Tampere University

Antti-Jussi Mäkipää

UWB INDOOR POSITIONING WITH KALMAN FILTERS

Master's Thesis
Faculty of Information Technology and Communication Sciences
Examiners: Lauri Suomela
Antti Larjo
Joni Kämäräinen
November 2023

ABSTRACT

Antti-Jussi Mäkipää: UWB Indoor Positioning With Kalman Filters
Master's Thesis
Tampere University
Signal processing and Machine Learning
November 2023

This thesis explores the application of Kalman Filters to Ultra-Wideband (UWB) indoor positioning systems (IPS), addressed through two primary research questions: "How can Kalman Filters be used in UWB indoor positioning?" and "How do different Kalman Filter implementations perform in the specific indoor positioning setup of this study?". The first question is explored through an extensive literature review, while the latter is addressed via practical experimentation in Veracell's office environment.

The literature review is bifurcated, focusing on the theoretical underpinnings of UWB and the practical application of Kalman Filters for indoor positioning. Insights into the characteristics of UWB, various UWB ranging methods, the foundational principles of Kalman Filters, and two specific types - Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF) - are elaborated. This review serves as a precursor to the empirical component of the thesis, aiming to identify and evaluate an alternative positioning methodology against Veracell's existing system.

The second research question is addressed through the collection of two distinct datasets from Veracell's indoor positioning environment and the implementation of EKF and UKF algorithms for positioning analysis. Experimentation results suggest that the integration of Kalman Filters enhances the accuracy of the UWB indoor positioning system.

While these findings are promising, additional research and empirical testing are requisite to validate the adaptability and efficacy of Kalman Filters across diverse indoor positioning contexts, ensuring their optimized performance in Veracell's and analogous IPS implementations.

Keywords: UWB, indoor positioning, Kalman Filters

TIIVISTELMÄ

Antti-Jussi Mäkipää: UWB sisätilapaikannus Kalman-suotimilla
Diplomityö
Tampereen yliopisto
Signaalinkäsittely ja koneoppiminen
Marraskuu 2023

Tässä diplomityössä tutkitaan Kalman-suotimien soveltamista Ultra-Wideband (UWB) sisätilapaikannusjärjestelmiin. Tutkimus voidaan jakaa kahteen tutkimuskysymykseen: "Miten Kalman-suotimia voidaan käyttää UWB-sisätilapaikannuksessa?"ja "Miten eri Kalman-suodinten toteutukset suoriutuvat tämän tutkimuksen erityisessä sisätilapaikannusjärjestelmässä?". Ensimmäiseen kysymykseen haetaan vastausta laajan kirjallisuuskatsauksen kautta, kun taas jälkimmäiseen kysymykseen vastataan käytännön kokeilujen avulla Veracellin toimistoympäristössä.

Kirjallisuuskatsaus jakautuu kahtia keskittyen UWB:n teoreettisiin perusteisiin ja Kalman-suotimien käytännön sovelluksiin sisätilapaikannuksessa. Kirjallisuusatsaus tarjoaa tietoa UWB:n ominaisuuksista, erilaisista UWB-etäisyysmittausmenetelmistä, Kalman-suotimien perusperiaatteista sekä kahdesta erityyppisestä Kalman-suotimesta - laajennetusta Kalman-suotimesta (EKF) ja epälineaarisesta Kalman-suotimesta (UKF). Tämä katsaus toimii pohjana diplomityön empiiriselle osuudelle, jonka tavoitteena on tunnistaa ja arvioida vaihtoehtoista paikannusmenetelmää Veracellin nykyiseen järjestelmään verrattuna.

Toinen tutkimuskysymys käsitellään keräämällä kaksi erillistä datajoukkoa Veracellin sisätilojen paikannusympäristöstä ja toteuttamalla EKF- ja UKF-algoritmeja paikannusanalyysiin. Kokeilujen tulokset viittaavat siihen, että Kalman-suotimien integrointi lisää UWB-sisätilojen paikannusjärjestelmän tarkkuutta.

Vaikka nämä löydökset ovat lupaavia, tarvitaan lisätutkimuksia ja empiirisiä testejä Kalman-suotimien soveltuvuuden ja tehokkuuden vahvistamiseksi monipuolisissa sisätilojen paikannusympäristöissä. Tämä varmistaa niiden optimoidun suorituskyvyn Veracellin ja vastaavien IPS-toteutusten yhteydessä.

Avainsanat: UWB, sisätilapaikannus, Kalman-suodin

PREFACE

First of all, I would like to thank Veracell Oy, for providing me this interesting thesis topic and allowed me to finalize my studies while working in the company. A huge thanks goes to my supervisor Lauri Suomela, whose continuous support and advice made the writing process less tedious. I would also like to thank Antti Larjo, who supported me with the technical aspects and also agreed to be one of the supervisors. Special mention goes to Philipp Müller, who helped me with Extended Kalman Filter (EKF) implementation. Also, thanks to Joni Kämäräinen for being one of the examiners.

Completing this thesis was the final step towards my graduation. The writing process even had a 6-month break when I was doing an exchange period in Taiwan. In the end, I would like to thank my friends and family for pushing me to finalize this project, even after the long break.

Tampere, 9th November 2023

Antti-Jussi Mäkipää

CONTENTS

1. Introduction 1
2. Background 3
2.1 UWB Characteristics and Definition 3
2.2 UWB Technology 3
2.3 UWB Positioning Parameters 4
2.3.1 Received signal strength 4
2.3.2 Angle of Arrival 5
2.3.3 Time Difference of Arrival 7
2.3.4 Time of Arrival 9
2.3.5 Time of Flight 9
2.4 Signal Propagation 11
2.5 Location Estimation 13
2.5.1 Optimization Method 13
2.5.2 Kalman Filters 14
2.5.3 Linear Kalman Filter 15
2.5.4 Extended Kalman Filter 17
2.5.5 Unscented Kalman Filter 18
3. Materials and methodology 20
3.1 Devices 20
3.2 Indoor Positioning Setup 22
3.3 Experiments 24
3.3.1 LOS and NLOS standard deviations 25
3.3.2 Data Gathering 25
3.3.3 Data Preparation 27
3.3.4 Kalman Filter Parameters 28
3.4 Implemented Software 30
3.5 Statistical Methods 31
4. Results 32
5. Discussion 36
6. Conclusion 39
References 40

ABBREVIATIONS AND MARKINGS

μ	Mean
σ	Standard deviation
AOA	Angle of arrival
BFGS	Broyden–Fletcher –Goldfarb–Shanno
BLE	low-energy Bluetooth
BW	Bandwidth
dB	Decibel
dBm	Decibel-milliwatts
EKF	Extended Kalman Filter
FCC	Federal Communication Commission
GNSS	Global Navigation Satellite System
IPS	Indoor Positioning System
IR	Impulse Radio
KDE	Kernel density estimation
LOS	Line-of-sight
MB-OFDM	Multi-band Orthogonal Frequency Division Multiplexing
NLOS	Non-line-of-sight
NLS	Non-linear Least Squares
RSS	Received signal strength
SQL	Structured query language
TDOA	Time difference of arrival
TOA	Time of arrival
TOF	Time of flight
UKF	Unscented Kalman Filter
UWB	Ultra-wideband

1. INTRODUCTION

As life expectancy grows around the world, simultaneously the incidence of dementia increases [1]. This causes challenges for the healthcare sector, as more nursing resources are needed. Studies have indicated that Indoor Positioning System (IPS) holds potential in healthcare, especially in environments with elderly patients [2][3]. For instance, IPS could reduce the number of home visits by keeping track of the elderly person. Additionally, the IPS could send alarms if the patient has left the building.

Global Navigation Satellite System (GNSS) has been utilized in outdoor positioning for years. However, GNSS is not the optimal solution for IPS because it lacks precision in indoor environments [4]. This led to other technologies arriving in the sector, for example, Ultra-wideband (UWB) and low-energy Bluetooth (BLE) [5][6]. These cost-effective technologies can provide enough accurate ranging results and still have sufficiently low power consumption for the devices to be battery-powered.

It is typical for indoor environments to have a lot of error sources (e.g. obstacles and dead spaces), which affect positioning accuracy. Kalman Filters can be utilized in these kinds of scenarios to improve positioning accuracy. Kalman Filters have been widely used for instance in vehicle velocity estimation [7] but also in positioning [8][9].

Utilizing Kalman Filters in indoor environments for example in nursing homes could lead to more precise UWB positioning. This could provide better consistency when the system wants to send alarms when a patient with dementia has left the building.

This thesis aims to provide information on UWB positioning in indoor environments and the utilization of different Kalman Filter implementations. The data is collected from a real office setup and the tests are implemented on the data. The main purpose of the thesis is to answer the following research questions.

- *RQ 1. How Kalman Filters can be used in UWB indoor positioning?*

The first research question is answered via a literature review of existing solutions for UWB positioning. However, the setup and the environment in these cases are different than in this thesis. This leads to the second research question.

- *RQ 2. How do different Kalman Filter implementations perform on the indoor positioning setup of the thesis?*

The second research question can be answered after indoor location data has been collected and the different Kalman Filter implementations have been tested.

The remainder of the thesis is structured in the following manner. Chapter 2 contains the literature review and information on the theoretical background. Chapter 3 provides information on the devices used in this thesis, data collection, implemented software, and statistical methods. Chapter 4 presents the results of the experiments. In chapter 5, the results are discussed. Ultimately, chapter 6 presents the conclusion. The bibliography is presented at the very end of the thesis.

2. BACKGROUND

This chapter will go through UWB characteristics and UWB positioning parameters, the importance of signal propagation in UWB and finally positioning with optimization methods and Kalman Filters.

2.1 UWB Characteristics and Definition

UWB is a wireless communication technology known since the 1900s. UWB technique provides fine time resolution which makes it suitable for ranging which requires accuracy. And due to the wide bandwidth, UWB signals provide decent material penetration.

The Federal Communication Commission (FCC) from the US proposed that any signal which has a bandwidth (*BW*) over 500 MHz or has a fractional bandwidth (B_f) larger than 0.20, is defined as an UWB signal [10]. Fractional bandwidth can be calculated with the following formula

$$B_f = \frac{BW}{f_C} = \frac{2(f_H - f_L)}{f_H + f_L}, \tag{2.1}$$

where f_C is center frequency of the UWB transmission, f_H and f_L are upper and lower frequencies of the -10 dB emission points [11]. UWB transmission center frequency is determined as the upper and lower -10 dB average, which is

$$f_C = \frac{f_H + f_L}{2}. \tag{2.2}$$

There are also many other regulations regarding UWB dependent on the region, which are listed on the Table 2.1

2.2 UWB Technology

Typically, one of the following technologies is used when utilizing UWB wireless communications: either Multi-band Orthogonal Frequency Division Multiplexing (MB-OFDM) or Impulse Radio (IR). Both of these technologies offer low power consumption and low-complexity radio technology. [12]

Table 2.1. *UWB regulations from different countries and regions. Adapted from [12].*

Regulations	f_L (GHz)	f_H (GHz)	f_C (GHz)	B_f	Emission Level (dBM/MHz)
FCC	3.1	10.6	6.85	109.5%	-41.3
ECC	3.1	4.8	3.95	43.0%	-41.3
	6.0	8.5	7.25	34.5%	-41.3
Japan	3.4	4.8	4.1	34.1%	-41.3
	7.24	10.25	8.75	34.3%	-41.3
Korea	1.0	10.0	5.5	163.6%	-66.5
Singapore	2.2	10.6	6.4	131.25%	-35.0

- **IR** UWB technology uses very short duration pulses in the transmission. These narrow pulses also provide fine time resolution, which makes Impulse Radio considerable option for accurate positioning. Furthermore, because of the very narrow pulses in the spectrum, this technology can reach up to several gigahertz of bandwidth. Impulse Radio is carrier-less and thus it doesn't require a mixer.
- **MB-OFDM** uses OFDM technology where the spectrum is divided to multiple subband where the bandwidth is at least 500 MHz. The information is passed among the sub-bands and ultimately transmitted via multi-carrier technique. In contrast to Impulse Radio, MB-OFDM requires carrier waves.

The devices used in this thesis are packed with a DW1000 (Qorvo Inc, North-Carolina, United States) UWB transceivers which utilize the Impulse Radio technology [13]. More precise explanation of the devices can be found in Chapter 3.

2.3 UWB Positioning Parameters

In order to estimate position of the target, it requires additional information, such as the distance or the angle between the devices in the UWB network. These types of parameters can be extracted from the UWB signals. Typically either one or more of the following parameters are used: signal strength, angle of arrival (AOA) or time measurements (time difference of arrival (TDOA), time of arrival (TOA) or time of flight (TOF)) [14]. In this thesis, TOF is used to determine the distances between the devices. The following sections will go through the techniques described above.

2.3.1 Received signal strength

The most straight-forward approach for positioning is received signal strength based technique, because estimating signal strength rarely requires notable modifications to the sys-

tems already existing. The method consists of estimating the received signal strength and also knowing the transmitted signal strength and the signal propagation model. Ideally the received signal strength (RSS) can be denoted as [15]

$$\bar{P}(d) = P_0 - 10n\log_{10}(\frac{d}{d_0}), \tag{2.3}$$

where $\bar{P}(d)$ is the average received signal power (dB) at a distance d, n is the path loss exponent and P_0 is the received signal power (dB) at a given reference distance d_0. Typically, the received signal power $P(d)$ can be modeled as a Gaussian random variable with variance σ^2 and mean $\bar{P}(d)$, in other words

$$10\log_{10}P(d) \sim \mathcal{N}\left(\bar{P}(d),\, \sigma^2\right). \tag{2.4}$$

From the Equation 2.4, distance estimation can be expressed with Cramer-Rao lower bound (CRLB) [16] as

$$\sqrt{\mathrm{Var}\{\hat{d}\}} \geq \frac{\ln 10}{10}\frac{\sigma}{n}d, \tag{2.5}$$

where $\hat{d}$ is the unbiased estimate of d. While ranging with RSS is fairly easy to implement, it is less accurate than the other techniques described below. [17]

2.3.2 Angle of Arrival

While RSS estimates the range between the nodes, AOA estimates the angle of the arriving signal between the nodes. Typically antenna arrays are used to measure AOA of the signal. Figure 2.2 illustrates an antenna array. As the signal arrives to the antenna array, the different antenna elements measure the differences of the signal arrival times and the angle information can be estimated. [18]

Let's consider N_a antenna array elements and denote the received signal at the *i*th antenna element as $r_i(t)$, which can be expressed in a following manner:

$$r_i(t) = \alpha s(t - \tau_i) + n_i(t), \quad i = 1, ..., N_a, \tag{2.6}$$

where α is the channel coefficient, $s(t)$ the transmitted signal, τ_i the delay of the signal arriving at ith antenna element, and finally $n_i(t)$ is zero mean white Gaussian noise with $\mathcal{N}_0/2$ spectral density.

The delay τ can be estimated as

$$\tau_i \approx \frac{d}{c} + \frac{l_i \sin\psi}{c}, \tag{2.7}$$

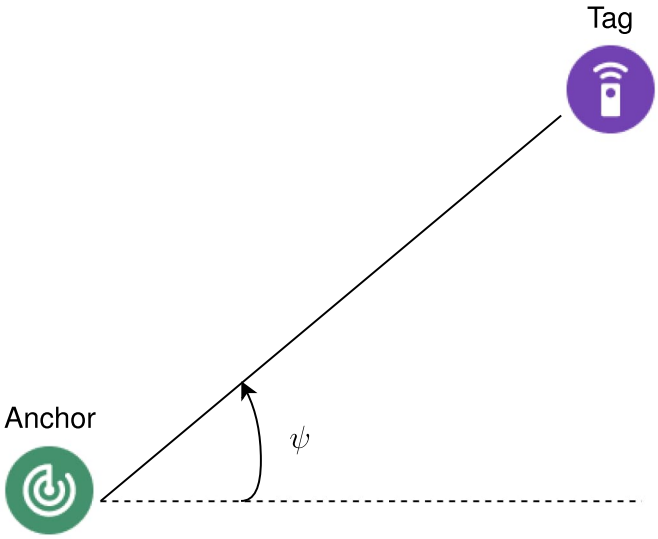

Figure 2.1. *Angle of arrival between two nodes.*

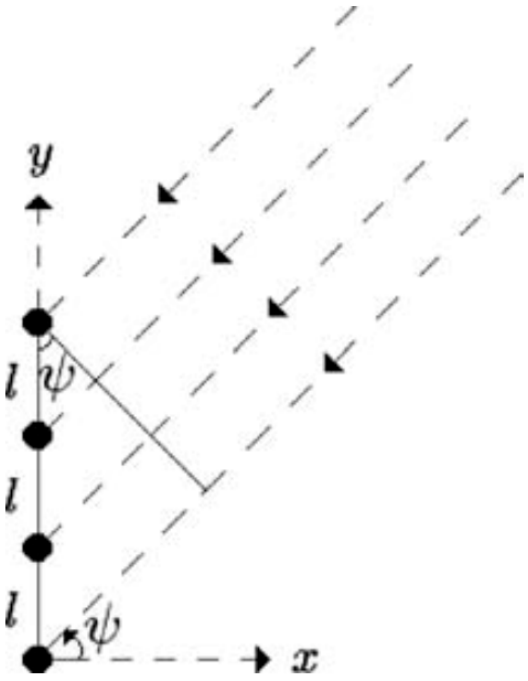

Figure 2.2. *Illustration of an antenna array. In this figure, l is inter element spacing, which is the distance between antenna elements, and ψ is the signal's angle of arrival. Adapted from [18].*

where d is the distance between the receiver's center of the antenna array and transmitter. In the equation above, l_i refers to

$$l_i = l(\frac{N_a + 1}{2} - i), \tag{2.8}$$

in which l is the inter-element spacing.

With an assumption of independent noise at separate antenna elements, the angle of arrival ψ can be estimated with the CRLB of the estimate being [19]

$$\mathrm{Var}\{\hat{\psi}\} \geq \frac{6c^2\mathcal{N}_0}{\alpha^2\tilde{E}N_a(N_a^2-1)l^2\cos^2\psi}, \tag{2.9}$$

where $\tilde{E}$ is the energy of transmitted signal's first derivative $s'(t)$, formulated as

$$\tilde{E} = \int_{\infty}^{\infty}[s'(t)]^2\,dt. \tag{2.10}$$

The DW1000 modules do not support the retrieval of AOA information, but with the upgraded version DW3000 [20] it is possible to do positioning with AOA information.

2.3.3 Time Difference of Arrival

Time difference of arrival (TDOA) is technique where multiple fixed nodes transmit a signal at the same time to a mobile node (downlink) or the mobile node transmits a message to the fixed nodes (uplink), and then the timestamp of the arrived signals can be compared [14]. TDOA can provide accurate results but the downside of the method is that the nodes clocks must be synchronized and this can be quite complex in many practical setups. TDOA requires at least three fixed nodes to estimate the location of the mobile node. Figure 2.3 represents a case where a mobile node (tag) sends a blink message to multiple fixed nodes (anchors) and the received timestamp is recorded.

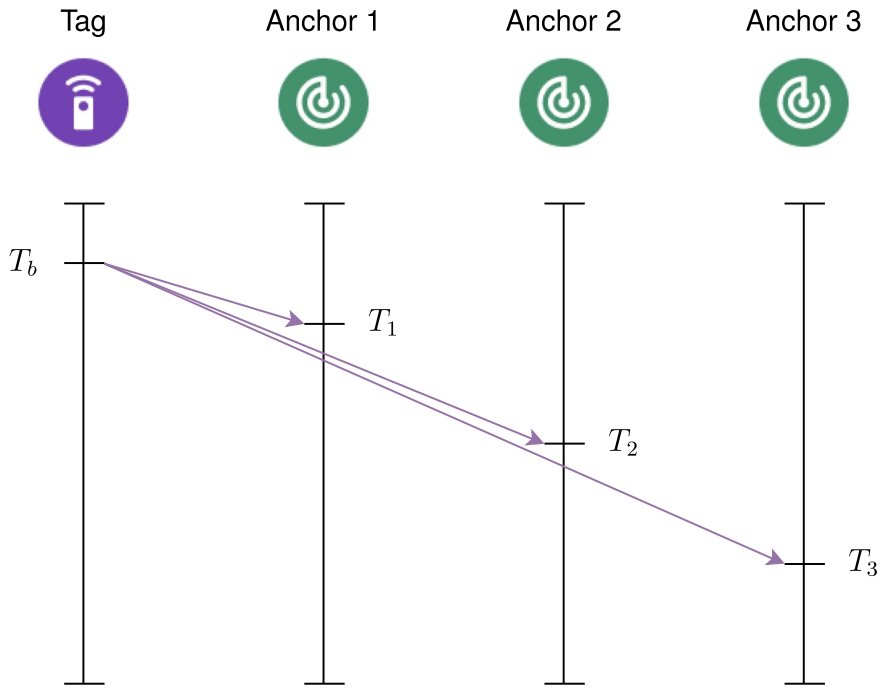

Figure 2.3. *Simple example of TDOA application.*

Presuming that the tag sends the signal at time T_b, the time that this signal arrived is

related to the distance between the tag and the anchors. Let $T_{TOF,i}$ be the signal time of flight between the tag and the *i*th anchor, c constant speed of light, d_i the distance between the tag and the *i*th anchor, (x_i, y_i) position of the *i*th anchor in two dimensions and T_i is the time of arrival to the *i*th anchor. Now we can describe the relationship between measured T_i and position of the tag with the following equations:

$$T_i = T_{TOF,i} + T_b \tag{2.11}$$
$$= \frac{d_i}{c} + T_b \tag{2.12}$$
$$= \frac{\sqrt{(x_i - x)^2 + (y_i - y)^2}}{c} + T_b. \tag{2.13}$$

We still have three unknowns in the equation, but we can get rid of the T_b by subtracting the time of arrivals to anchors. Now, we can formulate the equation as such

$$T_i - T_j = T_{TOF,i} + T_b - (T_{TOF,j} - T_b) \tag{2.14}$$
$$= T_{TOF,i} - T_{TOF,j} \tag{2.15}$$
$$= \frac{d_i}{c} - \frac{d_j}{c} \tag{2.16}$$
$$= \frac{\sqrt{(x_i - x)^2 + (y_i - y)^2} - \sqrt{(x_j - x)^2 + (y_j - y)^2}}{c}. \tag{2.17}$$

There is still two unknowns in the equation, but when we accommodate a third anchor, and form another equation we get

$$T_i - T_j = \frac{\sqrt{(x_i - x)^2 + (y_i - y)^2} - \sqrt{(x_j - x)^2 + (y_j - y)^2}}{c} \tag{2.18}$$

and also

$$T_i - T_k = \frac{\sqrt{(x_i - x)^2 + (y_i - y)^2} - \sqrt{(x_k - x)^2 + (y_k - y)^2}}{c}. \tag{2.19}$$

At this point we have two unknowns and two nonlinear formulas.

TDOA is usually referred to hyperbolic positioning, and the reason is, when the equations above are being solved, the values (x, y) that meet the requirements for the equation, form hyperbolas when plotted. The possible solutions are the points where these hyperbolas intersect. However, it is not guaranteed that the hyperbolas will intersect.

2.3.4 Time of Arrival

Time of Arrival (TOA) is similar to TDOA, the difference being that we know the tag's timestamp T_b. We formulate the equation similar to TDOA:

$$T_i = T_{TOF,i} + T_b \tag{2.20}$$

$$T_i - T_b = \frac{d_i}{c} \tag{2.21}$$

$$= \frac{\sqrt{(x_i - x)^2 + (y_i - y)^2}}{c}. \tag{2.22}$$

As previously mentioned, we cannot solve the unknowns with this information, so we add another anchor, and it results in having two equations:

$$T_i - T_b = \frac{\sqrt{(x_i - x)^2 + (y_i - y)^2}}{c} \tag{2.23}$$

and also

$$T_j - T_b = \frac{\sqrt{(x_j - x)^2 + (y_j - y)^2}}{c}. \tag{2.24}$$

While the geometric interpolation of TDOA was that the equation formed a hyperbola, in the TOA, a single equation forms a circle. Again, the possible locations of the tag are where the circles intersect. Having two anchors in TOA can yield two possible solutions, but adding a third anchor (thus receiving a third equation) guarantees a single solution [17].

2.3.5 Time of Flight

As the two methods above require clock synchronization, alternative methods have to be considered. A popular Time of Flight (TOF) method is called Two-Way Ranging, which is usually Single-Sided or Double-Sided. While Two-Way ranging does not require clock synchronization, it increases air-time of the nodes and that increases the power consumption, which has to be taken into account in the cases where the nodes are battery powered. [21]

Single-Sided Two-Way Ranging (SS-TWR) protocol is demonstrated in Figure 2.4. The tag sends a message to the anchor, and the anchors sends a message back. From this information, the distance between the anchor and the tag can be calculated with the following formula

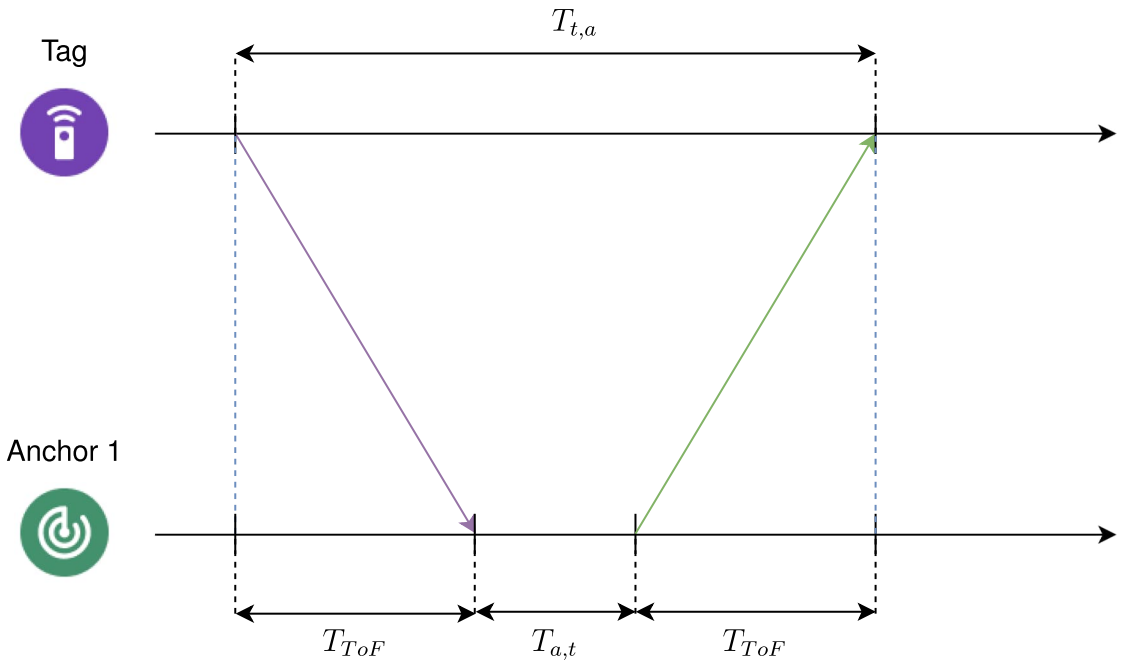

Figure 2.4. *Single-Sided Two-way Ranging.*

$$d = T_{TOF}\, c \tag{2.25}$$

$$= \frac{(T_{t,a} - T_{a,t})}{2}\, c. \tag{2.26}$$

Single-Sided method requires less messages than the Double-Sided version, but the accuracy falls when the clock offset increases. In order to use the SS-TWR, the timestamps have to be measured very accurately, which require precise oscillators [22].

Double-Sided Two-Way Ranging (DS-TWR) can be implemented with either three or four messages and here we consider the three message case because the distances in this thesis are acquired using three messages. Figure 2.5 illustrates the DS-TWR in a three message case.

From the Figure 2.5 can be seen that

$$T_{t,a} = 2T_{TOF} + T_{a,t} \tag{2.27}$$

$$T_{a,t'} = 2T_{TOF} + T_{t,a'}, \tag{2.28}$$

and multiplying $T_{t,a}$ and $T_{a,t'}$, results in

$$T_{t,a}T_{a,t'} = (2T_{TOF} + T_{a,t})(2T_{TOF} + T_{t,a'}). \tag{2.29}$$

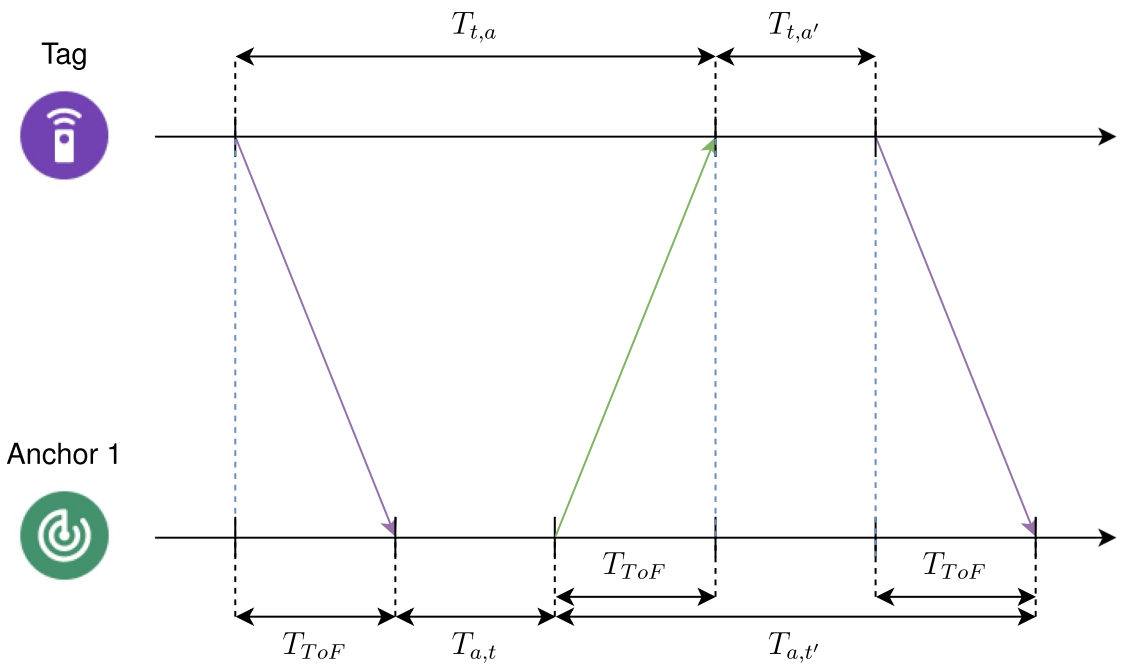

Figure 2.5. *Double-Sided Two-way Ranging.*

After a little expanding and rearranging we get:

$$T_{t,a}T_{a,t'} - T_{a,t}T_{t,a'} = 2T_{TOF}(2T_{TOF} + T_{a,t} + T_{t,a'}). \tag{2.30}$$

Then, using Equations 2.27 and 2.28 to replace terms to yield

$$T_{t,a}T_{a,t'} - T_{a,t}T_{t,a'} = 2T_{TOF}(T_{t,a} + T_{t,a'}) \tag{2.31}$$

$$T_{t,a}T_{a,t'} - T_{a,t}T_{t,a'} = 2T_{TOF}(T_{a,t'} + T_{a,t}). \tag{2.32}$$

Finally, we can solve the T_{TOF} from the equation pair:

$$T_{TOF} = \frac{T_{t,a} \cdot T_{a,t'} - T_{t,a'} \cdot T_{a,t}}{2(T_{a,t} + T_{a,t'})} \tag{2.33}$$

$$= \frac{T_{t,a} \cdot T_{a,t'} - T_{t,a'} \cdot T_{a,t}}{2(T_{t,a} + T_{t,a'})} \tag{2.34}$$

$$= \frac{T_{t,a} \cdot T_{a,t'} - T_{t,a'} \cdot T_{a,t}}{T_{a,t} + T_{a,t'} + T_{t,a} + T_{t,a'}}. \tag{2.35}$$

The derivation of this equation was originally proposed by Neirynck et al. [23].

2.4 Signal Propagation

As UWB ranging is very precise, the signal propagation plays a significant role in the measurement accuracy. The propagation can be divided to line-of-sight (LOS) propagation and non-line-of-sight (NLOS) propagation. LOS scenario is when the signal is able

to travel directly between the transceivers. NLOS scenario on the other hand has something blocking the signal between the transceivers, such as humans or objects. The signal doesn't get fully attenuated (ideally) by the blocking object in the NLOS scenario, but it reflects from other objects or walls. This immediately affects the ranging accuracy, as the signal takes more time to arrive to the other transceiver. In a paper released by Müller et al. [24], where they implemented an Extended Kalman Filter (EKF) and a Generalized Gaussian Mixture filter (GGMF) for UWB positioning, the group stated that separating LOS and NLOS measurements would provide better models for the EKF and GGMF. A similar approach is used in this thesis, and the LOS and NLOS measurements will be categorized. The rest of this section will go through how the measurements can be labeled as LOS or NLOS with the DW1000 modules.

Some UWB radios have built-in methods to identify whether the measurement is LOS or NLOS. The DW1000 does not have this, but it can be calculated with the values acquired in the ranging. DW1000 user manual [13] provides equations for calculating estimated Receive Signal power (RX power) and First Path power (FP power).

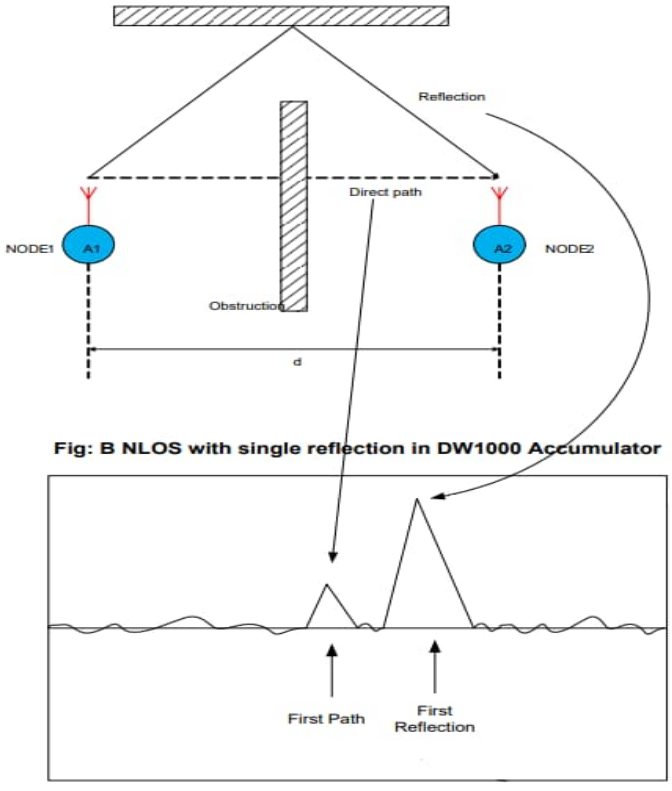

Figure 2.6. *Illustration of how objects and reflections affect the peak amplitudes. Adapted from Application Note 6 part 2 [25].*

From the Figure 2.6 we can see how the object blocking the path affects the height of the peaks. The amplitudes of the peaks can be compared, so let's denote the difference of the magnitudes as a:

$$a = |\text{RX power - FP power}| \text{ dB}, \tag{2.36}$$

and with the following equation the probability of LOS can be determined:

$$p_{LOS} = \begin{cases} 1 & a \leq 3.3 \\ 0.39178 \cdot a - 1.31719 & 3.3 < a < 6 \\ 0 & a \geq 6. \end{cases} \tag{2.37}$$

RX power and *FP power* are expressed in decibel-milliwatts (dBm). The threshold values and the multipliers are adapted from DW1000 Application Note 6 part 3 [25]. After the probability of LOS has been determined, some kind of threshold value needs to be chosen to label the measurements. Threshold value of 0.5 is used in this thesis, so the decision whether the measurement is LOS or NLOS can be expressed as:

$$LOS = \begin{cases} 1 & p_{LOS} > 0.5 \\ 0 & p_{LOS} \leq 0.5 \end{cases} \tag{2.38}$$

The chosen value might not be optimal and choosing the right threshold value is another task which is out of scope of this thesis.

2.5 Location Estimation

After the distances between tag and anchors have been calculated, we can estimate the position of the tag in two dimensional space with three anchors or in three dimensional space with four anchors respectively. This requires knowledge of the fixed anchors locations.

The position of the tag can be estimated with various techniques. One approach is to use non-linear optimization methods such as non-linear Least Squares (NLS) or a minimization method which is described below. The other method includes Kalman Filters. However, as the measurements from the devices are non-linear, linear Kalman Filter cannot be used by itself. Alternatively one can use non-linear Kalman Filter such as EKF or UKF.

2.5.1 Optimization Method

For this estimation problem, we used an optimization algorithm called Broyden–Fletcher–Goldfarb–Shanno (BFGS). BFGS is an iterative method for nonlinear unconstrained optimization problems and it uses curvature information to determine the gradient descent direction [26]. Typically optimization algorithms involve minimizing or maximizing a certain function [27]. In this case the function being minimized is a loss function, which can

be denoted as

$$f(\mathbf{x})_i = \sum_{i=0}^{n} \left| d_i - \sqrt{(x_{a,i} - x_t)^2 + (y_{a,i} - y_t)^2 + (z_{a,i} - z_t)^2} \right|, \tag{2.39}$$

where d_i is the estimated distance between a tag and an *i*th anchor, $(x_{a,i}, y_{a,i}, z_{a,i})$ is the known position of the *i*th anchor and finally $\mathbf{x} = (x_t, y_t, z_t)$ is the position of the tag.

In the BFGS algorithm, second derivatives are not needed for calculating the Hessian matrix. BFGS rather imposes similar conditions on the Hessian matrix inverses $\mathbf{H}_k$. There are few restrictions for BFGS. The updated approximation of $\mathbf{H}_{k+1}$ has to be symmetric and positive definite, and also requires to satisfy the following equation [28]:

$$\mathbf{H}_{k+1}\mathbf{y}_k = \mathbf{s}_k \tag{2.40}$$

where $\mathbf{y}_k$ and s_k are denoted as

$$\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x_k} = \alpha_k p_k, \quad \mathbf{y_k} = \nabla f_{k+1} - \nabla f_k. \tag{2.41}$$

After these conditions are met, the unique solution for $\mathbf{H}_{k+1}$ can be calculated with

$$\mathbf{H}_{k+1} = (I - \rho_k \mathbf{s}_k \mathbf{y}_k^T)\mathbf{H}_k(I - \rho_k \mathbf{y}_k \mathbf{s}_k^T) + \rho_k \mathbf{s}_k \mathbf{s}_k^T, \tag{2.42}$$

where ρ_k is denoted as

$$\rho_k = \frac{1}{\mathbf{y}_k^T \mathbf{s}_k}. \tag{2.43}$$

The algorithm begins with an initial guess of the tag's location $\mathbf{x}_0$ and appropriate Hessian matrix $\mathbf{H}_0$. $\mathbf{H}_0$ can be any definite positive matrix, however, typically $\mathbf{H}_0 = \mathbf{I}$ is used if no better estimate is available. After the initial values are set, the algorithm is iterated accordingly:

1. Calculate search direction $p_k = -\mathbf{H}_k \nabla f_k$
2. Set $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k p_k$, where α_k is the step size.
3. Calculate $\mathbf{H}_{k+1}$ with equation 2.42
4. Check if convergence criteria is met, if not, jump to step 1.

For finding suitable step size α_k, typically a Wolfe line search is used [29]. In the code, this BFGS method is implemented with a SciPy library [30].

2.5.2 Kalman Filters

Kalman filters are used to estimate a new state of a process, for example estimating a new location of a moving vehicle. Kalman filters can be divided into two main steps:

prediction step, where a guess is made based on previous information and the correction step, which uses a noisy measurement to correct the estimated step [31]. Figure 2.7 illustrates the basic process of a Kalman filter.

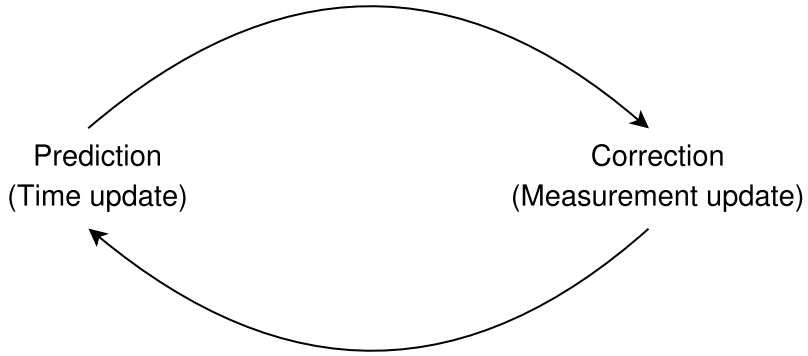

Figure 2.7. *Kalman filter process*

2.5.3 Linear Kalman Filter

Kalman filters describe the previously mentioned processes with linear models. This leads to the restrictions that the models are linear and the noise is assumed to be Gaussian.

The first model is called transition model and it describes the relationship between a new state and the previous state. Also in the transition model there are control commands and noise. This transition model can be formulated as

$$\mathbf{x_k} = \mathbf{F}\mathbf{x_{k-1}} + \mathbf{G}\mathbf{u_k} + \mathbf{w_k}, \tag{2.44}$$

where $\mathbf{x_k}$ describes the state matrix of the process in the *k*th time interval. For instance, the state matrix may include the position of the tracked object in two or three dimensions, which is exactly the case in this thesis. Matrix $\mathbf{F}$ represents the state transition matrix, which expresses the relationship of two consecutive states. Matrix $\mathbf{G}$ is called a control matrix which describes the effect of control commands (or control variables) $\mathbf{u_k}$ on the state. Lastly $\mathbf{w_k}$ is the process noise, which has a relation to $\mathbf{Q_k}$, which is the covariance matrix of the process noise. $\mathbf{Q_k}$ can be denoted as

$$\mathbf{Q_k} = E(\mathbf{w_k}\mathbf{w_k^T}), \tag{2.45}$$

where the right side of the equation is the expectation of squared error. The second model, observation model, is used to describe the relationship between the measurement and the new state, formulated as

$$\mathbf{z_k} = \mathbf{H}\mathbf{x_k} + \mathbf{v_k}. \tag{2.46}$$

Here, $\mathbf{z_k}$ is a noisy measurement vector at the *k*th time interval. The relation between state and the measurement is described by matrix $\mathbf{H}$ (observation matrix) and $\mathbf{v_k}$ is the measurement noise, which has the relationship to covariance matrix of the measurement $\mathbf{R_k}$. $\mathbf{R_k}$ can be formulated in a following way:

$$\mathbf{R_k} = E(\mathbf{v_k}\mathbf{v_k^T}), \tag{2.47}$$

which is similar to the process noise uncertainty.

Now that the two model equations are introduced, we can move on to the State Update Equation, which can be expressed as

$$\mathbf{x_k} = \mathbf{x_{k-1}} + \mathbf{K_k}(\mathbf{z_k} - \mathbf{H}\mathbf{x_{k-1}}), \tag{2.48}$$

where $\mathbf{K_k}$ is a Kalman Gain matrix notation, and the other variables remain the same from the equations above. Kalman Gain can be calculated with the following equation:

$$\mathbf{K_k} = \mathbf{P_{k-1}}\mathbf{H^T}(\mathbf{H}\mathbf{P_{k-1}}\mathbf{H^T} + \mathbf{R_k})^{-1}, \tag{2.49}$$

where $\mathbf{P_k}$ is the uncertainty matrix, which can be denoted as:

$$\mathbf{P_k} = (\mathbf{I} - \mathbf{K_k}\mathbf{H})\mathbf{P_{k-1}}(\mathbf{I} - \mathbf{K_k}\mathbf{H})^{\mathbf{T}} + \mathbf{K_k}\mathbf{R_k}\mathbf{K_k^T}, \tag{2.50}$$

where I is an identity matrix.

Now when all of the Kalman filter equations have been described, the whole process can be expressed step-by-step:

1. Guess an initial estimate of $\mathbf{x_0}$ and $\mathbf{P_0}$
2. Extrapolate the state with equation $\mathbf{x_{k+1}} = \mathbf{F}\mathbf{x_k} + \mathbf{G}\mathbf{u_k}$
3. Extrapolate uncertainty with $\mathbf{P_{k+1}} = \mathbf{F}\mathbf{P_k}\mathbf{F^T} + \mathbf{Q_k}$.
4. Compute Kalman Gain with Equation 2.49
5. Update the estimate with State Update Equation 2.48
6. Update the estimate certainty using Equation 2.50
7. Jump back to step two.

Steps 2 and 3 represent the Prediction (Time update) and steps 4, 5 and 6 the Correction (Measurement update) in Figure 2.7.

2.5.4 Extended Kalman Filter

As the measurements are Euclidean distances between the devices, the measurement function can be denoted as

$$h(\mathbf{x}) = \sqrt{(x_a - x_t)^2 + (y_a - y_t)^2 + (z_a - z_t)^2}, \tag{2.51}$$

where (x_t, y_t, z_t) is the position of the tag and (x_a, y_a, z_a) is the position of the anchor respectively.

The equation above is nonlinear, so obviously linear Kalman Filters cannot be used. One computationally inexpensive method is to use EKF. EKF approximates the non-linear function by forming a Taylor Series expansion at the nominal solution [32]. The whole procedure is very similar to linear Kalman Filter, basically the difference is to linearize the state transition matrix $\mathbf{F}$ and the observation matrix $\mathbf{H}$. The linearization in EKF is done by taking the partial derivatives of $\mathbf{F}$ and $\mathbf{H}$, and these resulted matrices are called Jacobian matrices. These Jacobians can be denoted as:

$$\mathbf{F} = \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}\Big|_{\mathbf{x_t}} \tag{2.52}$$

and

$$\mathbf{H} = \frac{\partial h(\mathbf{x})}{\partial \mathbf{x}}\Big|_{\mathbf{x_t}}. \tag{2.53}$$

As a constant position model is used for this Kalman Filter implementation, the state transition matrix $\mathbf{F}$ is linear, and therefore it doesn't need to be linearized. Thus, only the observation matrix has to be linearized. As the measurement function is Equation 2.51, the Jacobian of observation matrix can be written as

$$\mathbf{H} = \begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \frac{\partial h_1}{\partial x_2} & \cdots \\ \frac{\partial h_2}{\partial x_1} & \frac{\partial h_2}{\partial x_2} & \cdots \\ \vdots & \vdots & \end{bmatrix}. \tag{2.54}$$

As our system state for the tag is

$$\mathbf{x} = \begin{bmatrix} x_t & y_t & z_t \end{bmatrix}, \tag{2.55}$$

where (x_t, y_t, z_t) is the position of the tag, the Jacobian of the observation matrix H can then be calculated as

$$\mathbf{H} = \begin{bmatrix} \frac{\partial h}{\partial x} & \frac{\partial h}{\partial y} & \frac{\partial h}{\partial z} \end{bmatrix}. \tag{2.56}$$

The final result for the Jacobian of the observation matrix $\mathbf{H}$ will be:

$$\mathbf{H} = \begin{bmatrix} \frac{x_t - x_a}{\sqrt{(x_a - x_t)^2 + (y_a - y_t)^2 + (z_a - z_t)^2}} \\ \frac{y_t - y_a}{\sqrt{(x_a - x_t)^2 + (y_a - y_t)^2 + (z_a - z_t)^2}} \\ \frac{z_t - z_a}{\sqrt{(x_a - x_t)^2 + (y_a - y_t)^2 + (z_a - z_t)^2}} \end{bmatrix}^T. \tag{2.57}$$

This resulted Jacobian matrix $\mathbf{H}$ will be calculated every time new values are measured.

2.5.5 Unscented Kalman Filter

Another alternative is to use the Unscented Kalman Filter (UKF). UKF handles the nonlinearities via unscented transform. While EKF tries to approximate the nonlinear function, UKF tries to approximate the covariance and mean of the target distribution. Basically a number of sigma points are chosen deterministically which capture the covariance and the mean of the original distribution $\mathbf{x}$. The chosen sigma points are then propagated through the nonlinearity, and the covariance and mean are estimated from the transformed variable. Figure 2.8 illustrates the unscented transform.

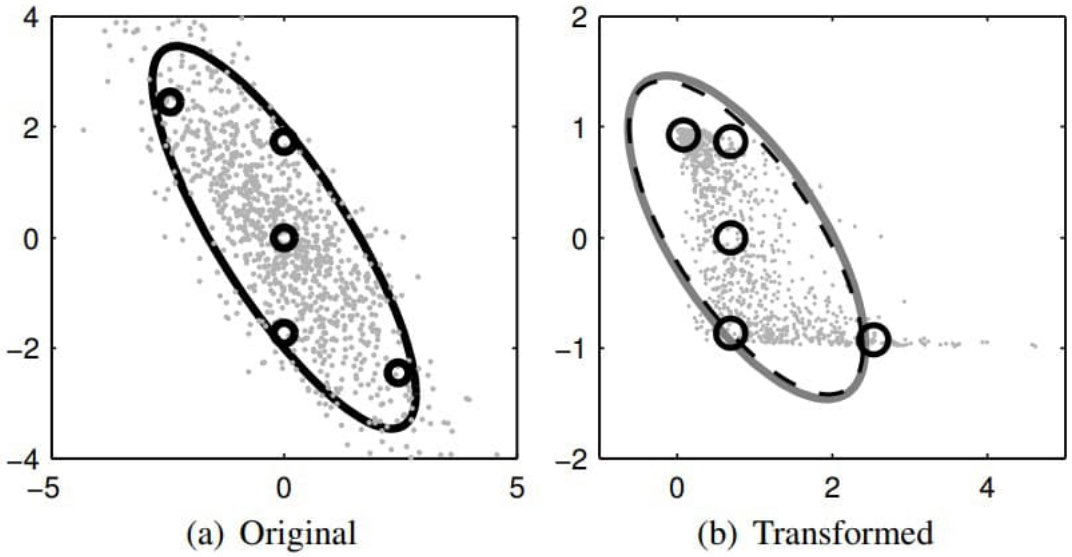

Figure 2.8. *Illustration of approximation to the transformation based on the unscented transform. The solid line represents the covariance of the true distribution and the dashed line represents the approximation. Adapted from [32].*

There are several methods for choosing the sigma points, for instance one proposed by

Van Der Merwe & Wan [33] and another by Julier & Uhlmann [34]. In this thesis the sigma points are chosen with the method proposed by Van Der Merwe as the Python library for the UKF implementation recommends so [35].

3. MATERIALS AND METHODOLOGY

This chapter provides detailed information on the devices used in this thesis for the position estimation. Also, the indoor positioning setup is explained thoroughly (i.e how and where the devices are placed) and how the data is gathered. Then, the experiments conducted in this thesis are discussed. Finally, software implementations and performance metrics are discussed in the last two subsections.

3.1 Devices

The devices used in this thesis are called tags, anchors and gateways. Each of these devices are IC's (Integrated Circuits) which serve a different purpose for fulfilling this positioning task. Although their hardware is almost identical, they are named uniquely to emphasize their functioning. Figure 3.1 visualizes the different devices used in the thesis. On the left side of the upper row is a gateway and next to it on the right is an anchor. The wristband below is a tag, and inside the casing resides a similar IC as the anchor and also a lithium polymer (LIPO) battery. The tags, which are mobile nodes, are the main interest

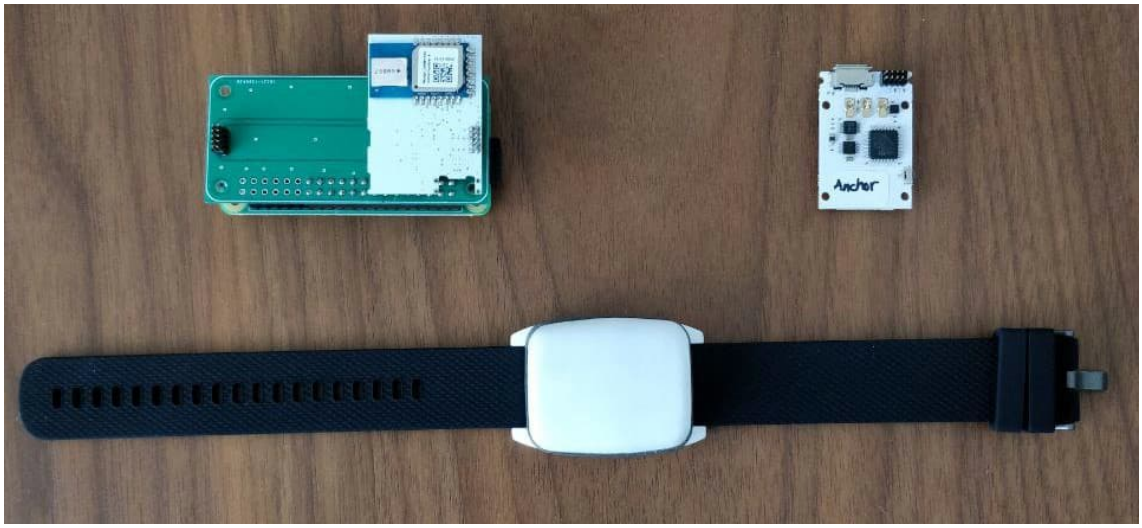

Figure 3.1. *Devices used in this thesis. In Figure 3.4 the anchor is inside the casings.*

in this thesis, because they are being tracked. They physically reside in a wrist-wearable bracelet, so a moving persons position can be estimated. The anchors are fixed nodes with a known position, but they are essential for calculating the distance to tag and also

for the position estimation. The anchors will be mounted on a wall with their own casings. Finally, the gateway serves as its name implies as a tool to transmit data to the cloud. The tag runs on battery, because it would be very restrictive for a person to be connected to a wire. The anchors can be powered via either battery or USB cable, and in this thesis they powered via USB. The gateways need to be connected to 230 V sockets because the gateway has much higher power consumption than the other devices.

On the hardware side, these devices are almost identical when considering electrical characteristics. Each of the IC's are packed with a STM32L071KZT6 (STMicroelectronics, Geneva, Switzerland) microprocessor, which is designed for low-power applications. This is essential due to the fact that the tags electronics reside in a wristband thus having limited space. If the power consumption would be higher, the battery would also need to be bigger which would affect the visual side of the wristband. The STM32 microprocessor handles all the communication with the peripherals such as the RF (radio frequency) circuit and the accelerometer.

The accelerometer integrated in the circuit is a digital 3-axis MC3419 (MEMSIC Semiconductor Co., Ltd., Tianjin, China). Accelerometer is a small component which can be found in many smartphones and it can be used to detect acceleration and motion. MC3419 is a Micro-Electro-Mechanical (MEMS) and therefore it is called a MEMS accelerometer. The accelerometer data is not used in this thesis, but this paragraph is included in this thesis because the accelerometer is part of the microcontroller and it could be utilized in sensor fusion (UWB + accelerometer data).

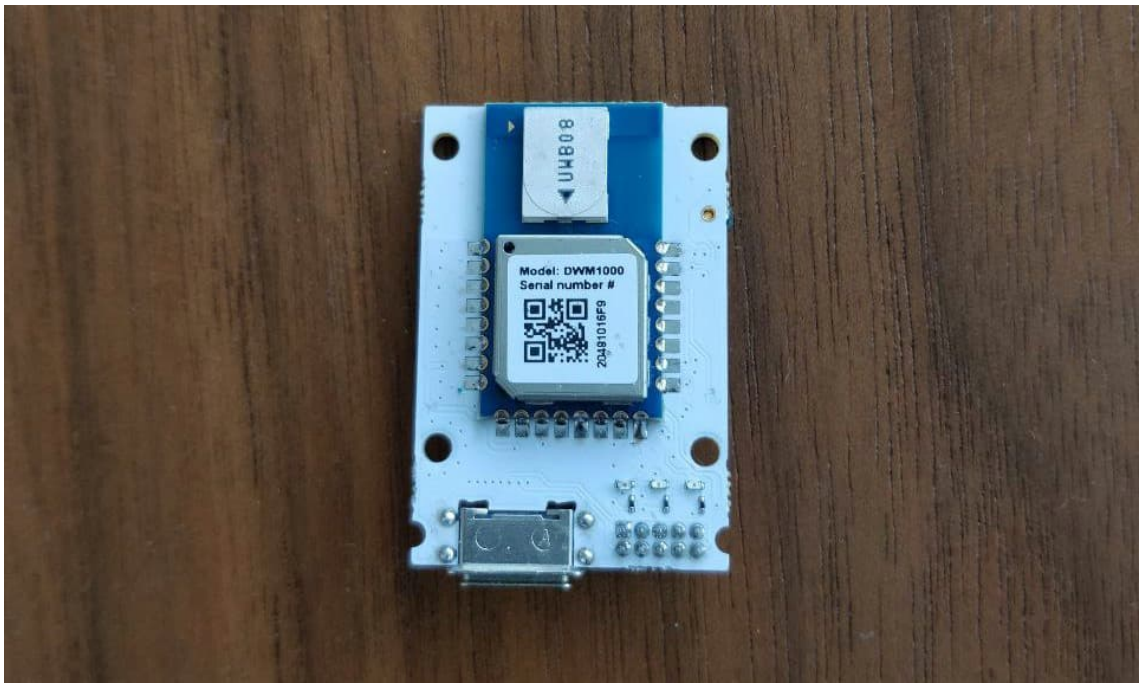

Figure 3.2. *Qorvo DW1000 UWB module attached to an IC.*

The other peripheral, RF circuit, is a Qorvo DW1000 module which was mentioned in

Chapter 2. The DW1000 module is essential for the ranging, because it can send accurately timed messages, and record received messages timestamps, which are used to calculate the distance between two devices. Because the tag is battery powered and typically RF circuits can draw big currents, it is necessary to keep the transmission times as small as possible. To decrease the power consumption even more, the device is set to deep-sleep mode between adjacent transmissions. A DW1000 module can be seen on the Figure 3.2.

The gateway handles the data transmission from the tags and anchors to the cloud, in our case Google Cloud Platform (GCP). In addition to the customized integrated circuits which the tags and anchors have, gateway also has a Raspberry Pi Zero (Raspberry Pi Foundation, Cambridge, United Kingdom) microcomputer. The IC transmits the distance data via Universal Asynchronous Receiver Transmitter (UART) to the Raspberry Pi and then the Pi transmits the data onwards to the cloud. The Raspberry Pi is essential part of the setup, because the tag and the anchors can only send UWB data.

3.2 Indoor Positioning Setup

In a real world application, the system would track a person moving in a room. For achieving this goal, a correct device setup must be done. The anchors should be placed to the walls of the room, approximately in a rectangular shape. After that, an origin should be decided in the room, typically in one of the corners of the room. Then, the anchors position should be measured related to the origin, so afterwards the moving tag's location can be estimated. In this setup, the upper right corner is chosen to be the origin. Figure 3.3 illustrates Veracell Ltd's office setup. The texts under the anchors represent their unique identifiers, so the devices know who are they communicating with and the rectangles in the image represent walls and rooms, such as the kitchen and the meeting room.

The anchors are mounted with two-sided tape for this setup to avoid making damage to the walls. This is possible as the anchor cases are lightweight. Wall plugs are not considered as they leave holes in the wall, and moving the anchors should be a swift task. As mentioned in the previous chapter, they can be powered with batteries or with USB. In the office setup, they are powered with USB as changing the batteries in testing and development phases would be a tedious task.

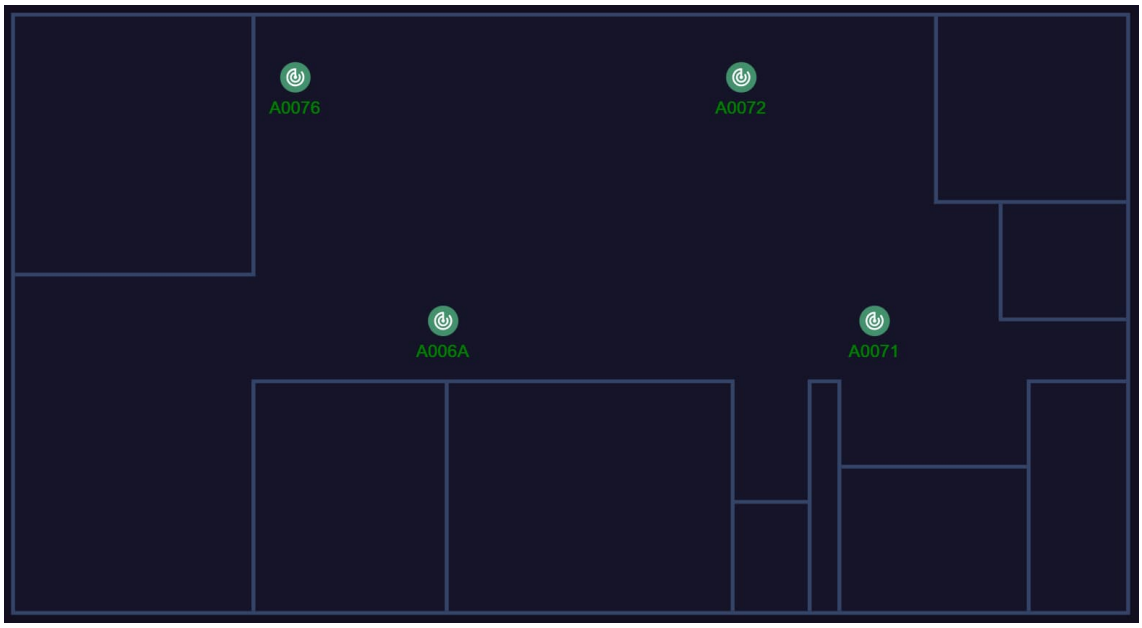

Figure 3.3. *Office indoor positioning setup.*

Figure 3.4. *Anchor mounted on a wall.*

After the anchors are placed, their position is measured and the gateway is powered, the setup is ready to produce data. The process of acquiring the three-dimensional location data from the tag is explained by the following procedure:

1. Tag blinks to all anchors.
2. Each anchor responds to the tag on their own turn.
3. Tag sends the final message to these anchors. At this point it is already possible to calculate distance from the tag to each anchor.
4. Anchors send their final responses to the tag.
5. Tag sends a message to gateway containing distances from each anchor.
6. Gateway estimates the position of the tag, and sends the data to a cloud platform.
7. A cloud function [36] inserts the position to a database.
8. The data can be accessed with Structured Query Language (SQL) queries and visualized with Jupyter Notebook [37].

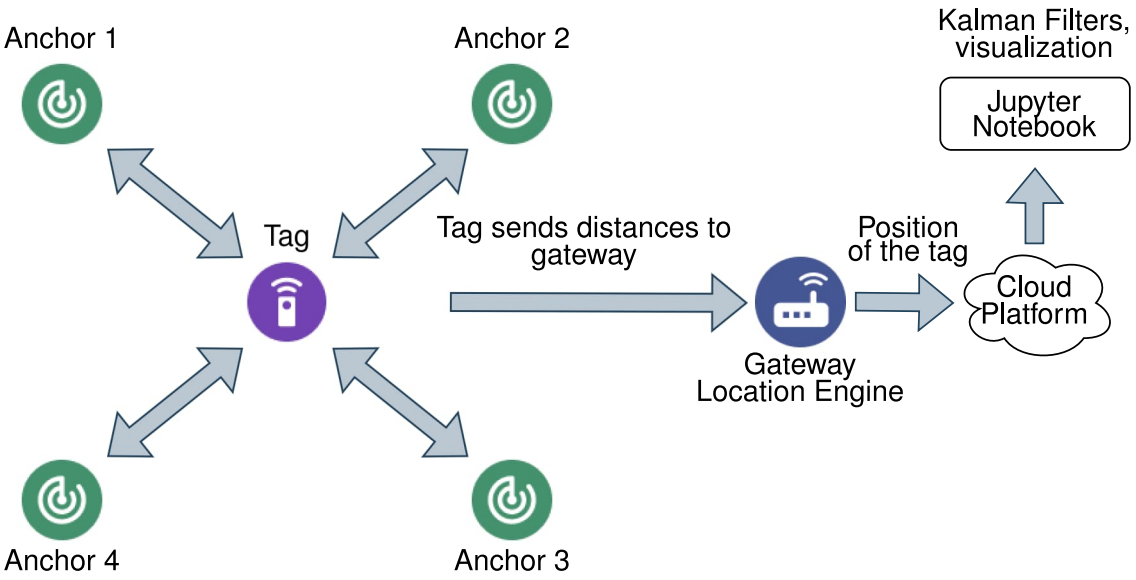

Figure 3.5. *Illustration of data pipeline.*

The same procedure of the data pipeline is illustrated in the Figure 3.5.

3.3 Experiments

To implement Kalman Filters for this positioning problem, a few different experiments were conducted in this thesis. The first one was to gather information about the UWB ranging errors in LOS and NLOS scenarios. The second and third experiments were to gather data from two different setups and compare the performance of the implemented Kalman Filters. The following subsections will go through the experiments conducted in this thesis.

3.3.1 LOS and NLOS standard deviations

It is typical to have some kind of prior information of measurements for the Kalman Filters. In this case, standard deviation of LOS measurements and NLOS measurements should be calculated, so the standard deviations can be updated to the measurement covariance matrix $\mathbf{R_k}$ dependent whether the measurement is LOS or NLOS.

For this experiment, the tag was left in a position where there is a clear line-of-sight to some of the anchors, and to the other anchors there were objects blocking the path to acquire NLOS ranging samples. The LOS probability was acquired using the heuristic Equation 2.37. The ground truth distances were also measured from the tag to each anchor, so the ranging error could be calculated. The results of this experiment are discussed in Section 4.

3.3.2 Data Gathering

After the standard deviations are calculated for LOS and NLOS scenarios, the datasets for the experiment number two and three are collected. For this thesis, two different datasets are gathered. The datasets contain distance measurements from each anchor to the moving tag and respective RX power and FP power values for each distance. The RX power and FP power values are for the LOS/NLOS heuristics.

For the data gathering process, the tag was attached to a wrist (see Figure 3.6), and it was carried along the room with constant height and constant speed. The steps were synchronized with the tag's blinking LED, so creating the ground truth dataset was a lot easier. The timestep for acquiring distances was set to $\Delta t = 1$, which means the tag started the messaging procedure (3.2) once in a second. Higher messaging frequencies could be used for these experiments, but in a real world application, it would increase the power consumption too much.

To test different scenarios for the Kalman Filters, the walking trajectory for the datasets were different. For the first trajectory, a simple path consisting of two lines and a turn was walked on. For the second path, the shape resembles a rectangle, so there are more turns. This will make the path a bit more complex, and thus provide a bigger challenge for the Kalman Filters. Figures 3.7 and 3.8 represent the walked trajectories respectively. In the figures, the starting points are marked, and the red paths are walked on. In the third experiment, the rectangle shaped path was walked clock-wise.

As the measured distances were retrieved from the database, some kind of alignment procedure had to be done to fetch the correct measurements. As the path was walked along, the exact UTC timestamps for the starting point and for the ending point was marked up, so then it could be deduced from which time period to search the measurements from.

Figure 3.6. *Tag attached to wrist.*

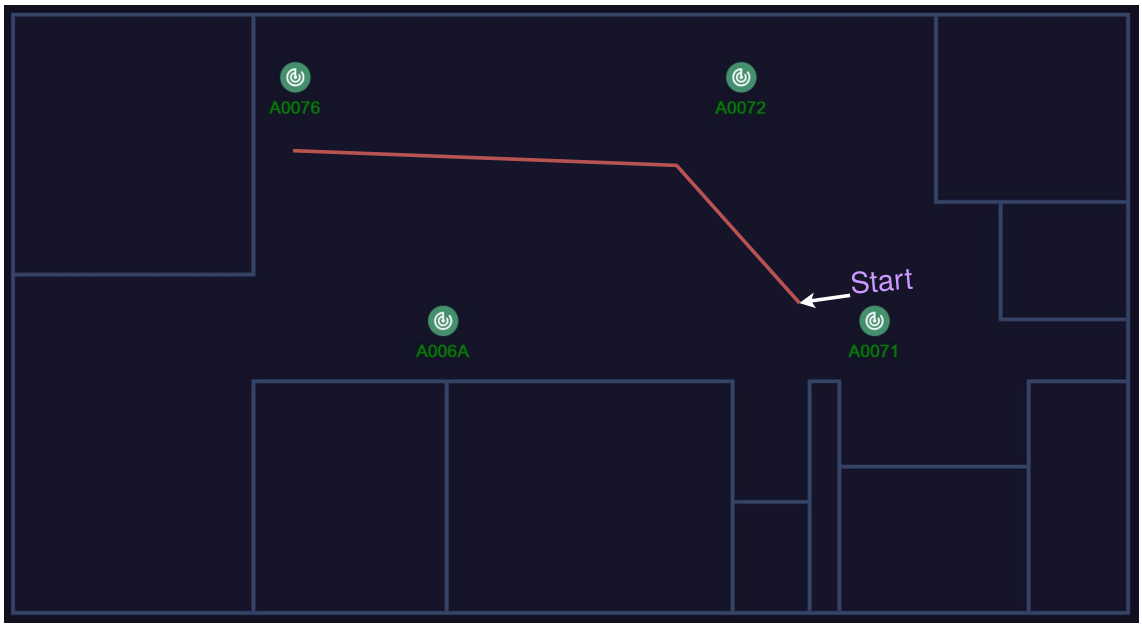

Figure 3.7. *Second experiment trajectory.*

Because the samples were sent from the tag to gateway and from there on wards to cloud, there was a small offset in the timestamps (caused by processing time, data transmission etc.), and this had to be taken into account. DBeaver, which is a open-source database tool [38], was used for retrieving the measurements. In the second experiment there were 17 data samples and in the third experiment 47 samples. Each sample has ideally four distances from the anchors, and this was the case in both experiments. However, in some

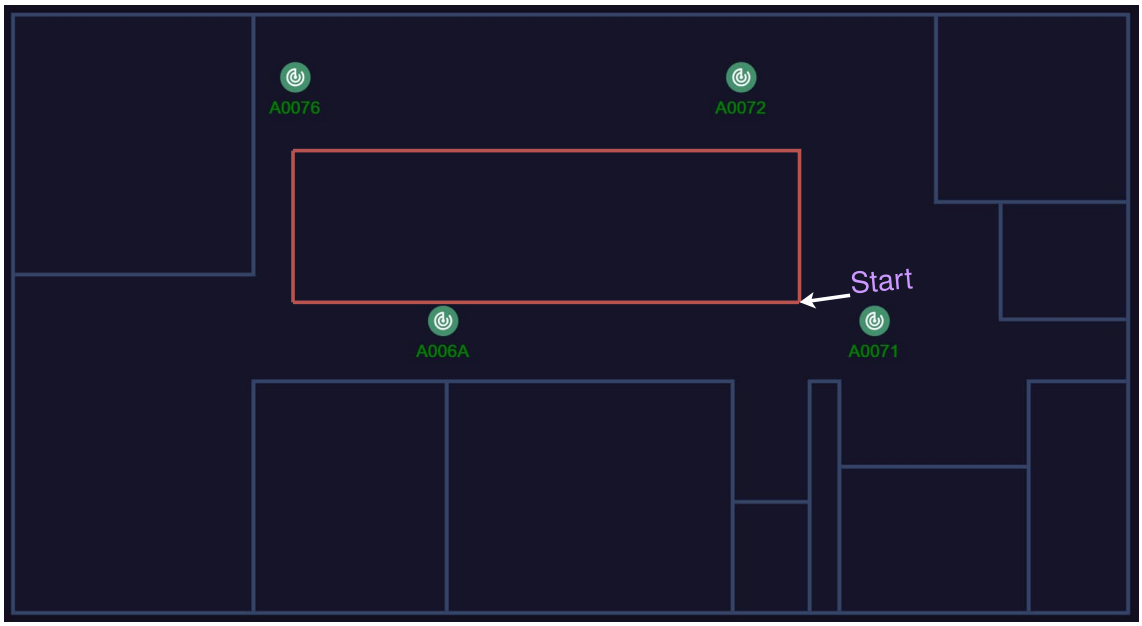

Figure 3.8. *Third experiment trajectory.*

other scenarios where there might be a too long distance to an anchor or large concrete walls blocking the signal, there may not always be four distance measurements.

3.3.3 Data Preparation

After the distances were acquired, the data had to be preprocessed before it could be fed to the Kalman Filters. The data retrieved with the SQL query had a lot of additional data, for instance tag battery information and acceleration data. Additionally, the distances were reported in millimetres, so they were converted to metres for the Kalman Filters.

Along the distance measurements there are RX power and FP power values in the dataset. To figure whether the samples were LOS or NLOS the Equation 2.37 was used. The samples were then tagged with a binary flag whether it is a LOS or NLOS, and then it could be identified when building the Kalman Filter's measurement covariance matrix $\mathbf{R_k}$.

Ground truth values of the position were created for Kalman filter performance evaluation. Detailed explanation of the accuracy measures is presented in Section 3.5. The ground truth values were generated with the help of interpolation, as it would have been very tedious and time consuming to measure each step in the both trajectories with a tape measure. In the second experiment three points where picked, so there would be two lines drawn between them and the datapoints were interpolated between these points. In the third experiment, each corner of the rectangle were picked as a reference point, and the ground truth points were interpolated between them. In both of these experiments, the z point was constant, as the tag was held at constant height. This is a simplification

for this thesis, as normally humans do not walk having their arm at a constant height. If the hand where the tag was attached to would have moved freely, it would have been very hard to know the ground truth positions in z-axis.

3.3.4 Kalman Filter Parameters

In order to estimate position of the tag with Kalman Filters, appropriate parameters need to be initialized. As mentioned in the Chapter 2.5.2, Kalman Filter has many components which need to be defined, for instance, the state vector $\mathbf{x}$, state transition matrix $\mathbf{F}$, process noise covariance matrix $\mathbf{Q}$, observation matrix $\mathbf{H}$ and finally the measurement covariance matrix $\mathbf{R_k}$. Parameters $\mathbf{x}$, $\mathbf{Q}$ and $\mathbf{H}$ are defined for three dimension positioning, but for the 2D positioning, the z-axis is just left out.

As we want to track the position of the tag, the state vector $\mathbf{x}$ is defined as:

$$\mathbf{x} = \begin{bmatrix} x_t & \dot{x}_t & y_t & \dot{y}_t & z_t & \dot{z}_t \end{bmatrix}, \tag{3.1}$$

where (x, y, z) is the position of the tag and $(\dot{x}, \dot{y}, \dot{z})$ are velocities of the tag in their respective dimensions. As we use the constant velocity model, the extrapolated state can be expressed with the following motion equations:

$$\begin{cases} x = x_{n-1} + \dot{x}_{n-1}\Delta t + \frac{1}{2}\ddot{x}_{n-1}\Delta t^2 \\ \dot{x} = \dot{x}_{n-1} + \ddot{x}_{n-1}\Delta t \\ y = y_{n-1} + \dot{y}_{n-1}\Delta t + \frac{1}{2}\ddot{y}_{n-1}\Delta t^2 \\ \dot{y} = \dot{y}_{n-1} + \ddot{y}_{n-1}\Delta t \\ z = z_{n-1} + \dot{z}_{n-1}\Delta t + \frac{1}{2}\ddot{z}_{n-1}\Delta t^2 \\ \dot{z} = \dot{z}_{n-1} + \ddot{z}_{n-1}\Delta t \end{cases}, \tag{3.2}$$

and because acceleration is assumed to be zero, the state transition matrix $\mathbf{F}$ can be denoted as:

$$\mathbf{F} = \begin{bmatrix} 1 & \Delta t & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & \Delta t & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3.3}$$

where $\Delta t = 1$, which is the interval for the measurements. The process noise covariance

matrix $\mathbf{Q}$ is

$$\mathbf{Q} = \sigma^2 \begin{bmatrix} \frac{\Delta t^4}{4} & \frac{\Delta t^3}{2} & 0 & 0 & 0 & 0 \\ \frac{\Delta t^3}{2} & \Delta t^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\Delta t^4}{4} & \frac{\Delta t^3}{2} & 0 & 0 \\ 0 & 0 & \frac{\Delta t^3}{2} & \Delta t^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{\Delta t^4}{4} & \frac{\Delta t^3}{2} \\ 0 & 0 & 0 & 0 & \frac{\Delta t^3}{2} & \Delta t^2 \end{bmatrix}, \tag{3.4}$$

where σ^2 is the process noise variance, and it is chosen to be 0.05. There are a lot of zeros in the matrix because we assume that the process noise is uncorrelated between the axes x, y and z.

As discussed earlier in Subsection 2.5.4, the measurements we receive are not linear. Observation matrix implies linearity, so we will use an observation function instead, denoted as

$$\mathbf{H} = \begin{bmatrix} \sqrt{(x_{a,1} - x_t)^2 + (y_{a,1} - y_t)^2 + (z_{a,1} - z_t)^2} \\ \sqrt{(x_{a,2} - x_t)^2 + (y_{a,2} - y_t)^2 + (z_{a,2} - z_t)^2} \\ \sqrt{(x_{a,3} - x_t)^2 + (y_{a,3} - y_t)^2 + (z_{a,3} - z_t)^2} \\ \sqrt{(x_{a,4} - x_t)^2 + (y_{a,4} - y_t)^2 + (z_{a,4} - z_t)^2} \end{bmatrix}, \tag{3.5}$$

where $(x_{a,i}, y_{a,i}, z_{a,i})$ is the position of the *ith* anchor and (x_t, y_t, z_t) is the position of the tag. As the anchors are fixed, their position remains the same, but for the tag, the position needs to be updated before the predict and update steps.

For the final parameter we need to set the measurement covariance matrix $\mathbf{R_k}$, it is initialized as follows:

$$\mathbf{R_k} = \begin{bmatrix} \sigma_{LOS,1} & 0 & 0 & 0 \\ 0 & \sigma_{LOS,2} & 0 & 0 \\ 0 & 0 & \sigma_{LOS,3} & 0 \\ 0 & 0 & 0 & \sigma_{LOS,4} \end{bmatrix}, \tag{3.6}$$

where $\sigma_{LOS,i}$ is the standard deviation of the measurement error. The value is either set to σ_{LOS} or σ_{NLOS} whether the measurement from the anchor is LOS or NLOS. The values for these standard deviations are calculated with the Equation 3.7 and the resulted values are given in Chapter 4.

The parameters defined above were used for the Kalman Filters itself, but for the Unscented Kalman Filter some additional parameters had to be defined. These parameters define how the sigma points are generated, and thus affects how the unscented trans-

form is performed. For the Van Der Merwe's sigma points, parameters α, β and κ had to be defined. α defines the spread of the sigma points around the mean, typically a small positive value is used. β incorporates prior knowledge of the distribution of the mean. For Gaussian distribution, $\beta = 2$ is optimal. Finally, κ is a scaling parameter, typically set either to n-3 or 0, where n is the dimension of the state vector.

Table 3.1. *Van Der Merwe Sigma points parameters*

Parameter	Value
α	0.01
β	2
κ_{2D}	-1
κ_{3D}	-3

The chosen values for the parameters are given in Table 3.1. κ has two possible values because it is dependent on the dimension of the state vector $\mathbf{x}$. All of these parameter values were suggested in Van Der Merwe's Doctoral dissertation [39].

3.4 Implemented Software

As the experiments included hardware, the first software implementations were on the firmware side. As the microcontrollers had a STM32 microprocessor, the same company's provided integrated development environment STM32CubeIDE was used. STM32CubeIDE can be utilised to code in either C/C++. For this project, the firmware was done in C [40]. The environments provides intuitive debugging interfaces and more tools. Luckily, most of the firmware code was already implemented, so only configuration of the DW1000 and a little firmware modifications had to be done.

For SQL queries, data preparation, Kalman Filters and visualization, Python 3.9 [41] was used. As there are sufficient amount of libraries for Python, they were utilized as well. For data retrieval, processing and visualization I used Psycopg [42], Pandas [43], NumPy [44], Matplotlib [45] and Seaborn [46]. For the Kalman Filter implementation, I used library called FilterPy, authored by Roger Labbe, as the book [35] had examples implemented with Python. The development environment used for Python programming was Microsoft Visual Studio Code, as it has useful functionalities.

For the experiments of this thesis, EKF and UKF were implemented. The code was implemented to resemble a Python library in a modular way, so these implementations can be later revisited, and easily modified to match the user needs. The visualization code was implemented in Python scripts.

The optimization method which is included in the Figures of the results section, was already implemented, so it was only used to compare the performance of the Kalman

Filters.

3.5 Statistical Methods

For the first experiment, the standard deviations of LOS and NLOS distributions can be calculated with the following equation

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}, \text{ where } \mu = \frac{1}{N}\sum_{i=1}^{N} x_i, \tag{3.7}$$

where x represent the dataset of the ranging errors. For experiments two and three, which incorporated the implementation of EKF and UKF, some kind of error metrics have to be introduced, in order to compare the results. For positioning problems it is typical to compare the predicted position to the ground truth position with Euclidean distance. For this thesis the positioning error was calculated with the following equation

$$e = ||\hat{\mathbf{x}}_\mathbf{t} - \mathbf{x}_\mathbf{t}||, \tag{3.8}$$

where $\hat{\mathbf{x}}_\mathbf{t}$ is the predicted position of the tag and $\mathbf{x}_\mathbf{t}$ the actual position. The error was calculated in 2D and 3D for each time step individually for both of the experiments. As for the resulted error set, three different accuracy measures were used for the evaluation:

- *Mean error*: Mean of the positioning errors
- *Median error*: Median of the positioning errors
- *95% error*: 95th percentile of the positioning errors.

Also, for a real-time positioning application the execution speed is crucial, so the execution times of each implementation were measured as well.

4. RESULTS

This chapter goes through figures and tables resulted from the conducted experiments.

In total 3180 ranging samples were acquired for the first experiment, which consisted of 1148 LOS samples and 2032 NLOS samples respectively. Figure 4.1 represents the distributions of the ranging errors. In the figure there are histograms of the LOS and NLOS samples highlighted with different colors, and the lines over the histograms represent a Kernel density estimation (KDE) which is a method to estimate the probability density function [47].

From the figure it can be seen that the distributions resemble a Gaussian distribution decently, especially the LOS distribution. The NLOS distribution is a bit more wider, which was also expected. Ideally the distributions would be zero mean, but there is negative offset in both of the distributions. This is most likely caused by the nonidealities in the UWB modules, especially the antennas. The devices could be calibrated to negate this offset, but that is out of scope of this thesis. The standard deviations of the distributions were calculated with the Equation 3.7. The resulted standard deviations are $\sigma_{LOS} = 0.0285$ and $\sigma_{NLOS} = 0.0368$ and the values are expressed in metres. These standard deviations were used to update the Kalman Filter's measurement covariance matrix $\mathbf{R_k}$.

For experiments two and three, both of the figures' axes are represented in metres and they are on the same scale as the dimension of the office's setup. The axes are inverted so the walked path in the figures would be similar to Figures 3.7 and 3.8, and thus the figures are easier to interpret. The icons of the anchors are left out of these figures to make them tidier. Also, there were a lot of objects in the office, such as tables, sofas and monitors, but they were left out of the figures for the same reason as the anchor icons.

The blue dots in both of the Figures 4.2 and 4.3 depicts the measured ground truth points and the blue lines represent the interpolated estimation of the walked trajectory. Alongside the interpolated trajectory, in the figure there are the Kalman Filter and optimization method estimated trajectories. In both of the figures, the methods are color coded identically to avoid misinterpretation. Orange trajectory is the estimated trajectory produced by the optimization method, green is the estimate produced by EKF and finally the red one is produced by UKF. All of the trajectories plotted in these figures are based on the two dimensional positioning.

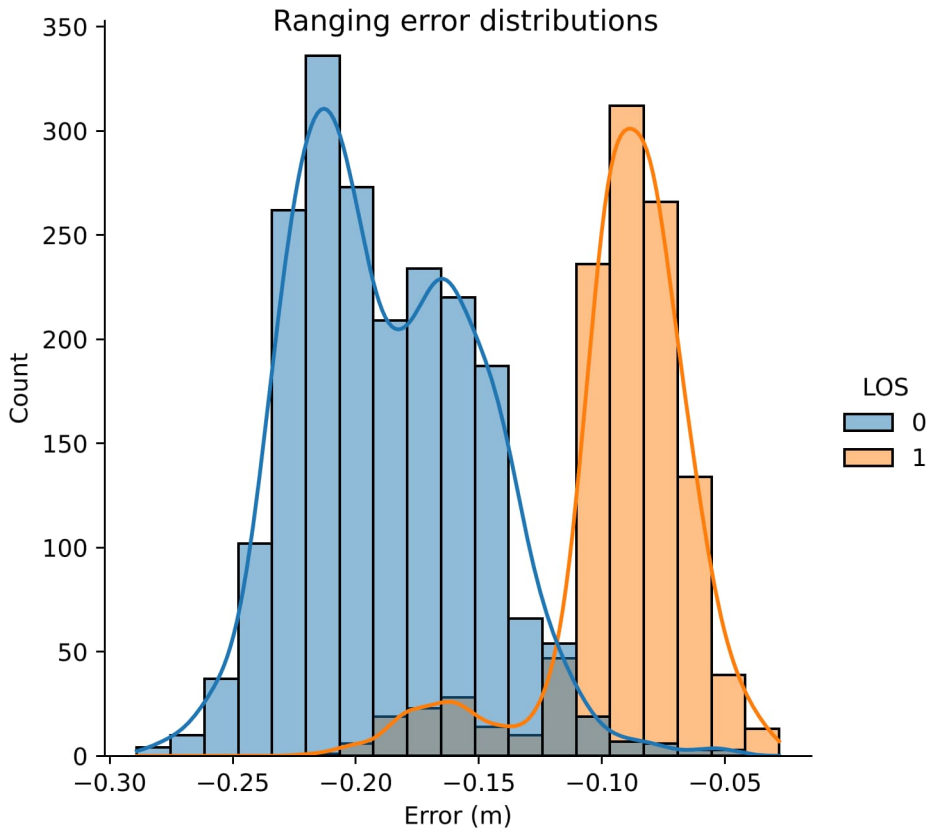

Figure 4.1. *Distribution of LOS and NLOS ranging errors.*

As stated earlier in the Section 3.4, the optimization method code was already implemented before this thesis, and it was left to the figures and tables for performance comparison purposes.

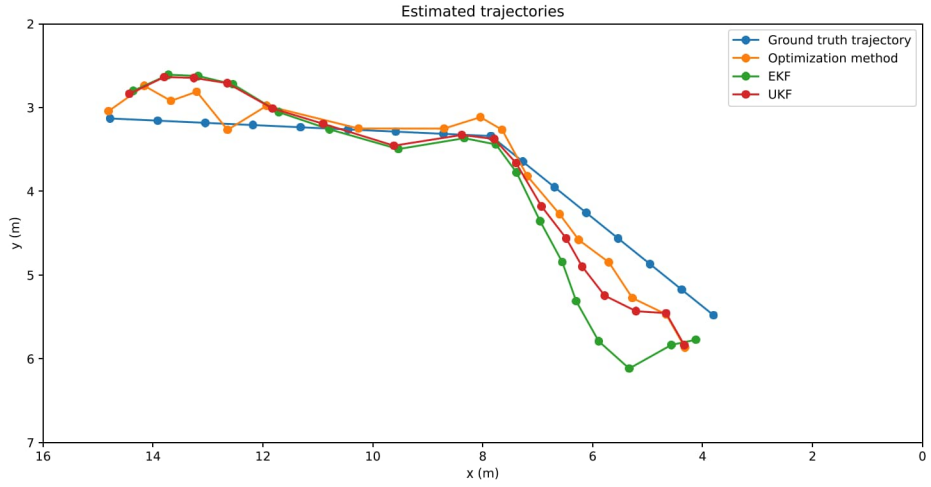

Figure 4.2. *Ground truth trajectory and estimated trajectories by optimization method, EKF and UKF for the experiment 2.*

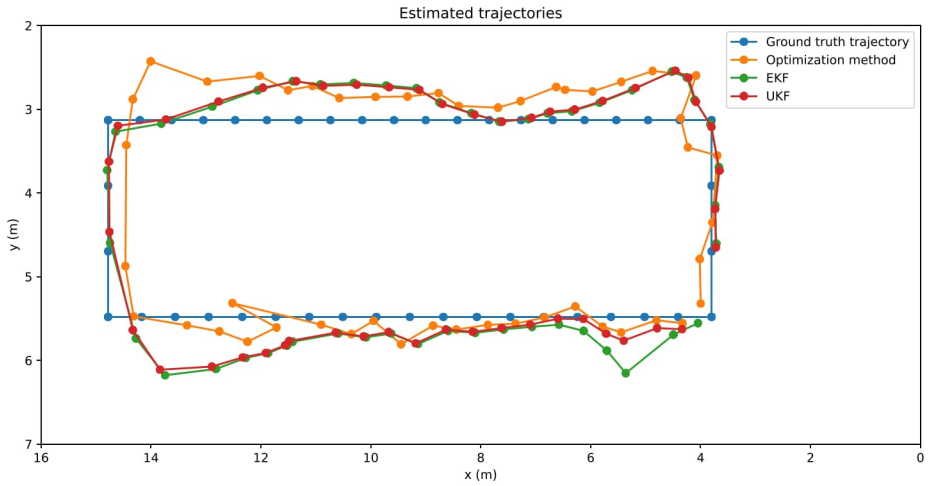

Figure 4.3. *Ground truth trajectory and estimated trajectories by optimization method, EKF and UKF for the experiment 3.*

Table 4.1 and 4.2 represent the resulted accuracy measures and execution times of the optimization method and Kalman Filters in experiments 2 and 3. Both of the tables include 2D and 3D implementations for the experiments. Execution time is calculated for every datapoint and it is measured in milliseconds. Mean, median and 95th percentile errors are measured in metres.

In the second experiment the reported accuracy scores are almost identical, with the exception of UKF having clearly lower error rates based on 95% measure. In two dimensions, UKF has the best performance according to 95% and mean error, and optimization method has the best median error. In three dimensions optimization method has the highest accuracy in mean and median error, but UKF has the best accuracy score for 95% error once again.

Table 4.1. *Execution times and accuracy measures of optimization method, EKF and UKF for experiment 2.*

Method	Time (ms)	Mean error (m)	Median error (m)	95% error (m)
Optimization 2D	74	0.54	**0.41**	1.36
EKF 2D	**2.8**	0.64	0.58	1.28
UKF 2D	11	**0.51**	0.55	**0.69**
Optimization 3D	101	0.62	0.52	1.45
EKF 3D	4	0.63	0.58	1.22
UKF 3D	14	0.63	0.66	0.96

For the second experiment, the errors in each category are higher, as the trajectory is much longer and has harder maneuvers. Also, the execution times are much longer, as the number of data points is higher. In two dimensions, EKF outperforms optimization

Table 4.2. *Execution times and accuracy measures of optimization method, EKF and UKF for experiment 3.*

Method	Time (ms)	Mean error (m)	Median error (m)	95% error (m)
Optimization 2D	211	0.79	0.79	**1.45**
EKF 2D	**8.6**	**0.77**	**0.72**	1.66
UKF 2D	27	**0.77**	0.77	1.62
Optimization 3D	294	1.73	2.04	2.55
EKF 3D	**9**	1.95	2.25	2.68
UKF 3D	36	1.02	0.95	1.77

method and UKF based on mean and median error, and optimization method has the lowest 95% error. In three dimensions, UKF has considerably better performance in all of the accuracy categories.

In both of the experiments, EKF has significantly the lowest execution time for 2D and 3D positioning. It is roughly four times faster than UKF and tens of times faster than optimization method. However, in most of the cases EKF has the lowest accuracy score, so this trade off has to be taken into account. As UKF and optimization method have quite similar accuracy scores, the next step is to compare the computation times, and UKF is clearly a better choice where low computation times are crucial.

It is also important to consider the estimated trajectories in Figures 4.2 and 4.3 rather than only focusing on the reported values in the tables. It appears that optimization method provides more accurate estimates when comparing single position points, but the trajectories have occasional zigzag patterns. This may not be the optimal case if some kind of metrics are being calculated from the position estimates, for example travelled distance. Both of the Kalman Filter implementations have a much more smoother trajectories, but they also react to turns a little bit slower. Also, in the first experiment EKF estimates the direction incorrectly, which leads to big inaccuracies in the estimates.

5. DISCUSSION

This chapter goes through the results from Chapter 4 which are reflected to the research questions presented in Chapter 1.

The literature review was done based on the first research question. I argue that the literature review provided valuable information to Veracell, as it has information about UWB itself, UWB positioning parameters and positioning with Kalman Filters. This way the company can consider for example is the optimization method the optimal solution for positioning, or should some other method be considered. This information is general from the UWB positioning perspective, and it is not tied only to the experiments of this thesis. The information is very useful for Veracell's product development, as in the future the RF modules might be replaced with other ones which do not support the same positioning parameters. Also, as the Chapters 2 and 3 contain figures from e.g positioning parameters and data pipelines, this thesis could be used to provide an introduction to newcomers for the product development.

A significant motivator which lead to the investigation of the Kalman Filters in the literature review, was the fact that the optimization method was known to output independent positioning results, which did not take into account anything else than the distances from the anchors and the anchors positions. This could cause sometimes very zig-zag like movement, which rarely represents the actual movement of the tracked object. The literature review produced an overview of Kalman Filters, which outlined the importance of how Kalman Filters model noise and uncertainty, as the distance measurements always incorporate some amount of noise. Also, one beneficial aspect of the Kalman Filters is that they use the tracked object's motion model (physical laws of motion for instance), which essentially helps producing more natural looking trajectories.

The experiments conducted in this thesis were to answer the second research question. The Kalman filters implemented for the experiments performed reasonably well considering the time limits. If one regards only the accuracy values reported in the tables, there are some cases where the optimization methods outperforms the Kalman Filters. However, also the shape of the trajectories has to be considered when the system wants to track human movement. In both of the Figures 4.2 and 4.3, optimization method's trajectory has jittering, which is not natural human movement in most scenarios. On the other

hand, both EKF and UKF managed to keep the trajectory relatively smooth.

Due to time constraints, the parameters of the Kalman filters are not optimized, and testing the filters with different parameters could be valuable. For example, different sigma points could be tested for UKF. However, as the real positioning scenarios are very dynamic and most of the time do not match the scenarios in this thesis, the optimization should not be given too much attention. It would be more reasonable to find parameters which would help the Kalman Filters to perform better generally. This kind of optimization would require more testing in different scenarios and environments.

The room setup could be generalized to any other indoor positioning setup which consists one room. On the other hand, the results can't be generalized to every scenario, as the movement in the experiments is rather robotic and simple, actual everyday human movement is much more complex. In addition, in other setups there might be more objects and obstacles blocking the path.

As the LOS diagnostics were gathered from the experiments two and three, one interesting aspect could be to examine how the number of LOS/NLOS measurements per time step affects the positioning accuracy. In the experiments of the thesis, the standard deviations were updated based on the signal being LOS or NLOS, but the impact of this aspect was not tested in the experiments two and three. Also, in the previously mentioned experiments, four distance measurements were acquired from the anchors each time step. Sometimes, there might be less than four measurements from the anchors due to some blocking objects. This would significantly impact the positioning accuracy. These two ideas are left for future experiments.

The anchor placement in the experiments two and three is rather optimal. Actual environments could consist of some kind of corridors and more complex shaped rooms for example. In this thesis, the room is shaped as a rectangle and the anchors had most of the time LOS to the tag. The results of this thesis does not yet tell which would be the optimal choice for the indoor positioning, but it gives valuable insights and provides a base understanding of the Kalman Filters. If the product development is pushed forward, these filters need to be tested in multi-room setups and more complex environments.

Future work could include trying different positioning approaches, such as Particle Filters. Particle Filter, which is a nonlinear model, is based on Monte Carlo simulation and Bayesian statistical theory. The way Particle Filter works, is that it generates weighted Monte Carlo samples of the tracked object's state at each time step [48]. Particle Filters are typically more computationally heavy compared to EKF and UKF, but useful information can be used to weigh the particles, such as a floor plan. Another interesting aspect would be to utilize Sensor Fusion, which is the process of combining data from different sensors. As the tags have an accelerometer in them, the accelerometry data could be used for the Kalman Filters. However, in this case, the sampling frequencies of the differ-

ent sensors needs to be taken into account, as the accelerometer typically works with a relatively high frequency compared to the distance measurements.

6. CONCLUSION

This thesis investigated Ultra-wideband (UWB) technologies and the utilization of Kalman Filters in indoor positioning. The main motivation for the thesis was to provide information for Veracell how Kalman Filters would perform in their indoor positioning system (IPS) compared to existing positioning solution. The experiments conducted in this thesis established proof-of-concept Kalman Filter implementations which utilized distance measurements and Line Of Sight (LOS) information.

This thesis answered two research questions. The first research question was related to how Kalman Filters could be utilized in UWB indoor positioning. The research question was answered by conducting a literature review on the properties of UWB and different Kalman Filter implementations, such as Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF). The second research question asked how different Kalman Filter implementations would perform in the indoor positioning setup of this thesis. This research question was answered by conducting three experiments. The first experiment consisted of collecting LOS information for the Kalman Filters. Experiments two and three consisted of implementing an EKF and UKF for two distinct trajectories. The data collected for these experiments was gathered in Veracell's office setup.

Based on the results I argue that Kalman Filters would suit Veracell's UWB indoor positioning system (IPS). They provided reasonable accuracy and significantly faster results than the optimization method. Of course, more investigation and empirical tests are required to implement Kalman Filters which would perform generally well in Veracell's indoor positioning system.

REFERENCES

[1] S. E. Tom, R. A. Hubbard, P. K. Crane, *et al.*, “Characterization of dementia and alzheimer’s disease in an older population: Updated incidence and life expectancy with and without dementia”, eng, *American journal of public health (1971)*, vol. 105, no. 2, pp. 408–413, 2015, ISSN: 0090-0036.

[2] T. Haute, E. Poorter, P. Crombez, *et al.*, “Performance analysis of multiple indoor positioning systems in a healthcare environment”, eng, *International journal of health geographics*, vol. 15, no. 1, pp. 7–7, 2016, ISSN: 1476-072X.

[3] G. M. Netscher, J. Jacquemot, B. Zylstra, *et al.*, “Td-p-018: Proactive alzheimer’s care through wearable computing at home: System architecture, simplified indoor positioning, and beta test results”, eng, *Alzheimer’s & dementia*, vol. 12, P159–P160, 2016, ISSN: 1552-5260.

[4] *Understanding GPS : principles and applications*, eng, 2nd ed., ser. Artech House mobile communications series. Boston: Artech House, 2005, ISBN: 978-1-58053-894-7.

[5] F. J. Aranda, F. Parralejo, F. J. Álvarez, and J. A. Paredes, “Performance analysis of fingerprinting indoor positioning methods with ble”, eng, *Expert systems with applications*, vol. 202, pp. 117 095–, 2022, ISSN: 0957-4174.

[6] F. Liu, J. Zhang, J. Wang, H. Han, and D. Yang, “An uwb/vision fusion scheme for determining pedestrians’ indoor location”, eng, *Sensors (Basel, Switzerland)*, vol. 20, no. 4, pp. 1139–, 2020, ISSN: 1424-8220.

[7] S. Antonov, A. Fehn, and A. Kugi, “Unscented kalman filter for vehicle state estimation”, eng, *Vehicle system dynamics*, vol. 49, no. 9, pp. 1497–1520, 2011, ISSN: 0042-3114.

[8] C. Naab and Z. Zheng, “Application of the unscented kalman filter in position estimation a case study on a robot for precise positioning”, eng, *Robotics and autonomous systems*, vol. 147, pp. 103 904–, 2022, ISSN: 0921-8890.

[9] T. Ayabakan and F. Keresteciogˇlu, “Rssi-based indoor positioning via adaptive federated kalman filter”, eng, *IEEE sensors journal*, vol. 22, no. 6, pp. 5302–5308, 2022, ISSN: 1530-437X.

[10] C.-H. Wu, Y.-S. Lin, C.-H. Wang, and C. H. Chen, “A compact ltcc ultra-wideband bandpass filter using semi-lumped parallel-resonance circuits for spurious suppression”, eng, in *2007 European Microwave Conference*, IEEE, 2007, pp. 532–535, ISBN: 9782874870019.

[11] H. Nikookar, *Introduction to ultra wideband for wireless communications*, eng, ser. Signals and communication technology. Berlin: Springer, 2009, ISBN: 1-4020-6633-3.

[12] L. Zhu, *Microwave bandpass filters for wideband communications*, eng, 1st edition, ser. Wiley Series in Microwave and Optical Engineering. Hoboken: John Wiley, 2012, ISBN: 1-280-59020-3.

[13] Qorvo, *Dw1000*. [Online]. Available: https://www.qorvo.com/products/p/DW1000 (visited on 07/22/2022).

[14] *UWB theory and applications*, eng. Chichester: Wiley, 2004, ISBN: 1-280-27234-1.

[15] Y. Qi, *Wireless geolocation in a non-line-of-sight environment*. Princeton University, 2003.

[16] H. V. Poor, *An introduction to signal detection and estimation*. Springer Science & Business Media, 1998.

[17] S. A. Zekavat, *Handbook of position location : theory, practice and advances*, eng, ser. IEEE series on digital & mobile communication ; 28. Oxford: Wiley-Blackwell, 2011, ISBN: 1-299-31839-8.

[18] Z. Sahinoglu, S. Gezici, and I. Güvenc, *Ultra-wideband Positioning Systems: Theoretical Limits, Ranging Algorithms, and Protocols*, eng. Cambridge: Cambridge University Press, 2008, ISBN: 9780521187831.

[19] A. Mallat, J. Louveaux, and L. Vandendorpe, "Uwb based positioning in multipath channels: Crbs for aoa and for hybrid toa-aoa based methods", eng, in *2007 IEEE International Conference on Communications*, IEEE, 2007, pp. 5775–5780, ISBN: 9781424403530.

[20] Qorvo, *Qorvo dw3000 user guide*. [Online]. Available: https://www.qorvo.com/products/d/da008154 (visited on 01/07/2023).

[21] T. Laadung, S. Ulp, M. M. Alam, and Y. L. Moullec, "Novel active-passive two-way ranging protocols for uwb positioning systems", eng, *IEEE sensors journal*, vol. 22, no. 6, pp. 5223–5237, 2022, ISSN: 1530-437X.

[22] T. Ye, M. Walsh, P. Haigh, J. Barton, and B. O'Flynn, "Experimental impulse radio ieee 802.15. 4a uwb based wireless sensor localization technology: Characterization, reliability and ranging", in *ISSC 2011, 22nd IET Irish Signals and Systems Conference, Dublin, Ireland. 23-24 Jun 2011*, Insitution of Engineering and Technology, 2011.

[23] D. Neirynck, E. Luk, and M. McLaughlin, "An alternative double-sided two-way ranging method", eng, in *2016 13th Workshop on Positioning, Navigation and Communications (WPNC)*, ser. Workshop on Positioning Navigation and Communication, IEEE, NEW YORK: IEEE, 2016, pp. 1–4, ISBN: 1509054405.

[24] P. Muller, H. Wymeersch, and R. Piche, "Uwb positioning with generalized gaussian mixture filters", eng, 2014-10.

[25] Qorvo, *Qorvo dw1000 application notes*. [Online]. Available: https://www.qorvo.com/products/p/DW1000#documents (visited on 12/05/2022).

[26] R. Fletcher, *Practical Methods of Optimization*, eng, 2nd ed. Hoboken: Wiley, 2013, ISBN: 1-118-72318-X.

[27] R. V. Rao, *Teaching Learning Based Optimization Algorithm And Its Engineering Applications*, eng, 1st ed. 2016. Cham: Springer International Publishing, 2016, ISBN: 3-319-22732-7.

[28] J. Nocedal, *Numerical Optimization*, eng, 2nd ed. 2006., ser. Springer Series in Operations Research and Financial Engineering. New York, NY: Springer New York, 2006, ISBN: 0-387-40065-6.

[29] P. Wolfe, "Convergence conditions for ascent methods", eng, *SIAM review*, vol. 11, no. 2, pp. 226–235, 1969, ISSN: 0036-1445.

[30] SciPy, *Scipy documentation*. [Online]. Available: https://docs.scipy.org/doc/scipy/reference/optimize.minimize-bfgs.html#optimize-minimize-bfgs (visited on 08/26/2022).

[31] G. Welch, G. Bishop, *et al.*, "An introduction to the kalman filter", 1995.

[32] S. Särkkä, *Bayesian filtering and smoothing*, eng. 2010, ISBN: 9781139344203.

[33] R. van der Merwe and E. Wan, "Gaussian mixture sigma-point particle filters for sequential probabilistic inference in dynamic state-space models: Signal processing theory and methods", eng, in *2003 IEEE INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, VOL VI, PROCEEDINGS*, IEEE, NEW YORK: IEEE, 2003, pp. 701–704, ISBN: 9780780376632.

[34] S. Julier and J. Uhlmann, "Unscented filtering and nonlinear estimation", eng, *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, 2004, ISSN: 0018-9219.

[35] R. Labbe, *Kalman and Bayesian Filters in Python*, eng. 2020.

[36] Google, *Google cloud functions*. [Online]. Available: https://cloud.google.com/functions (visited on 09/18/2022).

[37] T. Kluyver, B. Ragan-Kelley, and F. Pérez, "Jupyter notebooks – a publishing format for reproducible computational workflows", in *Positioning and Power in Academic Publishing: Players, Agents and Agendas : Proceedings of the 20th International Conference on Electronic Publishing*, F. Loizides and B. Schmidt, Eds., IOS Press, 2016, pp. 87–90.

[38] DBeaver, *Dbeaver software*. [Online]. Available: https://dbeaver.io/ (visited on 01/02/2023).

[39] R. Van Der Merwe, *Sigma-point Kalman filters for probabilistic inference in dynamic state-space models*. Oregon Health & Science University, 2004.

[40] B. W. Kernighan and D. M. Ritchie, *The C programming language*. 2006.

[41] G. Van Rossum and F. L. Drake, *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009, ISBN: 1441412697.

[42] F. Di Gregorio and D. Varrazzo, *Psycopg library*. [Online]. Available: https://www.psycopg.org/docs/ (visited on 01/04/2023).

[43] W. McKinney *et al.*, “Data structures for statistical computing in python”, in *Proceedings of the 9th Python in Science Conference*, Austin, TX, vol. 445, 2010, pp. 51–56.

[44] C. R. Harris, K. J. Millman, S. J. van der Walt, *et al.*, “Array programming with NumPy”, *Nature*, vol. 585, pp. 357–362, 2020. DOI: 10.1038/s41586-020-2649-2.

[45] J. D. Hunter, “Matplotlib: A 2d graphics environment”, *Computing in science & engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[46] M. Waskom, O. Botvinnik, D. O’Kane, *et al.*, *Mwaskom/seaborn: V0.8.1 (september 2017)*, version v0.8.1, Sep. 2017. DOI: 10.5281/zenodo.883859. [Online]. Available: https://doi.org/10.5281/zenodo.883859.

[47] C. Minoiu, *Kernel density estimation based on grouped data : the case of poverty assessment*, eng, ser. IMF working paper ; WP/08/183. Washington, District of Columbia: International Monetary Fund, 2008 - 2008, ISBN: 1-4623-9111-7.

[48] J. S. Liu, *Monte Carlo Strategies In Scientific Computing*, eng, ser. Springer Series in Statistics. New York, NY: Springer, 2001, ISBN: 9780387952307.