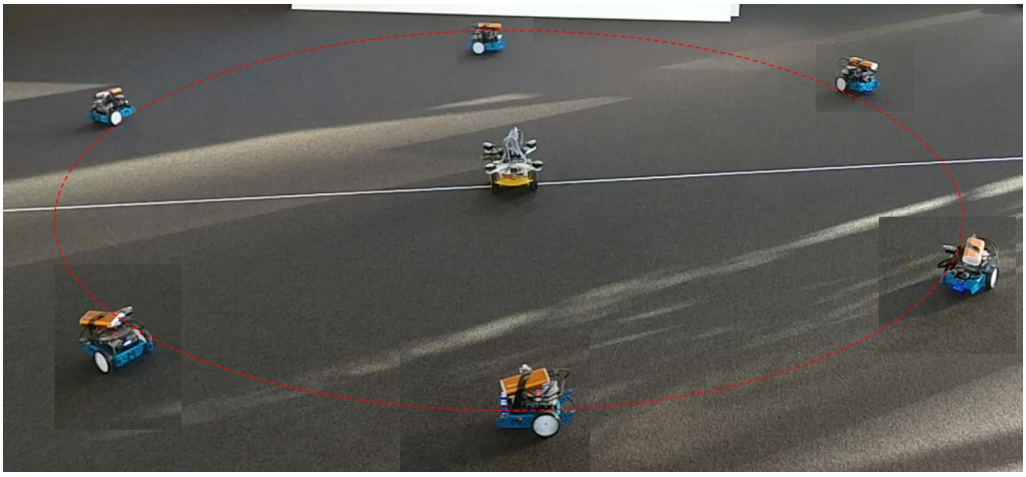

Fig. 5.6. Example of setup for tracking tests

The tests were performed at 1 and 1.6 meters and once again with an obstacle impeding LOS to test performance with NLOS at a 1.6 meter radius.The results are shown in the figures below for the standard setup with the tracker setup of 40 cm between anchors. For the NLOS tests, the obstacle was placed at (0,-100) and spanned 50 cm in each direction. Figs. 5.7-5.9 show the actual path taken by the smaller robot and the position estimates found dynamically by the tracking system.

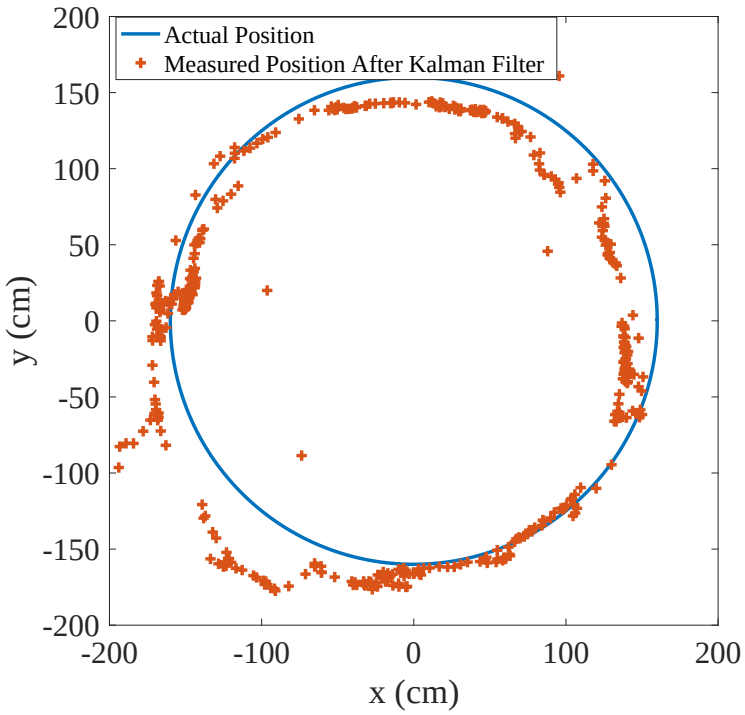

Fig. 5.7. Tracking target moving with radius of 160 cm: 40 cm setup

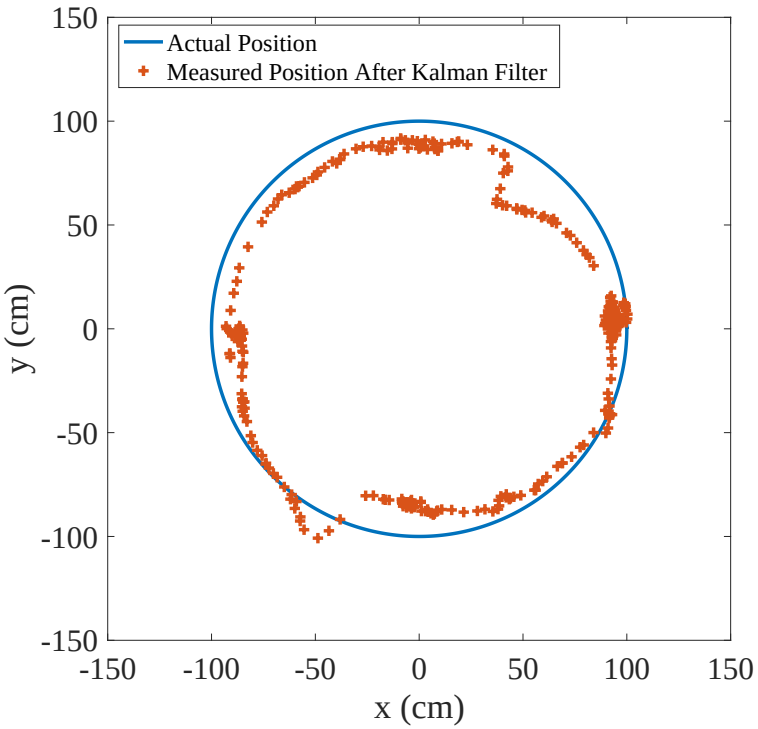

Fig. 5.8. Tracking target moving with radius of 100 cm: 40 cm setup

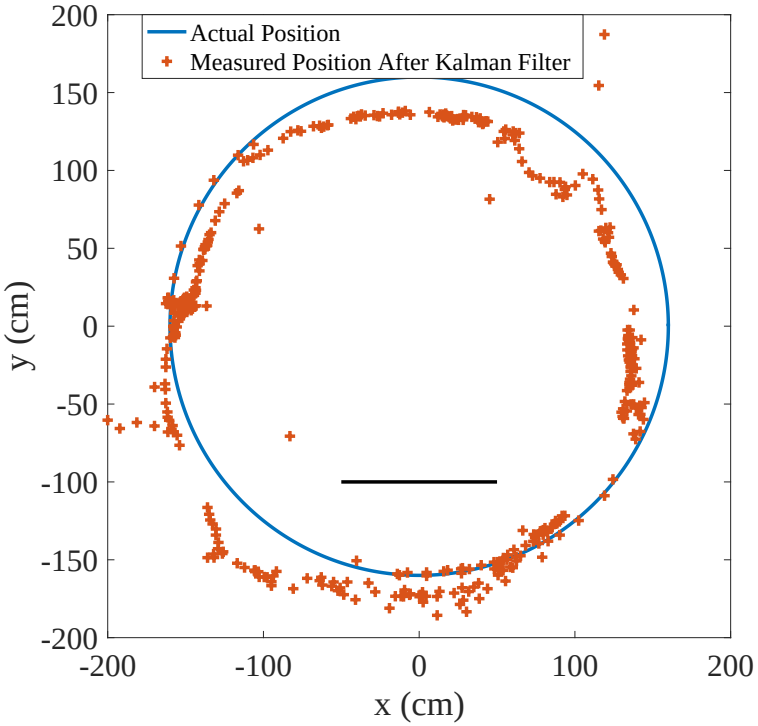

Fig. 5.9. Tracking target moving with radius of 160 cm: 40 cm setup NLOS

These same tests were repeated for the 30 cm setup, and the plots of the tracking results are shown in Figs. 5.10-5.12.

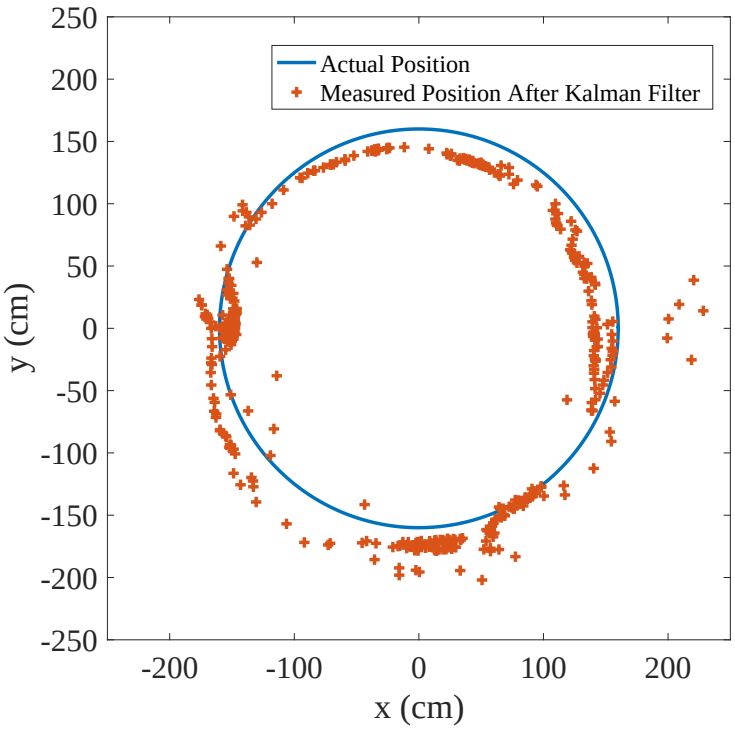

Fig. 5.10. Tracking target moving with radius of 160 cm: 30 cm setup

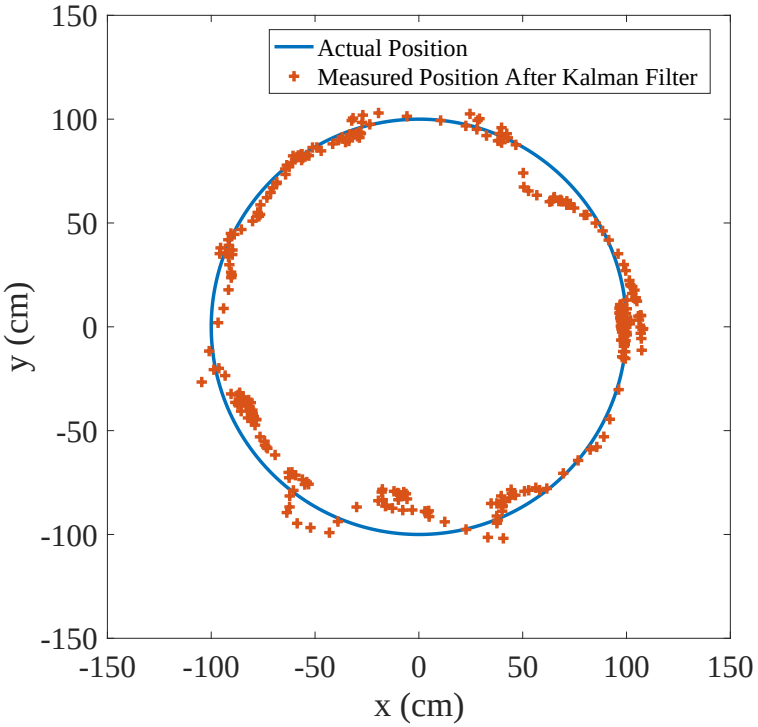

Fig. 5.11. Tracking target moving with radius of 100 cm: 30 cm setup

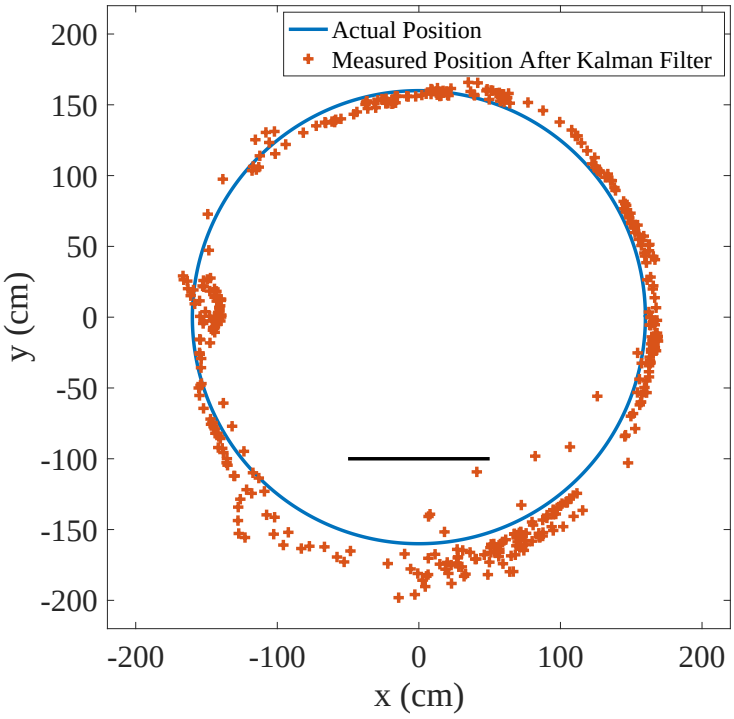

Fig. 5.12. Tracking target moving with radius of 160 cm: 30 cm setup NLOS

Once again the tracking tests are repeated for the third configuration of anchor distances, and Figs. 5.13-5.15 show the resulting data plots for these tests.

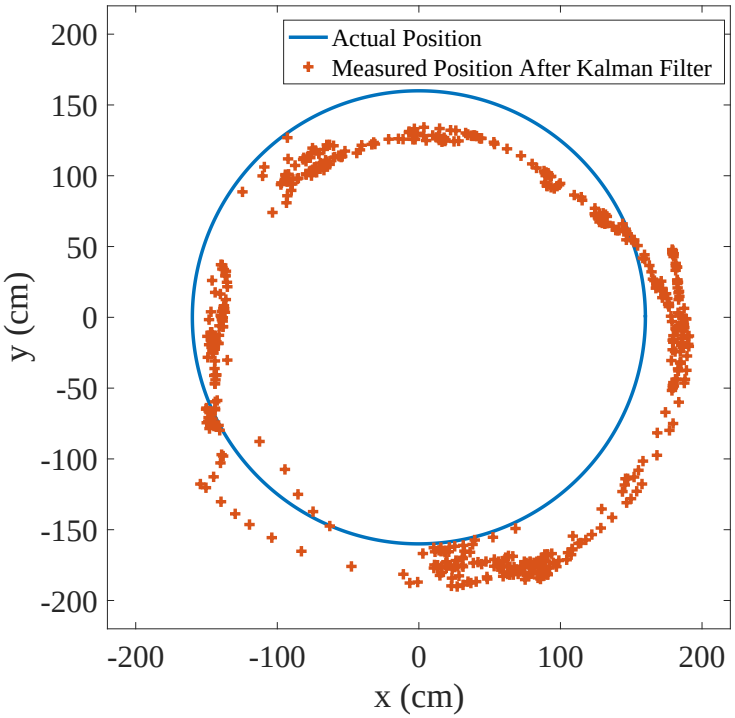

Fig. 5.13. Tracking target moving with radius of 160 cm: 20 cm setup

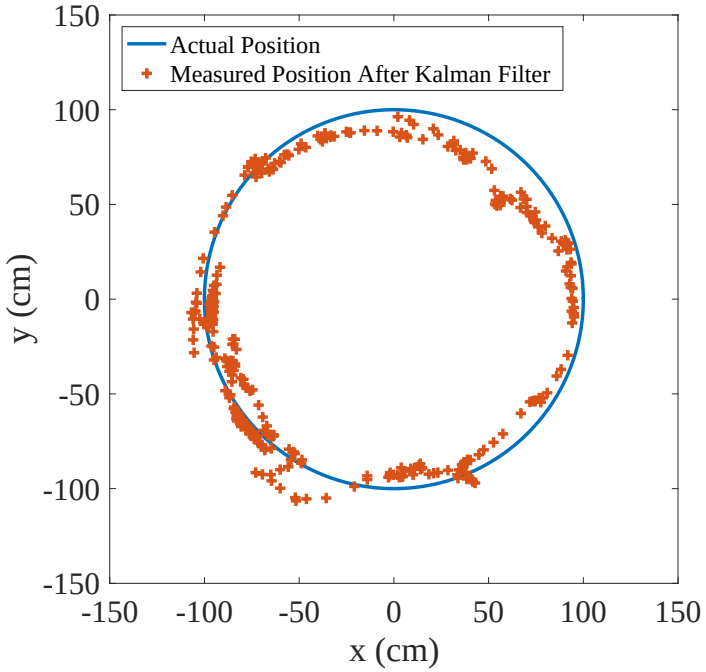

Fig. 5.14. Tracking target moving with radius of 100 cm: 20 cm setup

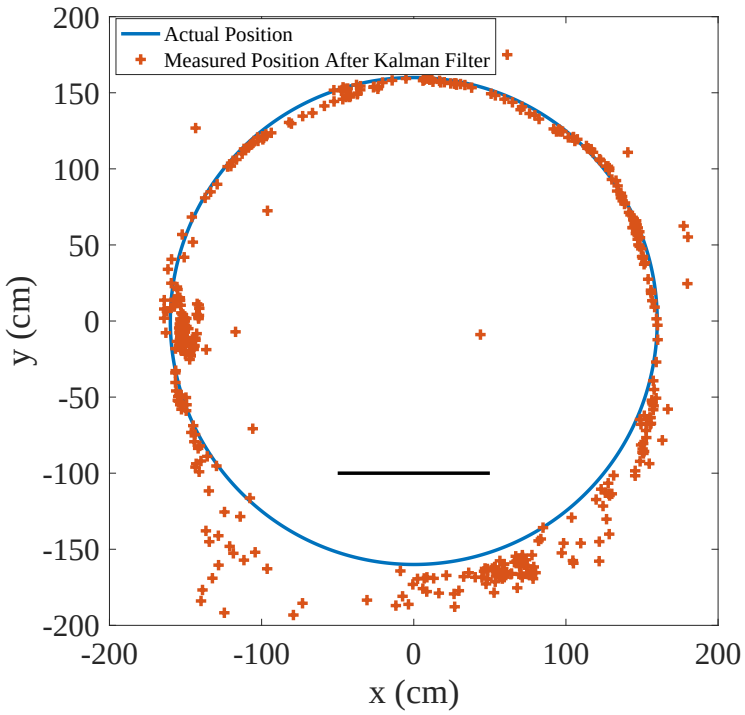

Fig. 5.15. Tracking target moving with radius of 160 cm: 20 cm setup NLOS

5.4 Following

The evaluation of the following capabilities of the tracker was done by fitting the target to the mBot robot and making it move in a circle and having the tracker follow it, as shown in Fig. 5.16. The distance from the tracker to the target was set to 60 cm. The tests were carried out for three different setups of distance between anchors: 20 cm, 30 cm, and 40 cm. The tests were repeated twice for a total traveled distance of 30 m with each setup. Table 5.5 shows the mean distance between the tracker and the target for each of the setups and the standard deviation of this measurement.

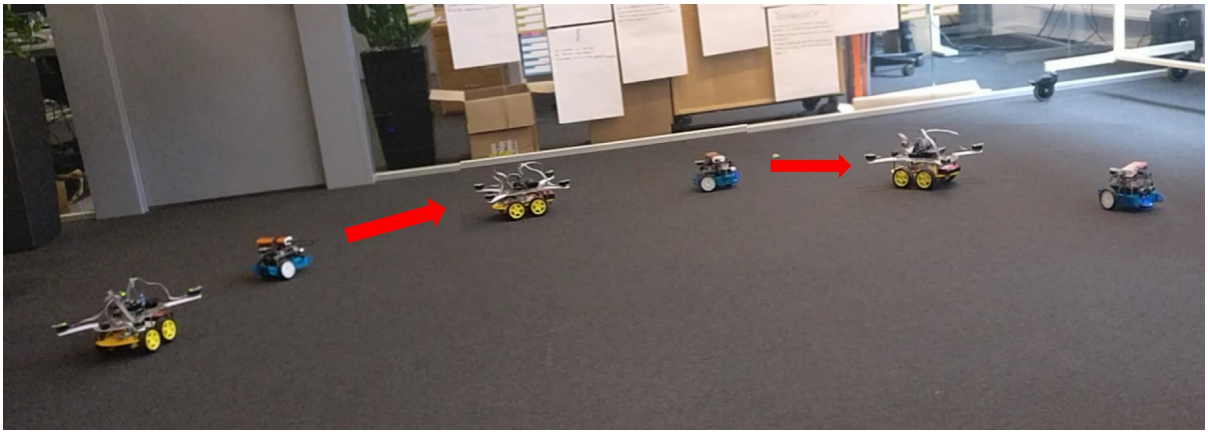

Fig. 5.16. Example of setup for following tests

Table 5.5 – Following results

Setup	Mean distance (cm)	Std. dev. (cm)
20 cm	64.48	11.10
30 cm	63.87	9.68
40 cm	69.19	11.20

5.5 Timing and Power Analysis

The timing and power specifications of the tracker device and the target were studied using a LeCroy WavePro 960 oscilloscope and an AP015 current probe. The probe was placed at the output of the 5 V power source that supplies the Arduino and the DWM1000 modules.

Fig. 5.17 shows the current drawn by the tracking device. Each of the four pulses grouped together is the ranging function of one anchor. The lower current limit is marked by the cursor and is around 250 mA. This lower current is related to the Arduino nominal current plus the idle current of the DWM1000 modules. When an anchor is in the ranging process, the current increases to approximately 400 mA that lead to a peak power of 2 W. These additional 150 mA are related to the TX/RX current of the DWM1000 and it coincides with the specification.

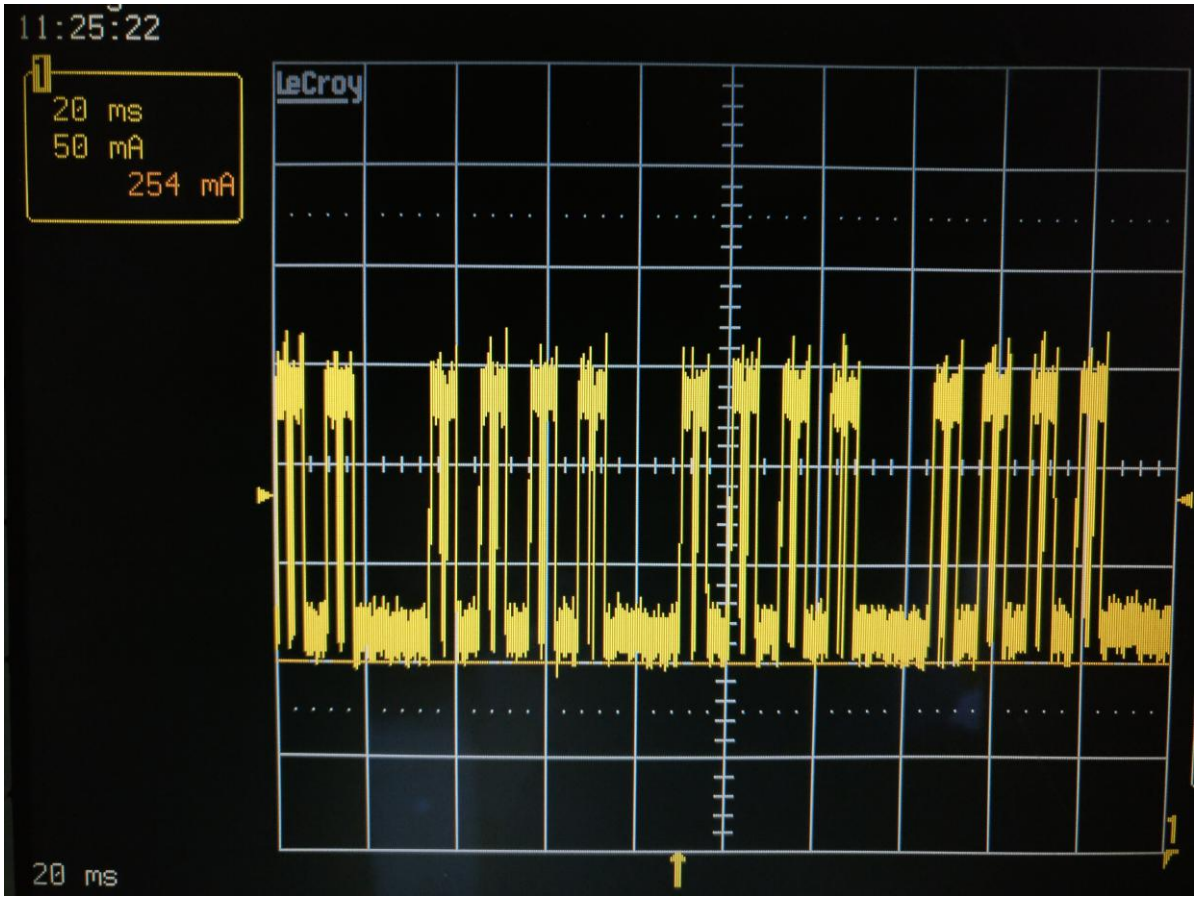

Fig. 5.17. Current drawn by the tracker device

From Fig. 5.17 it is also possible to analyze the timing of the ranging function. Ranging for one anchor takes 6 ms and the time between each ranging poll is 11 ms, which leads to a single ranging frequency of 90 Hz. It takes 38.7 ms to complete the four distance measurements which corresponds to a complete (new readings from all anchors) ranging frequency of 18 Hz. However, since a new position is calculated with each new ranging measurement, the only factor slowing down the process is the time it takes to calculate the new position. Therefore, the actual data rate of the system is found to be approximately 70 Hz. The time between each complete ranging method is probably related to the serial port data write and the data transfer to the Arduino mini controlling the motors.

The current drawn by the target is shown in Fig. 5.18. The cursor is set at 279 mA which is the top part of the measured current and leads to an average power of 1.4 W. As opposed to the current pattern seen in the tracking device, the target keeps a high current and has some pulses going down at some times. This is because the target module is in the receive state all times, except for short intervals after it finishes a transmission. Each ranging function can be seen as a group of very short pulses that represent each of the messages in the ranging protocol. The timing can be calculated again for the target device, and the results are the same as for the tracking device: 90 Hz for one-to-one ranging and 70 Hz for the complete data rate.

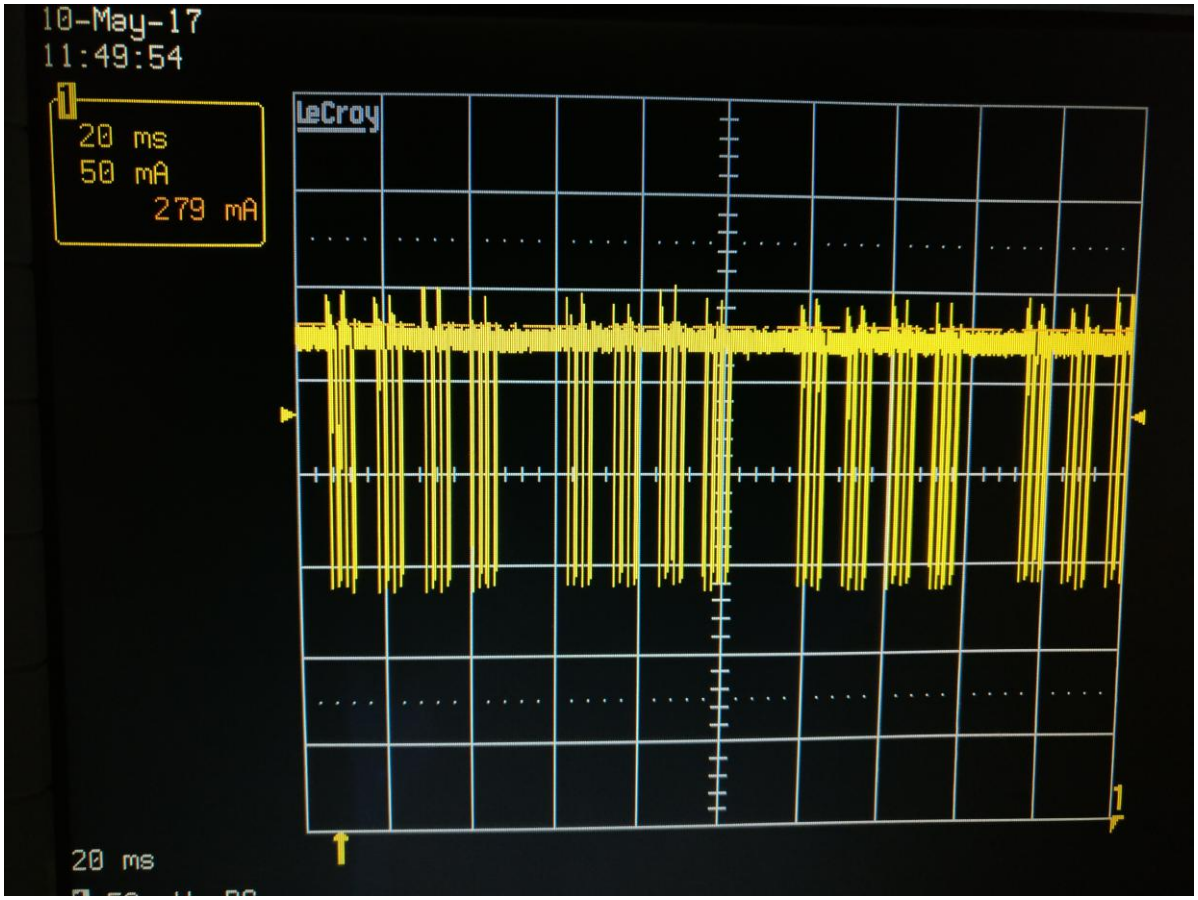

Fig. 5.18. Current drawn by the target

6
Discussion

In this chapter we discuss different aspects of the system. The implementation is analyzed to see which design decisions were beneficial and which ones could have been changed for achieving a better functionality. The results will be discussed by comparing them with previous studies found in literature as well as evaluating how the system behaves in different scenarios. The main objective of analyzing the results is to validate the hypothesis of the *flipped* anchor topology and evaluate the feasibility of such a system in the proposed applications.

6.1 Ranging Implementation and Results

The software implementation of the ranging functionality was satisfactory. The configuration used in the DWM1000 was optimal for the desired application (short range and high accuracy). A sampling frequency of around 70 Hz was achieved with the asymmetric double-sided two-way ranging scheme. Although the sampling frequency can be increased by using fewer messages in the ranging scheme or using a TDOA method, the achieved sampling frequency was shown to be sufficient for the tracking and following functionalities of the prototype.

The ranging results exceeded our expectations. According to the literature review carried out at the beginning of this project, the accuracy of UWB ranging systems is between 10 cm and 30 cm, for most implementations. The results presented in Table 5.1, show an average error of 15 cm for ranges between 25 cm and 10 m, however, for ranges between 50 cm and 4 m, the average error is 1.7 cm. As the main desired application is to follow an object in a short range scenario, the 1.7 cm gives great projection on the feasibility of the flipped anchors topology.

The ranging results show that the error increases proportional to the distance. This can be related to the short range configuration of the DWM1000 and/or the offset correction. The latter was implemented with only a few measurements between 1 m and 4 m, and it is possible that a re-calibration is needed for higher distances. Even though the error increases significantly with distance, the standard deviation has a minimal change. This behaviour shows the precision of the UWB modules and provides confidence in the repeatability of the tests.

The final ranging test aimed to compared the best result of a direct line-of-sight scenario with the a non-line-of-sight situation. It can be observed in Table 5.2 that the absolute error increases from 1.34 cm to 2.81 cm, which is an increment of 110%. Even though this number might seem high, the percent error is still low, which hints that the flipped anchor topology could work in NLOS scenarios. However, the standard deviation of the sample increases from 1.37 cm to 5.86 cm, meaning that the measurements are not as precise in the NLOS situation.

6.2 Positioning Implementation and results

In order to perform positioning with this system, multilateration was used along with a linear least squares algorithm. Positioning measures how accurately the system can pinpoint the non-moving target in two-dimensional space. The goal when evaluating this implementation was to see how accurate and precise the system is when attempting to find a static target,as well as how these results are affected when the system is scaled up or down. Empirically (by viewing the GUI), the LLS method along with down projection of the initial guess from the LLS algorithm were deemed sufficient to find the approximate location of the target with high accuracy and at a speed that is required for this type of system to be useful. The scalability of the prototype was examined by changing the distances between each anchor on the robot.

The Results chapter shows metrics on how well the solution performs at varying distances with three different anchor configurations and in both LOS and NLOS situations. The findings reveal that at distances up to 100 cm all three setups can find the target within 25 cm with the highest accuracy being 2.42 cm at 25 cm away. As the distance of the target from the origin increases, accuracy suffers linearly at first, but there exists a threshold at which the error spikes rather than continuing linearly (at around 200-300 cm depending on the configuration). For example, this is shown for the 30 cm setup where from 100 cm to 200 cm the error only increases by about 10 cm, but from 200 cm to 300 cm the error grows by as much as 35 cm. Similar behavior can be observed in the other configurations as well.

Noticeably, the 40 cm configuration achieves the highest positioning accuracy followed by the 30 cm and 20 cm setups respectively. These results are expected since a shorter distance between the anchors means that errors which occur in ranging have a more significant effect on the multilateration technique. However, although the 20 cm setup performed the worst in comparison to the other two architectures, it was found that even this system performed well at short distances. This is significant since it is desirable to be able to package the tracking system as compactly as possible in some applications. While the longer arm composition may be a better solution to use as a static tracking system, the results found in the positioning tests indicate that a smaller system can still be useful when required for short distance positioning. Standard deviation behaved similarly in this sense. As distance increased the deviation also increased, and it was (in general) larger for the shorter setups. These results imply that these smaller systems will perform more poorly, but, as stated previously, that does not mean that they would have no use at all.

When an obstacle was introduced to test NLOS performance conflicting results were found in terms of accuracy. In certain situations such as for the 20 cm setup the accuracy appears to increase significantly (approximately 25 cm decrease in error), but in others such as the 30 cm case there is a large decrease in accuracy. At first glance this seems strange, but when viewing the large discrepancy in standard deviation from the LOS situation it becomes clear that the average accuracy findings suffer greatly from the deviation. This discovery along with the relatively few simulations run with NLOS makes it difficult to draw any clear conclusion from the readings. The tracking tests in the following section demonstrate the effect that NLOS has on the system more clearly and concretely.

6.3 Tracking Results

Tracking involves locating a moving target in real time while the anchor base remains still. Accuracy and precision remain important in this case, but a new element is now critical: the adaptive Kalman Filter. The filter ensures that erroneous values are kept to a minimum and its predictive capabilities allow for quick updates to new position estimates which are required for a moving target. As mentioned in previous sections there are many parameters that affect the performance of the Kalman Filter, and many simulations and empirical tests were run to choose the values which best allowed for the system to behave as intended. Rather than including figures for tests of each different combination of parameters the report includes different situations for the same setup as defined in the Kalman Filter section of the

Implementation chapter.

From the figures shown it is observed that the system tracks the moving object with slightly better accuracy when the anchors are a larger distance apart. The difference, however, is slight which is surprising based on the large error differences between configurations during positioning tests. The reason for this may be due to the fact that the different setups do not have very large discrepancies in accuracy for these shorter distances, and it may be necessary to perform tests further away to see any significant distinction. This was not done in this thesis due to space constraints. From viewing the plots it is clear that at the shorter 1 meter distance the tracking algorithm performs better as expected. Furthermore, it is noticed that the density of points is not consistent throughout the circle plotted by the tracking system for the 1.6 meter measurements. This occurs when there is high error in ranging from each of the anchors due to the incidence angle of the incoming signal from the target to each anchor. According to Fig. 5.7 - 5.15 such situations most often occur when the target is at a diagonal from the origin. This causes the positioning algorithm to ”retreat” and estimate positions incorrectly near previously recorded points until the error is reduced once again. When appropriate readings can be found again the estimates quickly return to the new correct measurements. However, due to the small period of erroneous values, this leaves a gap between several points throughout the dynamic tracking. It is possible that this issue is not significant when tracking an object moving more erratically or perhaps one not at a constant speed, but in order to demonstrate how well the system can perform this function it was desired to have strict control over the object's movement so that results could be stated more definitively.

For the NLOS situation, there is sporadic behavior when the target is blocked by the obstacle. However, the deviations caused are not quite as significant as one may expect from results garnered in the ranging and positioning NLOS tests. The system is still able to make fairly appropriate estimates of the target position.

6.4 Following Results

The tests performed for following evaluate the ability of the tracker to shadow another object's movement. They assess both the positioning functionality in a dynamic environment as well as the movement function, which controls the motors based on the position of the target. The results presented in Table 5.5 show a mean absolute error of 5.84 cm which is less than 10% of the specified distance between the tracker and the target.

The average standard deviation among setups is 10.65 cm. The standard deviation represents the variation in the distance between the tracker and the target while the test is performed. The high results in standard deviation are most likely linked to the fact that the tracker has a higher speed than that of the mBot robot; given this, the tracker would approach the target very quickly and then stop until the target surpassed the 60 cm boundary again. This constant approaching and stopping generates an error in the following precision as can be seen in the standard deviation.

One of the main advantages of using the following functionality over the tracking functionality is that the tracker always tries to keep the target in front of it (an x-coordinate close to 0 and a positive y coordinate). This functionality reduces the error linked to the incidence angle between the target module and the anchors. As seen in the tracking tests, when the target is moving around the anchors the error in the positioning increases depending on the angle of incidence between the DWM1000 module in the target and the modules in the anchors.

The capabilities for following were also tested in other scenarios that are not reported in this document due to the impossibility to repeat the tests with a high level of confidence. These tests were mostly carried out by following humans in indoors and outdoors environments. The tracker performs as intended: following the target and maintaining the predefined distance. It showed high reliability while moving in unobstructed scenarios as well as more complex environments such as an office with doors, walls, and divisions.

Even though the positioning and tracking results showed high accuracy in non-line-of-sight scenarios, the following functionality showed a high degree of disturbance when direct line-of-sight is permanently removed. For example, carrying the target in the front pocket while the tracker is following behind, leads to an unpredictable behavior of the tracker, which can crash against the followed object or a wall. However, when the line-of-sight is removed for a short period of time (for example a person walking between the target and the tracker), the functionality and availability of the tracker are barely affected, which implies an improvement over machine vision systems.

6.5 Possible Applications

The results have shown that UWB ranging gives high-accuracy results, which makes the tracker ideal for applications where the distance between two objects is to be found. Event though the accuracy of the positioning is not as accurate as the ranging, it is still better than other positioning systems and technologies such as GPS, WiFi or BLE at distances within a few meters. Nevertheless, results also showed and increase in the positioning error proportional to the distance which limits the field of application to short range scenarios. The configuration of the DWM1000 UWB module can play an important role in this limitation and is to be studied to understand the trade-off between accuracy and range.

Even though the position accuracy was not as high as expected, the tracker showed very good following capabilities using the different anchor setups. This shows the suitability of the framework to be implemented in these types of applications. As mentioned previously, the error-range relation will limit the functionality of the tracker when following objects and will set a boundary to the possible following applications.

In general, the framework presented in this thesis can be further developed and used in several applications. For example, it can be used for tracking and following humans in indoor or outdoor environments. The target can also be a robot or a series of robots, allowing the user to build autonomous convoys or caravans. Furthermore, the tracker can act as a portable high-accuracy (compared to other systems) tracking system that allows for positioning multiple tags in a nearby area without the need of installing anchors or references across the room.

7

Conclusion

A framework for a relative real-time tracking system based on UWB has been presented. The tracker prototype fulfills the specifications defined at the beginning of this project and proves that the *flipped* anchors method is a viable solution for tracking and positioning. The results show that UWB ranging has a high level of accuracy and precision reaching a minimum error of 1.34 cm at 300 meters with a standard deviation of only 1.37 cm, making this technology ideal for this kind of system. The accuracy was also high with the implemented filters and positioning schemes, achieving an error of only 16.90 cm with a 3.78 cm standard deviation at 100 meters. This surpasses the desired goal of achieving accurate position estimates within 30 cm. The capabilities for following of the prototype were also tested and showed exemplary functionality and reliability at distances within 5 meters.

Due to the high data rate of 70 Hz it is rare for the system to miss information for a significant amount of time. This coupled with the inherent material penetration properties of UWB allows for a system that rarely produces completely erroneous results or has a hard failure. This type of robustness is important as it was a goal set for the project and allows reliable demonstration of the final product.

The Future Work chapter provides many ways of expanding upon this implementation and improving its performance. The thesis was completed using common components and the code will be made public through Cybercom AB's Github repository in order to allow for improvements without a large number of restrictions.

The final prototype including all necessary components had a cost of approximately $295. It is difficult to categorize this value as low or high since there are very few systems that utilize this same approach. However, indoor positioning systems utilizing UWB can cost as much as $670 [58]. In terms of power consumption, the results show a 400 mA peak consumption in the tracking device, which can be divided into 250 mA drawn by the Arduino M0 and 150 mA drawn by the DWM1000 when transferring or receiving data. It is clear that the main sources of dissipation are these two components and any attempt to reduce power should start by re-evaluating this selection.

Lastly, the thesis set out to verify if an UWB system can remove the necessity to have direct LOS between the anchors and the target. Based on the results it appears that such an accomplishment is possible. Although many findings in this work when dealing with LOS are inconclusive, they show promise in terms of what may be accomplished if this implementation is expanded. The ranging capabilities of the UWB modules show that they are certainly capable of penetrating solid material. Furthermore, the tracking and following tests showed that the device can competently continue to estimate the position of the target if there is a temporary disturbance in LOS. Issues arise when LOS is constantly unavailable; this can lead to high obfuscation of signals and erroneous positioning estimates.

The framework for a relative real-time tracking system is the first stepping stone for consumer products that wish to have the capability to track and follow a specific target with a compact solution. This thesis has laid out the groundwork for further implementations that can enhance the capabilities of such a system in order for it to be a viable approach to solving common problems encountered in tracking devices. The thesis proved that the development of such a system is not only possible, but that it has functionality that exceeds expectations in terms of accuracy and precision. Furthermore, this work has been done in order to allow for a robust and scalable platform to allow for further enhancements.

8

Future Work

The framework was implemented in a way that can easily be developed further. This was achieved by using the Arduino platform which is widely known by engineers both in the embedded and non-embedded fields. The Arduino platform has a large amount of documentation and open communities that help in the development and troubleshooting of applications. Moreover, the interaction between the Arduino microcontrollers and the DWM1000 was based on the *arduino-dw1000* library, which uses standard nomenclature and functions for configuring and controlling the DWM1000 modules from the Arduino. Scalability was a concrete goal of the project and by using components and methodology that is fairly common, the ability to grow the framework is simplified.

The main purpose of this thesis was to develop a framework for relative positioning rather than an end user application. Therefore, the future work is presented from the framework perspective, that is, how could the framework be improved for better functionality or what is missing before it can be turned into a final application.

The results showed a trade off between range and accuracy that limits the possible applications and functionality of the system. The observed trade off is valid for only one configuration setup which was tested in the DWM100, however, other configurations and combinations of setup parameters are to be tested to better understand the relation between range and accuracy. With better comprehension of this relation it would be possible to define different modes of operations depending on the requirements of the application which could be either long range and lower accuracy or short range and high accuracy. The final configuration used on the DWM1000 will also affect the power consumption of the system and it can be interesting to study this relation in order to characterize the DWM1000 power consumption according to its configuration.

The ranging protocol showed to be reliable and fast enough for the desired application, however, improving the sampling frequency of the ranging would improve the resolution of the following functionality and allow the implementation of adaptive filters of higher complexity. One way to increase the sampling frequency is to re-use the communications messages from the target to the anchors. For example, instead of sending a poll for every anchor, just send one poll from the target and use the received time at each anchor for the ranging protocol. Another way for increasing the sampling frequency is to use TDOA instead of ADS-TWR. In that case, only one message from the target to all the anchors would be enough for positioning the target. However, it relies on the clock synchronization on the anchors side which is not permitted by the DWM1000 module. The robustness of the prototype is quite sound, rarely failing and maintaining high data rates. However, this can still be improved upon by the previously mentioned methods as higher data rates lead to more readings to simplify tracking and following.

Since the beginning of the project is was proposed to use four anchors as it represents better the flipped version of an indoor positioning system using four references in the corners of a room. It was assumed that four anchors placed in a small robot would give a better omnidirectional symmetry and could behave better with a target moving around it. Furthermore, it was also assumed that using four anchors gives a better accuracy than a system with less anchors. Nevertheless, it would be interesting to study the possibility of implementing the system with three anchors or less as this could reduce the cost, complexity, size, and power consumption of the device.

Another functionality that, although not critical, can improve the reliability of the system is fault tolerance. Several functions can be implemented in order to handle errors or failures in the system. For example, upon the failure of one of the anchors, the system could go into a graceful degradation mode where only three anchors are used for positioning the target. Parameters as the first path power, received power and TOF can be monitored and combined to detect errors in the measurements and provide higher confidence.

As mentioned in the implementation section there are many ways to implement positioning methods and adaptive filters. There was not enough time during this thesis work to attempt numerous methods for these functions, but for others continuing the work there are several possibilities that may lead to improvements. For instance, the Kalman Filter used is the simplest form of this adaptive method. The algorithm has other implementations such as the Extended Kalman Filter (EKF) and the Iterative Kalman Filter (IKF). Particle filtering is also a useful method to perform adaptive error correction as well as a technique known as elliptical gating. Sensor fusion can also be used to upgrade the performance of the filtering. These methods are more computationally expensive and may lead to other limitations in the system, but they are worth exploring to see if they offer improvements over the current system.

Similarly to the adaptive filter methods, there are many ways of performing multilateration predictions. The method used in this thesis is LLS which is efficient and quick but may not be the most accurate (as we used other methods to mitigate prediction errors as well). As mentioned earlier in the paper this can be extended to weighted least squares or non-linear least squares. While these approaches weren't deemed necessary for this basic prototype they may, if implemented, be able to improve performance.

Although optimizing the movement functionality was not in the main objectives of this thesis, it was developed in a reliable and time-efficient way by using a second Arduino Mini to control the motors. The choice of using a different Arduino for this task was made in order to not disturb the ranging program flow running on the Arduino M0. There is a possibility of implementing a real-time operating system (RTOS) in the Arduino M0 that can schedule the ranging and movement tasks in an efficient way, without disturbing each other. However, as the ranging is highly dependent on time measurements and processing delays, a great amount of caution is needed in order to maintain a high accuracy.

The movement of the robot can also be improved by implementing an obstacle detection functionality. This will improve the reliability of the robot while following an object in complex environments as it can autonomously navigate through obstacles while still keeping track and following the target. An obstacle detection functionality can also be used to detect NLOS situations and go into a fail safe mode if necessary.

Depending on the final application it may be necessary to increase the number of targets or tags being tracked. This can be done by adding identifiers to each target and allowing the tracking system to independently poll each target. While this will most likely reduce overall data rate, there may be methods to achieve appropriate functionality for such an application.

9

Bibliography

[1] A. V. Gulalkari *et al.*, "Object tracking and following six-legged robot system using kinect camera based on kalman filter and backstepping controller," *Journal of Mechanical Science and Technology*, vol. 29, no. 12, pp. 5425–5436, 2015.

[2] Y. Isobe, G. Masuyama, and K. Umeda, "Target tracking for a mobile robot with a stereo camera considering illumination changes," in *2015 IEEE/SICE International Symposium on System Integration (SII)*. IEEE, 2015, pp. 702–707.

[3] M. Gupta *et al.*, "A novel vision-based tracking algorithm for a human-following mobile robot," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. PP, no. 99, pp. 1–13, 2016.

[4] W. Ye *et al.*, "Vision-based human tracking control of a wheeled inverted pendulum robot," *IEEE Transactions on Cybernetics*, vol. 46, no. 11, pp. 2423–2434, 2015.

[5] H. Xiao *et al.*, "RGB-D sensor-based visual target detection and tracking for an intelligent wheelchair robot in indoors environments," *International Journal of Control, Automation and Systems*, vol. 13, no. 3, pp. 521–529, 2015.

[6] Q. Ren *et al.*, "Real-time target tracking system for person-following robot," in *2016 35th Chinese Control Conference (CCC)*. TCCT, 2016, pp. 6160–6165.

[7] A. Leigh *et al.*, "Person tracking and following with 2D laser scanners," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 726–733.

[8] N. Hirose, R. Tajima, and K. Sukigara, "Personal robot assisting transportation to support active human life—human-following method based on model predictive control for adjacency without collision," in *2015 IEEE International Conference on Mechatronics (ICM)*. IEEE, 2015, pp. 76–81.

[9] A. Wasik *et al.*, "Lidar-based relative position estimation and tracking for multi-robot systems," in *Robot 2015: Second Iberian Robotics Conference*. Springer, 2016, pp. 3–16.

[10] K. Misu and J. Miura, "Specific person detection and tracking by a mobile robot using 3d lidar and espar antenna," in *Intelligent Autonomous Systems 13*. Springer, 2016, pp. 705–719.

[11] J. Han and T. Jin, "Sound source based mobile robot control for human following in a networked intelligent space," *International Journal of Control and Automation*, vol. 8, no. 7, pp. 67–74, 2015.

[12] A. Jiménez *et al.*, "A computerized system to determine the provenance of finds in archaeological sites using acoustic signals," *Journal of Archaeological Science*, vol. 36, no. 10, pp. 2415–2426, 2009.

[13] H. Schweinzer and M. Syafrudin, "Losnus: An ultrasonic system enabling high accuracy and secure tdoa locating of numerous devices," in *Indoor Positioning and Indoor Navigation (IPIN), 2010 International Conference on*. IEEE, 2010, pp. 1–8.

[14] C. Medina, J. C. Segura, and A. De la Torre, "Ultrasound indoor positioning system based on a low-power ss sensor network providing sub-centimeter accuracy," *Sensors*, vol. 13, no. 3, pp. 3501–3526, 2013.

[15] M. R. Mahfouz *et al.*, "Recent trends and advances in UWB positioning," in *IEEE MTT-S Int. Microwave Workshop on Wireless Sensing, Local Positioning, and RFID*. IEEE, 2009, pp. 1–4.

[16] H. Liu *et al.*, "Survey of wireless indoor positioning techniques and systems," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 37, no. 6, pp. 1067–1080, Nov. 2007.

[17] J. D. Choi and W. E. Stark, "Performance of ultra-wideband communications with suboptimal receivers in multipath channels," *IEEE Journal on selected areas in communications*, vol. 20, no. 9, pp. 1754–1766, 2002.

[18] B. Hepp, T. Nägeli, and O. Hilliges, "Omni-directional person tracking on a flying robot using occlusion-robust ultra-wideband signals," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 189–194.

[19] P. K. Enge, "The global positioning system: Signals, measurements, and performance," *International Journal of ss Information Networks*, vol. 1, no. 2, pp. 83–105, 1994.

[20] U. S. Airforce. (2017) GPS accuracy. [Online]. Available: http://www.gps.gov/systems/gps/performance/accuracy/

[21] R. Mautz, "Overview of current indoor positioning systems," *GEODESY AND CARTOGRAPHY*, vol. 35, no. 1, pp. 18–22, 2009.

[22] A. Wasik *et al.*, "Graph-based distributed control for adaptive multi-robot patrolling through local formation transformation," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 1721–1728.

[23] R. Mautz, *Indoor positioning technologies*. ETH Zurich, Department of Civil, Environmental and Geomatic Engineering, Institute of Geodesy and Photogrammetry Zurich, 2012.

[24] J. Zhang *et al.*, "UWB systems for wireless sensor networks," *IEEE Proceedings*, vol. 97, no. 2, pp. 313–331, Feb. 2009.

[25] M. Z. Win *et al.*, "History and applications of UWB," *Proc. IEEE*, vol. 97, no. 2, pp. 198–204, 2009.

[26] S. Ingram, D. Harmer, and M. Quinlan, "Ultrawideband indoor positioning systems and their use in emergencies," in *2004 Position Location and Navigation Symposium (PLANS)*. IEEE, 2004, pp. 706–715.

[27] R. J. Fontana, "Recent system applications of short-pulse ultra-wideband (UWB) technology," *IEEE Transactions on microwave theory and techniques*, vol. 52, no. 9, pp. 2087–2104, 2004.

[28] T. Gigl *et al.*, "Analysis of a UWB indoor positioning system based on received signal strength," in *Positioning, Navigation and Communication, 2007. WPNC'07. 4th Workshop on*. IEEE, 2007, pp. 97–101.

[29] Y. Chu and A. Ganz, "A UWB-based 3d location system for indoor environments," in *2005 2nd International Conference on Broadband Networks (BroadNets)*. IEEE, 2005, pp. 1147–1155.

[30] M. Z. Win and R. A. Scholtz, "Ultra-wide bandwidth time-hopping spread-spectrum impulse radio for wireless multiple-access communications," *IEEE Transactions on communications*, vol. 48, no. 4, pp. 679–689, 2000.

[31] ——, "Characterization of ultra-wide bandwidth wireless indoor channels: A communication-theoretic view," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 9, pp. 1613–1627, 2002.

[32] P. Tomé *et al.*, "UWB-based local positioning system: From a small-scale experimental platform to a large-scale deployable system," in *2010 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE, 2010, pp. 1–10.

[33] B. Waldmann *et al.*, "An ultra-wideband local positioning system for highly complex indoor envi-

ronments," in *Localization and GNSS (ICL-GNSS), 2012 International Conference on*. IEEE, 2012, pp. 1–5.

[34] H. Soganci, S. Gezici, and H. V. Poor, "Accurate positioning in ultra-wideband systems," *IEEE Wireless Communications*, vol. 18, no. 2, 2011.

[35] M. R. Mahfouz *et al.*, "Integration of UWB and wireless pressure mapping in surgical navigation," *IEEE Transactions on Microwave Theory and Techniques*, vol. 57, no. 10, pp. 2550–2564, 2009.

[36] ——, "Towards sub-millimeter accuracy in UWB positioning for indoor medical environments," in *Biomedical Wireless Technologies, Networks, and Sensing Systems (BioWireleSS), 2011 IEEE Topical Conference on*. IEEE, 2011, pp. 83–86.

[37] J. Tiemann *et al.*, "Ultra-wideband aided precision parking for wireless power transfer to electric vehicles in real life scenarios," in *2016 IEEE 84th Vehicular Technology Conference (VTC-Fall)*. IEEE, 2016, pp. 1–5.

[38] M. Segura *et al.*, "Experimental demonstration of self-localized ultra wideband indoor mobile robot navigation system," in *2010 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE, 2010, pp. 1–9.

[39] M. Kok, J. D. Hol, and T. B. Schön, "Indoor positioning using ultrawideband and inertial measurements," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 4, pp. 1293–1303, 2015.

[40] Ubisense. (2017). [Online]. Available: https://ubisense.net

[41] Zebra. (2017) Dart ultra wideband (UWB) technology. [Online]. Available: https://www.zebra.com/gb/en/solutions/location-solutions/enabling-technologies/dart-uwb.html

[42] Decawave. (2017). [Online]. Available: http://www.decawave.com

[43] Bespoon. (2017). [Online]. Available: http://bespoon.com

[44] A. Alarifi *et al.*, "Ultra wideband indoor positioning technologies: Analysis and recent advances," *Sensors*, vol. 16, no. 5, p. 707, 2016.

[45] K. C. Cheok *et al.*, "UWB tracking of mobile robots," in *2010 IEEE 21st International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*. IEEE, 2010, pp. 2615–2620.

[46] D. Porcino and W. Hirt, "Ultra-wideband radio technology: potential and challenges ahead," *IEEE communications magazine*, vol. 41, no. 7, pp. 66–74, 2003.

[47] G. Breed, "A summary of fcc rules for ultra wideband communications," 2005.

[48] W. P. Siriwongpairat and K. R. Liu, *Ultra-Wideband Communications Systems: Multiband OFDM Approach*. Hoboken, NJ, USA: John Wiley & Sons, 2007.

[49] L. S. Committee *et al.*, "IEEE std 802.15. 4tm-2003," *IEEE Standard for Information technology Telecommunications and information exchange between systems Local and metropolitan area networks Specific requirements Part*, vol. 15, 2003.

[50] G. Shor. (2008) How Bluetooth, UWB, and 802.11 stack up on power consumption. [Online]. Available: http://www.eetimes.com/document.asp?doc_id=1273546

[51] Z. Sahinoglu, S. Gezici, and I. Guvenc, *Ultra-wideband positioning systems*. New York, NY, USA: Cambridge University Press, 2008.

[52] S. Gezici, "A survey on wireless position estimation," *Wireless personal communications*, vol. 44, no. 3, pp. 263–282, 2008.

[53] S. Goswami, *Indoor location technologies*. New York, NY, USA: Springer, 2013.

[54] *IEEE Standard for Local and metropolitan area networks— Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)*, IEEE Std. 802.15.4-2011, 2011.

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

[56] F. Santos and I. Tecnico, "Localization in ss sensor networks," *ACM Journal Name*, vol. 5, pp. 1–19, 2008.

[57] P. Brida and J. Machaj, "A novel enhanced positioning trilateration algorithm implemented for medical implant in-body localization," *International Journal of Antennas and Propagation*, vol. 2013, 2013.

[58] Pozyx Accurate Positioning. (2015) How does positioning work. [Online]. Available: https://www.pozyx.io/Documentation/how_does_positioning_work

[59] Penn State University. (2017) Regression methods. [Online]. Available: https://onlinecourses.science.psu.edu/stat501/

[60] Engineering Statistic Handbook. (2013) Nonlinear least squares regression. [Online]. Available: http://www.itl.nist.gov/div898/handbook/pmd/section1/pmd142.htm

[61] W. Murphy and W. Hereman, "Determination of a position in three dimensions using trilateration and approximate distances," *Department of Mathematical and Computer Sciences, Colorado School of Mines, Golden, Colorado, MCS-95*, vol. 7, p. 19, 1995.

[62] V. Pierlot, M. Urbin-Choffray, and M. Van Droogenbroeck, "A new three object triangulation algorithm based on the power center of three circles," in *International Conference on Research and Education in Robotics*. Springer, 2011, pp. 248–262.

[63] G. Welch and G. Bishop, "An introduction to the kalman filter," 1995.

[64] Arduino. (2017) Arduino M0 Pro. Arduino. [Online]. Available: http://www.arduino.org/products/boards/arduino-m0-pro

[65] Thotro. (2017) Arduino-dw1000. [Online]. Available: https://github.com/thotro/arduino-dw1000

[66] Decawave, "IEEE 802.15.4 2011 UWB Transceiver Module," DWM1000, Datasheet, 2015. [Online]. Available: http://www.decawave.com/sites/default/files/resources/dwm1000-datasheet-v1.3.pdf

[67] Arduino. (2017) Arduino Pro Mini. Arduino. [Online]. Available: https://www.arduino.cc/en/Main/ArduinoBoardProMini

[68] ST, "Dual full-bridge driver," L298, Datasheet, 2000. [Online]. Available: https://www.sparkfun.com/datasheets/Robotics/L298_H_Bridge.pdf

[69] Olimex. (2017) BB-L298. Olimex. [Online]. Available: https://www.olimex.com/Products/RobotParts/MotorDrivers/BB-L298/open-source-hardware

Appendices

A

Circuit Schematics

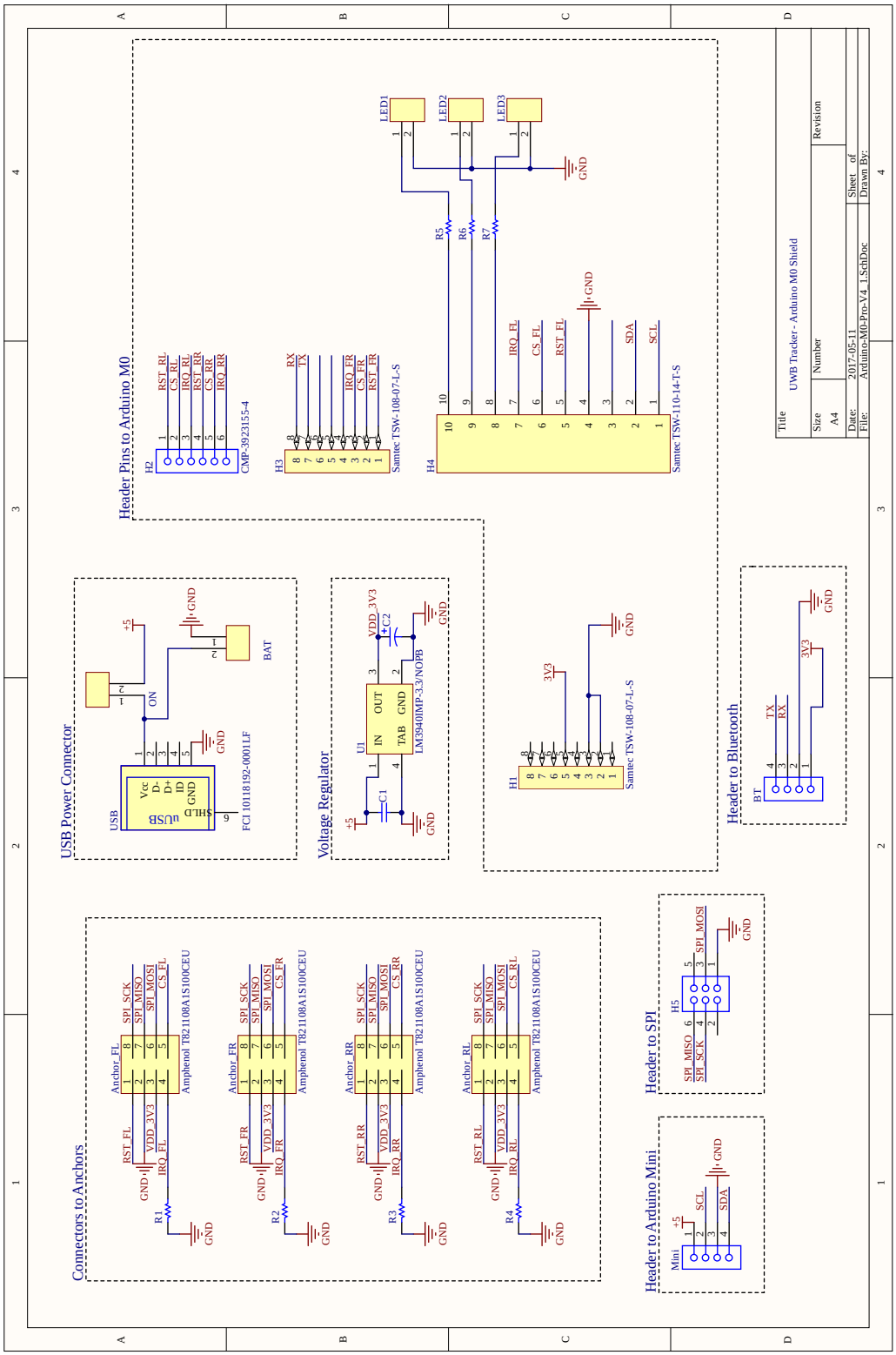

Fig. A.1. Arduino M0 shield schematics

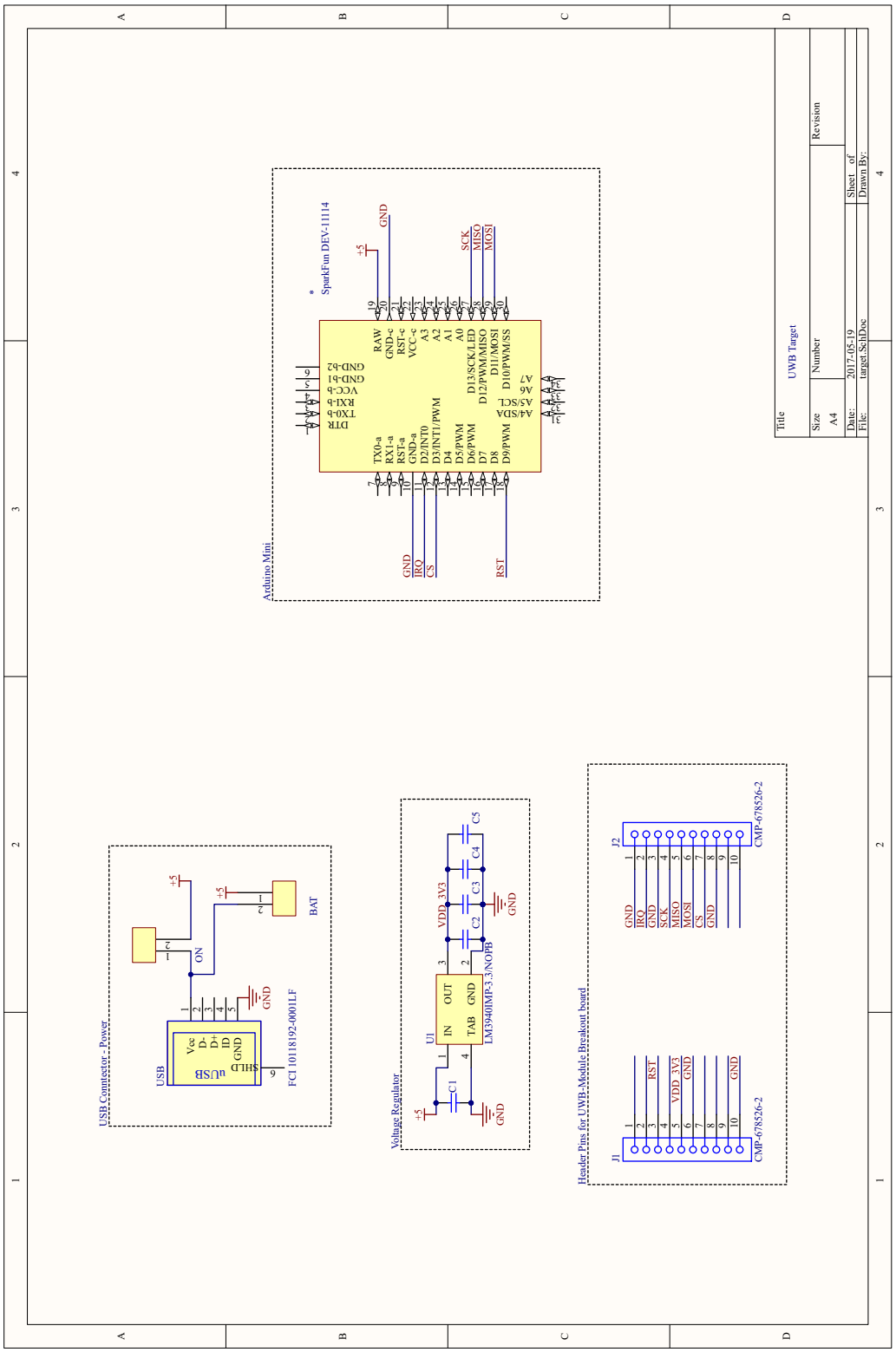

Fig. A.2. Target module schematics